



University of South Wales
Prifysgol De Cymru
Faculty of Computing, Engineering and Science

**MSc Project: Enhancing Phishing Detection Accuracy: A
Lightweight Domain-based Approach Using Supervised
Machine Learning**

Name: Shohag Mia
Enrollment ID: 30107619

First Supervisor
Ian Wilson
Associate Professor
Faculty of Computing, Engineering and Science

Second Supervisor
Dr. Gaylor Boobyer

Year Of Study: 2023-2024
Course: MSc Artificial Intelligence

Declaration

University of South Wales

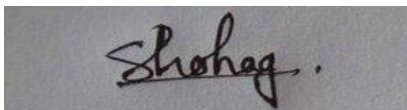
Prifysgol De Cymru

Faculty of Computing, Engineering and Science

STATEMENT OF ORIGINALITY

This is to certify that, except where specific reference is made, the work described in this project is the result of the investigation carried out by the student, and that neither this project nor any part of it has been presented, or is currently being submitted in candidature for any award other than in part for the MSc award, Faculty of Computing, Engineering and Science from the University of South Wales.

Signed

A rectangular box containing a handwritten signature in black ink. The signature appears to be 'Shohag' followed by a period.

Student

Abstract

Phishing attacks pose a significant threat in the digital age by tricking victims into disclosing sensitive information through social engineering. These attacks are diverse and evolving, making detection a challenging task for both individuals and security experts. Existing detection methods are often static, relying on predefined rules and patterns. This thesis addresses this issue by proposing a data-driven, domain-based model using advanced machine learning algorithms for phishing detection. The study evaluates 6 classifiers. Among all, the Random Forest Classifier, with an accuracy of 80.3%, proved to be the most effective, showing strong precision (76% for legitimate and 84.4% for phishing emails) and recall (86.4% for legitimate and 73% for phishing emails), alongside balanced F1-scores. Despite its longer training time (33.03 seconds) and moderate testing time (1.9 seconds), its reliability and performance make it superior and suitable to use in real world application. The findings expected to help security expert to design more sophisticated tool to combat phishing.

Acknowledgements

I want to express my heartfelt gratitude to my supervisors Ian Wilson and Dr. Gaylor Boobyer for their unwavering support, insightful feedback, and guidance throughout this project. Their expertise has truly shaped my work, and their feedback pushed me to reach higher standards.

I am also deeply thankful to my father, mother, and brother – their love, encouragement, and belief in me have been my greatest source of strength. Without their constant support, this journey would have been far more difficult. Lastly, I appreciate everyone who has played a role in helping me grow and achieve more than I ever thought possible.

Lastly, I would like to express my sincere appreciation to the University of South Wales for providing the resources, support, and nurturing environment that made this project possible.

Table of Contents

Declaration	2
Abstract	3
Acknowledgements	4
Chapter 1 – Introduction	9
1.1. Introduction.....	9
1.2. Problem Statement.....	10
1.3. Motivation.....	11
1.4. Thesis Significance.....	12
1.5. Thesis Outline.....	12
Chapter 2 – Literature Review	13
2.1. Introduction.....	13
2.2. Techniques of Phishing Combatting.....	13
2.2.1. Visual-Similarity or Content-Based Approaches.....	13
2.2.2. Rule-based and Heuristic-based Approaches.....	14
2.2.3. ML-based Approaches.....	15
2.2.4. Existing Gaps and Scope to be Filled.....	16
Chapter 3 – Methodology	17
3.1. Introduction.....	17
3.1.1. Data Collection.....	17
3.1.2. Data Preprocessing.....	17
3.1.3. Supervised Machine Learning Algorithm Selection.....	17
3.1.4. Evaluation Metrics.....	17
3.1.5. Final Model Deployment.....	18
3.2. Raw Data Acquisition.....	19
3.3. Data Preprocessing.....	19
3.3.1. Feature Extraction.....	19
3.3.2. Handling Class Imbalance.....	21
3.3.3. Feature Selection.....	23
3.4. Selected Supervised Machine Learning Algorithms.....	25
3.5. Evaluation Metrics.....	27
3.6. Final Model Deployment.....	28

Chapter 4 – Experimental Results	29
4.1. Random Forest Classifier.....	29
4.2. Decision Tree Classifier.....	30
4.3. Gaussian Naive Bayes.....	31
4.4. SGDClassifier.....	33
4.5. Extra Trees Classifier.....	34
4.6. Multilayer Perceptron (MLP).....	35
4.7. ROC Curve analysis.....	37
4.8. Precision-recall Curve analysis.....	38
4.9. Performance Comparison Among the Classifiers.....	39
4.10. The final model (best performer).....	39
Chapter 5 – Discussion	41
4.1. Introduction.....	41
4.2. Comparison with Existing Studies.....	41
Chapter 5 – Conclusion	46
References	48

List of Tables

Table 3.1 Generated feature list	20
Table 3.2. Detailed of the machine learning algorithms	27
Table 4.1. Classification report of Random Forest Classifier	29
Table 4.2. Classification report of Decision Tree Classifier	30
Table 4.3. Classification report of Gaussian Naive Bayes	32
Table 4.4. Classification report of SGDClassifier	33
Table 4.5 Classification report of Extra Trees Classifier	35
Table 4.6 Classification report of Logistic Regression	36
Table 5.1. Comparison with the existing studies	43

List of Figures

Figure 1.1. Number of unique phishing websites (2011-2022)	10
Figure 3.1 Methodological Framework	18
Figure 3.2 Before handling class imbalance	22
Figure 3.3 After handling class imbalance	22
Figure 3.4 Feature importance	24
Figure 3.5 Correlation between the features	25
Figure 4.1 Confusion matrix (Random Forest Classifier)	30
Figure 4.2 Confusion matrix (Decision Tree Classifier)	31
Figure 4.3 Confusion matrix (Gaussian Naive Bayes)	33
Figure 4.4 Confusion matrix (SGDClassifier)	34
Figure 4.5 Confusion matrix (Extra Trees Classifier)	35
Figure 4.6 Confusion matrix (MLP)	36
Figure 4.7. ROC Curve Analysis	37
Figure 4.8 Precision-Recall Curve Analysis	38

Chapter 1 – Introduction

1.1. Introduction

Currently, the world is going through a technological revolution, and the Internet has become more accessible than ever before. As of April 2024, the number of internet users worldwide was 5.44 billion, which is 66 percent of the global population (Statista, 2024). Due to the exponential growth of the internet, many traditional systems as well as people's daily lifestyles have undergone significant transformations. Simultaneously, suspicious online activities have increased alarmingly, leading to the increased occurrence of 'cybercrime' which, in turn, undermines not only the global economy and national security but also social stability and individual interests (Tamal et al., 2024). According to the 2020 Official Annual Cybercrime Report, cybercrime is one of the greatest challenges that humanity will face in the next two decades (Pereira, 2020). In the current digital landscape, cybercriminals use a variety of tactics to ensnare their targets, with phishing emerging as the most prevalent yet dynamic and dangerous strategy. Phishing has been defined in various ways due to its continuous evolution and contextual differences, leading to the absence of a universally accepted definition (Tamal et al., 2024; Alkhalil et al., 2021). Broadly speaking, phishing is a form of social engineering in which attackers use social skills to gather information about individuals, organizations, or their computer systems. Instead of relying on technical or coding-based methods, these attackers exploit human vulnerabilities and psychological manipulation, hence the term "human hacking" is often linked to phishing (Klimburg-Witjes and Wentland, 2021). Typically, a phishing attack follows three main phases: Lure, Hook, and Catch. Unlike other targeted cyberattacks, phishing is highly heterogeneous, affecting various targets with different motivations and goals (Tamal et al., 2024). This variability, combined with constantly evolving and sophisticated tactics, makes phishing detection a significant challenge for both individual users and security professionals. As a result, phishing has become one of the most organized, challenging, and difficult-to-detect cyber threats of the 21st century. Despite being an old technique, the number of unique phishing attacks continues to rise (see Figure 1.1). According to the Anti-Phishing Working Group (APWG), the fourth quarter of 2023 saw 1,077,501 phishing attacks, with almost five million attacks throughout the year, marking 2023 as the worst year for phishing on record (Anti-Phishing Working Group, 2023). This clearly indicates a continuous upward trend in

phishing incidents. Phishing activity trends reported by the APWG also revealed that in 2015 and 2016, the primary targets were Internet Service Providers (ISPs) and retailers. However, in 2017 and 2018, there was a shift towards targeting various payment services such as Shopify Payments and Amazon Pay. From 2019 to 2023, the most targeted areas included software-as-a-service (SaaS), payment portals, financial institutions, social media platforms, and e-commerce websites. In 2023, social media became the most targeted sector, accounting for 43% of all attacks. Phishing attacks against financial institutions decreased from 24.9% in Q3 to 14% in Q4, while attacks on online payment services constituted 4% of all attacks. This rising trend and the shifting focus of phishing attacks highlight the shortcomings of current anti-phishing methods, demonstrating that existing countermeasures are insufficient in detecting and preventing these threats.

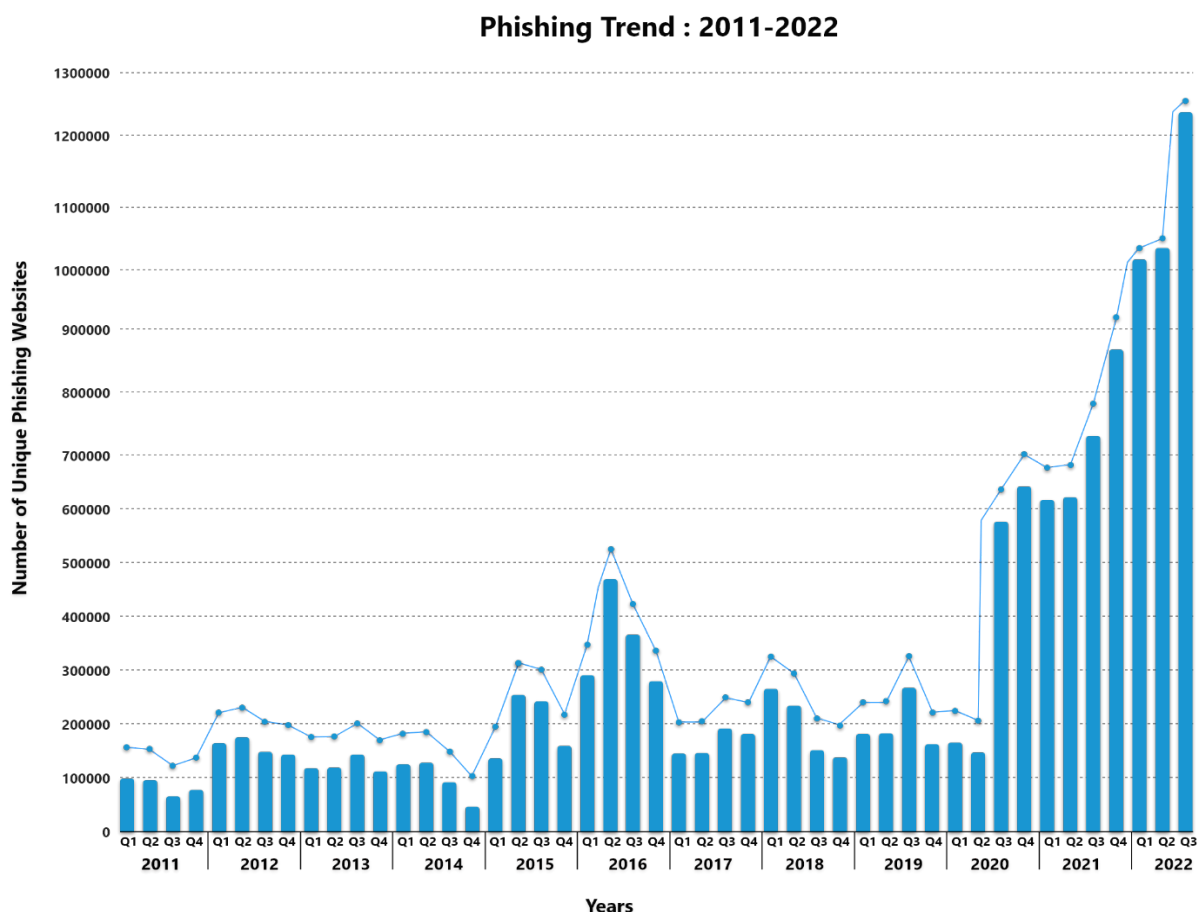


Figure 1.1. Number of unique phishing websites (2011-2022)

1.2. Problem Statement

Most current phishing detection techniques are static in nature and often produce a high rate of false positives. Unlike other forms of cybercrime, where the attacker's motives and victim types are consistent, phishers have varying goals, motivations, and target types, leading them to frequently change their attacking techniques. This dynamic nature of fraudulent activities has made phishing detection increasingly challenging, contributing to an exponential rise in phishing attacks in recent years. This growing trend underscores the limitations of existing anti-phishing tools and necessitates the development of more advanced techniques by researchers and security experts.

1.1. Thesis Objective

The primary objective of this study is to create a safer cyberspace for individuals and organizations by bolstering our collective resilience against phishing attacks. To this end, this study is expected to design and develop of an efficient, lightweight, data-driven, URL domain-based predictive model for phishing detection utilizing state-of-the-art supervised machine learning algorithms. The specific objectives of this study can be presented below:

- ❑ Conduct a comprehensive literature review to identify current phishing detection techniques, their limitations, and areas for new contributions.
- ❑ Collect a comprehensive dataset of phishing and legitimate URLs from credible and reliable online sources (e.g., OpenPhish, DomCop).
- ❑ Extract relevant features (e.g. domain properties) indicative of phishing attempts based on previous studies and URLs analysis.
- ❑ Clean and prepare the extracted data for optimal machine learning model performance.
- ❑ Explore, evaluate, and fine-tune various supervised machine learning algorithms to determine the most effective one for phishing detection.

1.3. Motivation

In January 2015, an email arrived in the inboxes of several employees at Bangladesh Bank (BB). The email appeared to be from a job seeker named Rasel Ahlam, politely requesting assistance in downloading his CV and cover letter from a provided link. However, this seemingly routine message was far from ordinary. It was a spear-phishing attempt, and Rasel

Ahlam was a fabricated identity used by the notorious North Korean hacking group known as Lazarus, as identified by the FBI. The subsequent events are well-documented: hackers successfully siphoned off \$81 million from Bangladesh Bank's account at the Federal Reserve Bank of New York through the SWIFT network, in what is now infamous as the Bangladesh Bank Cyber Heist. This incident is not unique to Bangladesh; it is part of a broader global challenge where cybercrime poses a significant threat to financial institutions, organizations, and individuals alike. With the rapid expansion of internet access and advancements in Information and Communication Technology (ICT), cybersecurity has become a critical concern worldwide. Bangladesh (my homeland), despite its substantial progress in ICT under the government's forward-looking policies, is increasingly vulnerable in the cyber domain. So, as a Computer Science and Engineering student, I feel a deep responsibility to contribute to addressing the growing threat of cybercrime, with a particular focus on combating phishing attacks.

1.4. Thesis Significance

The significance of the study can be drawn in many ways. Primarily, the proposed approach aims to curb the increasing trend of phishing attacks. Unlike traditional static strategies for phishing threat mitigation, this thesis suggests a machine learning-based, real-world data-driven method to detect phishing sites using domain-based features. This approach is expected to classify the dynamically changing fraudulent strategies with higher accuracy. Additionally, since the proposed solution is lightweight and entirely client-side, it eliminates the need for third-party services or tools. Ultimately, the development and deployment of a machine learning-based anti-phishing tool, browser extension, or security recommendation tool based on this approach could revolutionize the security systems.

1.5. Thesis Outline

The remaining chapters of this thesis are structured as follows: **Chapter 2**– Literature Review: A comprehensive literature review is conducted step-by-step. **Chapter 3** – Methodology: This chapter details the methodological approach taken in this research. **Chapter 4 - Experimental Results:** The findings of the research are presented in detail, aligned with the research objectives and questions. **Chapter 5** – Discussion: This chapter

interprets the findings in the context of existing studies. **Chapter 6 – Conclusion:** The thesis concludes by highlighting its limitations and suggesting future research directions.

Chapter 2 – Literature Review

1.1. Introduction

Phishing identification has emerged as a significant challenge, driven by the increasingly sophisticated tactics employed by attackers. Researchers are continuously developing more advanced models to predict or detect suspicious websites. However, the rapidly evolving nature of phishing techniques often outpaces these detection tools, leading to the compromise of sensitive information belonging to individuals and organizations.

1.2. Techniques of Phishing Combatting

1.2.1. Visual-Similarity or Content-Based Approaches

To counteract phishing, researchers often rely on visual-similarity or content-based models (Jain et al., 2020; Wardman et al., 2011). Typically, features such as logos, HTML tags, text formatting, Cascading Style Sheets (CSS), images, text content, and Document Object Model (DOM) structures are used to differentiate between legitimate and phishing websites (Jain and Gupta, 2017). For instance, Chen, Ma and Huang (2020) proposed an intelligent visual technique for detecting phishing websites, categorizing them as very similar, locally similar, or non-imitating. The findings showed that the wavelet Hashing (wHash) mechanism with a color histogram outperformed the perceptual Hashing (pHash) mechanism, while the SIFT technique achieved accuracies of up to 99.95% for specific data sets like Microsoft, Dropbox, and Bank of America. In another study (Chiew et al., 2015), researchers proposed a two-phase model where they collected the logo from the suspicious web page in the first phase and then performed a Google image search to obtain the ethnicity portrayed in the second phase. However, they did not mention the exact accuracy of their proposed system. Another study (Ardi and Heidemann, 2016) proposed a browser plugin called AuntieTuna that could more reliably detect phishing sites using cryptographic hashing. In this case, they used the DOM (Document Object Model) elements of the browser and claimed a detection accuracy of more than 50% with no FP (false positive) alarm. However, the main drawback of

this approach is that if the phisher produces different DOMs for the same website, or if the website includes only photos, this kind of approach will not work. A text-based approach where text features were analyzed to detect suspicious SMS has been proposed in another study (Mishra and Soni, 2019). However, another review study (Jain and Gupta, 2017) revealed that this type of approach will not work for different languages. Concurrently, some of the phishing websites contain only images. For those cases, this kind of approach won't work.

1.2.2. Rule-based and Heuristic-based Approaches

For detecting phishing sites, rule-based and heuristic-based models are prominent approaches for detecting phishing websites. In some cases, a combination of these models, as demonstrated by Jeeva and Rajsingh (2016), is employed to enhance detection. These approaches are often easier to implement compared to more complex methods. They rely on "if-then" rules, which are generated based on existing phishing site samples. For example, Adewole et al. (2019) developed a hybrid rule-based model that generated 55 rules from 30 features of phishing websites, achieving an average detection accuracy of 96.8%. However, a significant limitation of rule-based methods is their inability to learn from experience, unlike machine learning algorithms (Tamal et al., 2024). This drawback prevents these models from effectively distinguishing between legitimate and phishing sites when attackers change their tactics. Additionally, both rule-based and heuristic-based models struggle with handling incomplete information. Recently, URL-based features have gained popularity among researchers for identifying suspicious websites (Al-Haija and Badawi, 2021; Aljofey et al., 2022). For instance, Feroz and Mengel (2015) employed lexical and host-based features to detect phishing URLs, achieving accuracy rates between 93% and 98%. Furthermore, Sahingoz et al. (2019) proposed a language-independent, real-time anti-phishing system that utilized 27 NLP-based features derived from raw URLs, leveraging conventional machine learning algorithms for detection. A recent study by Ubung et al. (2019) demonstrated a 95% accuracy rate in detecting phishing sites through ensemble learning, specifically using majority voting, based on seven key URL features. These findings, along with other research, highlight the effectiveness of URL-based features as a powerful tool for identifying phishing sites compared to alternative methods. However, the success of URL-based detection is

closely tied to the quality of the features selected, making feature selection a critical aspect of phishing detection strategies.

1.2.3. ML-based Approaches

In ML-based approaches, both supervised and deep-learning algorithms are frequently followed to combat phishing attacks. For instance, Pandey and Mishra (2023) aimed to develop a rapid and efficient anti-phishing solution that can detect phishing websites without relying heavily on third-party services. Their study revealed that dominant color features and popular brand names are effective indicators for phishing website detection, with the Random Forest algorithm outperforming other machine learning algorithms by achieving a true positive rate of 98.43% and an accuracy of 99.13%. The proposed approach showed good performance for real-time applications with a prediction runtime of 7.6 seconds per webpage. However, the study's limitations include a primary focus on dominant color features and popular brand names, which might neglect other relevant features. Additionally, the effectiveness of the model could be compromised by evolving phishing website designs and tactics over time, and its reliance on popular brand names may hinder the detection of novel phishing attacks targeting lesser-known brands. On the other hand, Jha, Muthalagu, and Pawar (2023) employed a ML-based approach with the objective of developing a tool capable of accurately distinguishing phishing websites from legitimate ones in real time. Their findings indicated that the pipelined Logistic Regression model demonstrates high accuracy, around 98%, in detecting phishing websites while maintaining real-time performance. Despite these promising results, the model's performance might be susceptible to changes in phishing techniques, and its effectiveness could vary across different datasets or geographical regions. In contrast, Nanda and Goel (2024) proposed a Bidirectional Long Short-Term Memory (BiLSTM), Gated Highway Attention (GHA) block, and Convolutional Neural Network (CNN) approach to develop a robust and efficient phishing URL detection system that overcomes limitations in existing techniques. Their BiLSTM-GHA-CNN model significantly outperformed state-of-the-art techniques in detecting phishing URLs. However, this approach also has limitations, including dependence on the quality and diversity of the training dataset, potential challenges in generalizing the model to new phishing tactics, and the computational complexity and resource requirements for real-time deployment in large-scale environments. Kumar et al. (2023) took a different

ML-based approach by developing a novel phishing detection technique capable of operating at the transport layer, thereby bypassing the limitations of existing application-layer methods. Their findings indicated that the proposed model effectively detects phishing URLs in encrypted traffic without decryption, using machine learning algorithms such as Random Forest, XGBoost, and Light GBM. Light GBM achieved the highest accuracy of 95.40% in detecting phishing URLs. However, the model's effectiveness relies on the discriminative power of features extracted from TLS traffic, which could be affected by changes in TLS protocols or attacker behavior. Finally, Tamal et al. (2024) propose a URL-based approach with the objective of developing a robust, effective, and reliable phishing detection system. Their key findings show that the OFVA method effectively extracts relevant features from URLs for phishing detection, with the Random Forest (RF) classifier outperforming other algorithms in terms of accuracy (97.52%), precision, and AUC. However, the study focuses on model development and evaluation, and its practical implementation and performance in real-world scenarios are not explicitly addressed.

1.2.4. Existing Gaps and Scope to be Filled

Although there are few phishing detections approaches out there, most of the existing approaches are comparatively old, inherently static and often give a high false-positive alarm (Vayansky and Kumar, 2018; Odeh, Keshta and Abdelfattah, 2021; Odeh, Keshta and Abdelfattah, 2021). On the other side, phishers frequently change their attacking approaches. As a consequence, this dynamism, coupled with constantly evolving and sophisticated tactics, makes phishing detection a critical challenge for both individual users and security professionals alike (Tamal et al., 2024). This significant disparity between the dynamic nature of phishing attempts and the static and typical nature of existing detection methods underscores the urgent need for innovative and out of the box approaches to combat this growing threat. In light of this, this thesis aims to addresses this need by proposing the development of an efficient, lightweight, data-driven, domain-based model for phishing detection utilizing state-of-the-art machine learning algorithms.

Chapter 3 – Methodology

2.1. Introduction

In this chapter the thesis methodology was described in step-by-step manner. In figure 3.1, the methodological framework is systematically outlined, which illustrates the comprehensive procedural approaches employed in this research. The methodology section is structured into several distinct phases, each critical to achieving the research objectives:

2.1.1. Data Collection

This phase involves the collection of a diverse dataset, focusing on domain-based features relevant to phishing detection. The data is sourced from various repositories, ensuring a broad representation of phishing and legitimate domains.

2.1.2. Data Preprocessing

In this phase, the raw data undergoes several preprocessing steps to enhance its quality and suitability for analysis. These steps include:

- **Data Cleaning:** Removing inconsistencies, missing values, and outliers to ensure the dataset is clean and reliable.
- **Feature Extraction:** Identifying and extracting key domain-based features that are indicative of phishing activity, such as domain age, WHOIS information, and DNS records.
- **Feature Selection:** Applying statistical methods and machine learning techniques to select the most relevant features, reducing dimensionality and improving model performance.

2.1.3. Supervised Machine Learning Algorithm Selection

This phase involves selecting a range of supervised machine learning algorithms from various families, such as decision trees, support vector machines, and ensemble methods. Each algorithm is chosen based on its suitability for the specific characteristics of the dataset and the nature of phishing detection.

2.1.4. Evaluation Metrics

Standard evaluation metrics are applied to compare the performance of the selected machine learning algorithms. Metrics such as accuracy, precision, recall, F1-score, and AUC-ROC are utilized to provide a comprehensive assessment of each model's effectiveness in detecting phishing attacks.

2.1.5. Final Model Deployment

The phase culminates in the deployment of the best-performing model. The selected model is rigorously tested and validated, ensuring its robustness and reliability in real-world scenarios. The deployment phase also includes considerations for scalability, efficiency, and integration into existing cyber security infrastructures.

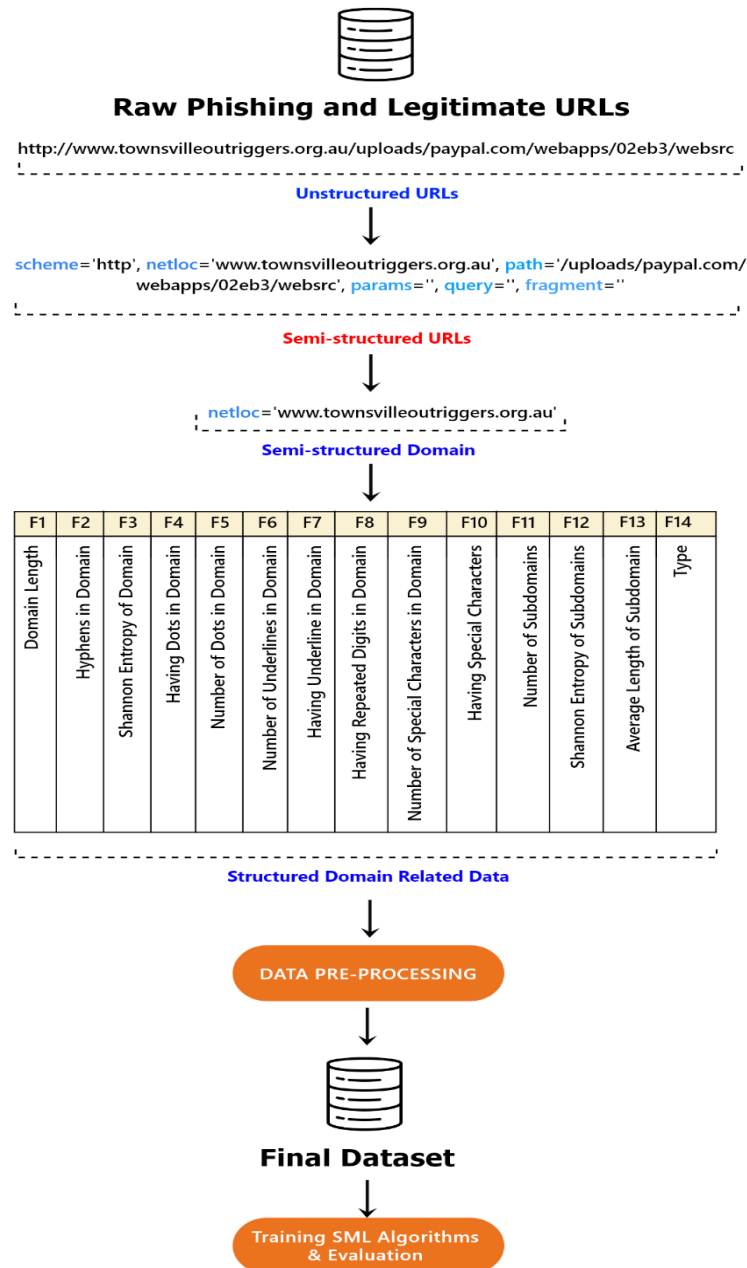


Figure 3.1 Methodological Framework

2.2. Raw Data Acquisition

In the initial phase of the study, raw and unstructured phishing and legitimate URLs were gathered and consolidated from various credible and trustworthy sources. Given the importance of data volume and quality in machine learning-based approaches, this study leveraged a large dataset to mitigate potential issues such as data insufficiency, bias, or class imbalance, which could otherwise lead to poor or inaccurate predictions. The dataset comprised a total of 245,502 URLs, of which 67,467 were identified as phishing URLs

sourced from the OpenPhish Database (OpenPhish, 2024). OpenPhish is a phishing intelligence platform that assists organizations in detecting and preventing phishing attacks. Conversely, 178,035 legitimate URLs were obtained from DomCop, a web-based software designed to facilitate the discovery of expired or expiring domains and to archive these domains (DomCop, 2024). All URLs were retained in their original, unstructured format (e.g., <https://quillbot.com/>), lacking any specific organization or structure suitable for direct analysis.

2.3. Data Preprocessing

2.3.1. Feature Extraction

In the second phase, the "urllib.parse" Python module was employed to transform the raw URLs into semi-structured components, such as the scheme, network location, and path. Once the URLs were converted into this semi-structured format, the next step involved extracting domain-related features (see Figure 3.1). This process began with a series of functions to each domain in the dataset, aiming to quantify specific attributes indicative of suspicious or phishing behavior (see Appendix A). Firstly, the length of the domain was calculated, measuring the total number of characters it contained—a straightforward feature captured by the `domain_length` function. Following this, the number of hyphens within the domain was counted using the `count_hyphens` function, as an excessive number of hyphens often signals phishing. The Shannon entropy of the domain was then computed using the `shannon_entropy` function, which measures the randomness or unpredictability of the domain name. A higher entropy might suggest that the domain was algorithmically generated, a common trait of phishing domains. The presence of dots in the domain was detected using the `has_dots` function, and the total number of dots was counted with the `count_dots` function. Both features provide insight into the complexity or potential manipulation within the domain name structure. Similarly, the presence and number of underscores were identified through the `has_underline` and `count_underlines` functions, respectively. Another significant feature was the presence of repeated digits within the domain, determined by the `has_repeated_digits` function, as domains with repeated digits may attempt to mimic legitimate ones through minor typographical differences. The presence and count of special characters were captured using the `has_special_characters`

and `count_special_characters` functions, respectively. Special characters may be used in phishing domains to bypass filters or imitate legitimate domains. Additionally, the number of subdomains was calculated using the `count_subdomains` function, potentially indicating the complexity or an attempt to obscure the true domain. The entropy of these subdomains was then measured with the `subdomain_entropy` function to assess randomness within these components. Finally, the average length of the subdomains was computed using the `avg_subdomain_length` function, offering insight into the typical structure of the domain. Once these features were extracted, they were incorporated into the original dataset, enriching it with detailed domain-specific attributes. The enhanced dataset was then saved as a CSV file for further analysis. The complete list of extracted features is presented in table 3.1.

Table 3.1 Generated feature list

SN	Feature	Description	Type
F1	Domain Length	The total number of characters in the domain name.	Int
F2	Hyphens in Domain	Indicates if the domain name contains hyphens.	Bool
F3	Shannon Entropy of Domain	A measure of the randomness or complexity of the domain name.	Float
F4	Having Dots in Domain	Indicates if the domain name contains dots (other than in the TLD).	Bool
F5	Number of Dots in Domain	The total number of dots in the domain name (excluding the TLD).	Int
F6	Having Underline in Domain	Indicates if the domain name contains an underline.	Bool
F7	Number of Underlines in Domain	The total number of underline characters in the domain name.	Int
F8	Having Repeated Digits in Domain	Indicates if the domain name has repeated digits	Bool
F9	Having Special Characters	Indicates if the domain name contains any special characters (e.g., #, \$, &).	Bool
F10	Number of Special Characters in Domain	The total number of special characters in the domain name.	Int
F11	Number of Subdomains	The number of subdomains present in the domain.	Int

F12	Shannon Entropy of Subdomains	A measure of the randomness or complexity of the subdomains.	Float
F13	Average Length of Subdomains	The average length of each subdomain in the domain name.	Float
F14	Type	The label indicating whether the domain is phishing or legitimate.	Bool

2.3.2. Handling Class Imbalance

Upon extracting the relevant features from the dataset, it became evident that there was a significant imbalance between the classes, as illustrated in Figure 3.2. In the context of machine learning, such an imbalanced class distribution poses a critical challenge, as models trained on disproportionate datasets are prone to bias. This often results in the majority class being overrepresented in predictions, while the minority class suffers from high misclassification rates, ultimately leading to poor overall model performance (Hambali, Oladele and Adewole, 2020).

To mitigate this issue, the study employed the Synthetic Minority Oversampling Technique (SMOTE), a sophisticated statistical approach specifically designed to address class imbalance. SMOTE works by generating synthetic samples for the minority class, effectively increasing its representation in the dataset. This process involves selecting instances from the minority class and creating new, similar instances by interpolating between these points. The technique was applied to the training dataset (X_{train} , y_{train}), leading to the creation of a resampled dataset where the minority class was synthetically oversampled to achieve parity with the majority class, thereby ensuring a more balanced distribution (as depicted in Figure 3.2). The resulting resampled training data was then integrated into a new DataFrame. This enhanced dataset, now balanced and optimized for training, was subsequently saved as a CSV file under the name "Resampled_FinalDataset.csv" for future use in model development and evaluation (see Appendix A for more details).

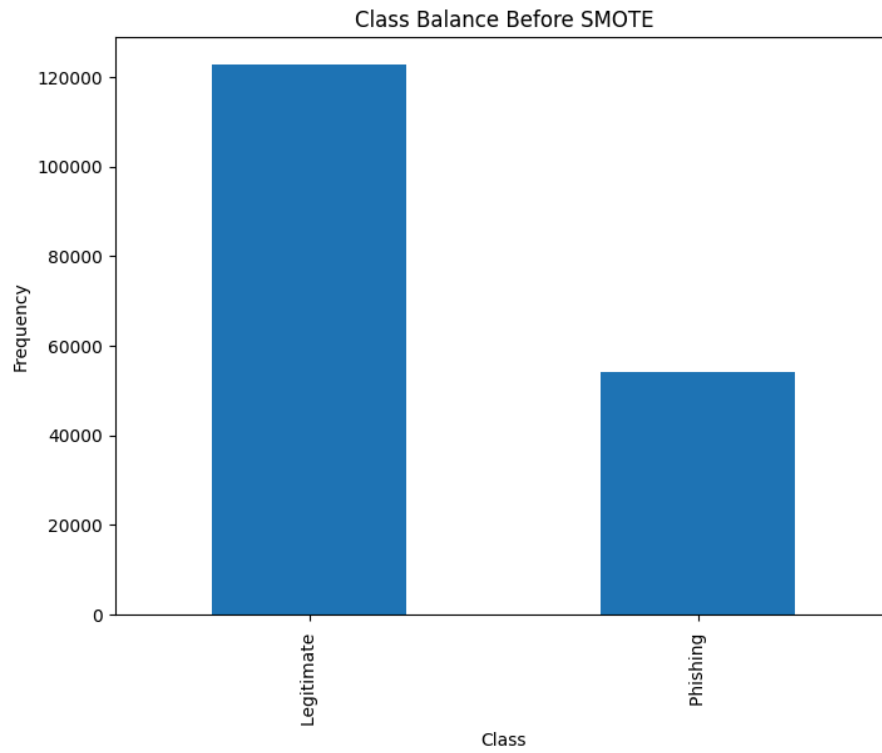


Figure 3.2 Before handling class imbalance

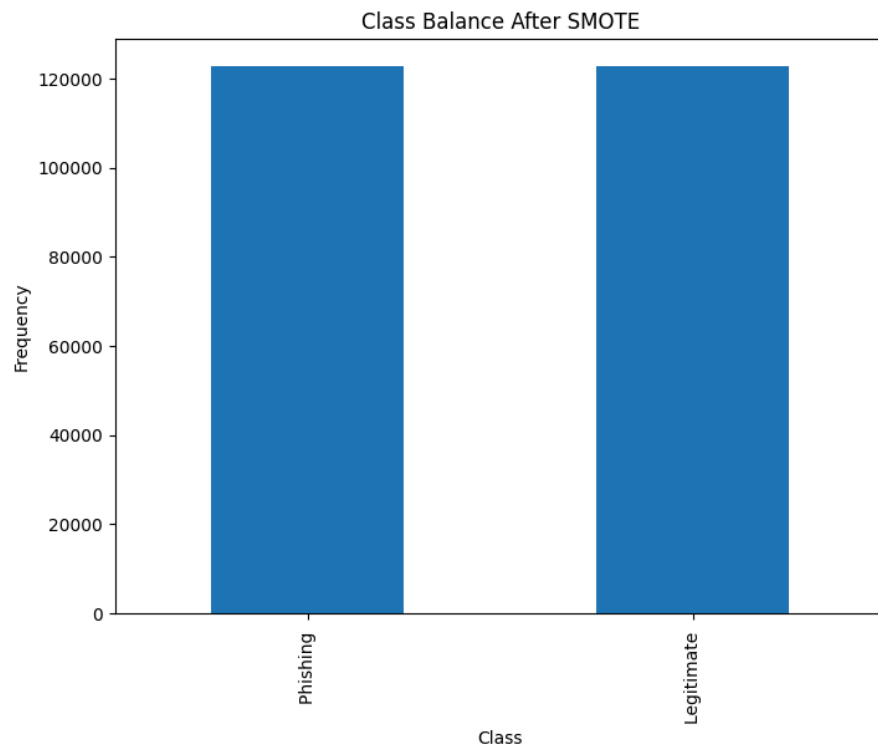


Figure 3.3 After handling class imbalance

2.3.3. Feature Selection

To enhance the efficiency and accuracy of the supervised machine learning (SML) classifiers while reducing model complexity, this study employed feature selection techniques to identify the most informative features. Specifically, the Random Forest classifier was utilized to evaluate feature importance through permutation importance, and the relationships between features were visualized. Initially, the necessary libraries, such as Seaborn for visualization, Pandas for data manipulation, and Scikit-learn for machine learning tasks, were imported. The dataset was then loaded into a DataFrame and split into features (X) and the target variable (y). The data was subsequently divided into training and testing sets. A Random Forest classifier with 100 estimators was defined and trained on the training set. Permutation importance was computed on the test set to assess the significance of each feature. The results were sorted, and a bar plot was created using a viridis colormap to visualize the importance of features, with grid lines added for clarity (see Figure 3.4). The findings revealed that in predicting phishing attacks, the most important features were domain length, Shannon entropy of the domain, Shannon entropy of the sub-domains, and the average length of the sub-domains. Conversely, features such as the presence of an underline in the domain and the number of underlines in the domains were found to be the least significant.

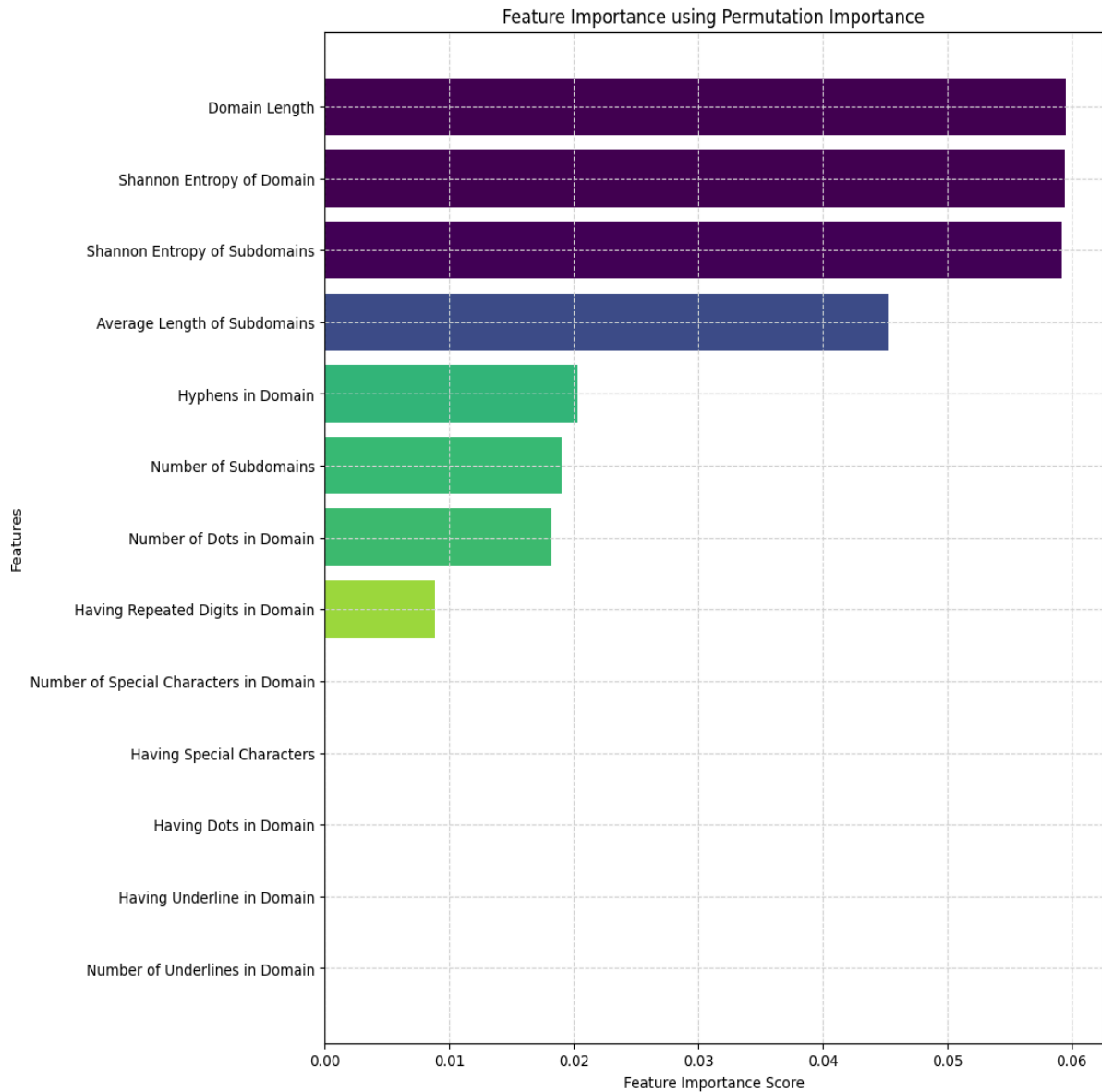


Figure 3.4 Feature importance

Subsequently, a heatmap was also generated to display the correlation matrix of the features, utilizing the coolwarm colormap to highlight positive and negative correlations, aiding in understanding feature relationships (see Figure 3.5).

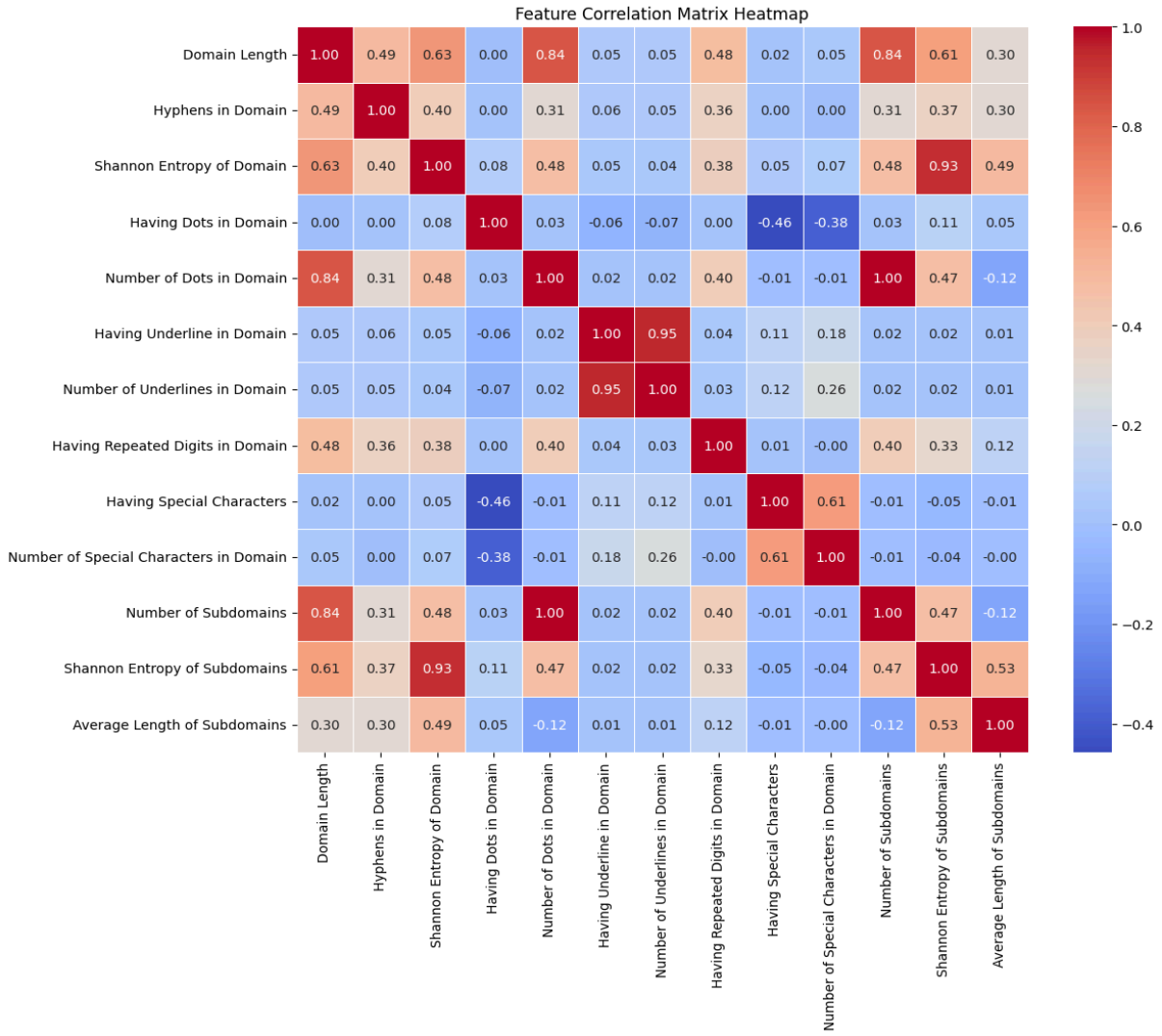


Figure 3.5 Correlation between the features

2.4. Selected Supervised Machine Learning Algorithms

This study employed six widely-used supervised machine learning algorithms to identify the optimal model for predicting phishing domains: Random Forest, Extra Trees, Decision Tree, Gaussian Naive Bayes, Stochastic Gradient Descent Classifier (SGDClassifier), and Multilayer Perceptron (MLP), each with distinct working principles and mathematical foundations. For instance, Random Forest is an ensemble learning technique that builds multiple decision trees using different subsets of data and features. The final prediction is determined by majority voting from all the trees. Mathematically, the prediction is represented as:

$$\hat{y} = \text{mode}(T_1(x), T_2(x), \dots, T_n(x))$$

Where, $T_i(x)$ is the prediction from the i-th decision tree.

On the other hand, Extra Trees, another ensemble method, operates similarly but introduces randomness in how the data is split at each node by choosing thresholds randomly instead of optimizing for the best split. Like Random Forest, the final prediction is based on majority voting:

$$\hat{y} = \text{mode}(T_1(x), T_2(x), \dots, T_n(x))$$

In contrast, Decision Tree works by recursively splitting data into subsets based on feature values that maximize information gain or minimize impurity. For classification, it uses measures like Gini impurity or entropy, with the following equations:

$$\text{Gini}(D) = 1 - \sum_{i=1}^n p_i^2 \quad \text{and} \quad \text{Entropy}(D) = - \sum_{i=1}^n p_i \log_2(p_i)$$

Where, $P_i(x)$ represents the proportion of class i in the dataset D

Gaussian Naive Bayes is based on Bayes' theorem and assumes that features are conditionally independent given the class. For continuous data, it assumes a Gaussian (normal) distribution. The posterior probability is calculated using:

$$P(C_k|x) = \frac{P(x|C_k)P(C_k)}{P(x)}$$

Where, $P(C_k)$ is the likelihood of the data given class C_k and $P(C_k)$ is the prior probability of the class.

SGDClassifier uses Stochastic Gradient Descent, an optimization algorithm that iteratively updates the model's weights based on small batches of data, making it scalable to large datasets. The weight update rule is expressed as:

$$w = w - \eta \nabla L(w)$$

where η is the learning rate, and $\nabla L(w)$ is the gradient of the loss function with respect to the weights.

Lastly, Multilayer Perceptron (MLP) is a type of neural network with one or more hidden layers. Each neuron in the network computes a weighted sum of the inputs, applies an activation function, and passes the result to the next layer. The output of a neuron can be represented as:

$$y = f \left(\sum_{i=1}^n w_i x_i + b \right)$$

Where f is the activation function, w_i are the weights, x_i are the inputs, and b is the bias term.

These algorithms, with their varied approaches to handling classification problems, were chosen for their ability to manage the complexities involved in phishing domain detection. Table 3.2 presents the key hyperparameters settings that this study considered.

Table 3.2. Detailed of the machine learning algorithms

ML Family	Algorithms	Key Hyperparameters
Ensemble Methods	Random Forest	n_estimators=100, *, criterion='gini', max_features='sqrt'
	Extra Trees	n_estimators=100, *, criterion='gini', min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features='sqrt'
Decision Trees	Decision Tree	splitter='best', criterion='gini', max_depth=None, min_samples_leaf=1, min_samples_split=2
Naive Bayes	Gaussian Naive Bayes	var_smoothing=1e-09, *, priors=None
Linear Models	SGDClassifier (Stochastic Gradient Descent)	tol=0.001, loss='hinge', penalty='l2', *, alpha=0.0001, fit_intercept=True, l1_ratio=0.15, max_iter=1000,
Neural Networks	Multilayer Perceptron (MLP)	activation='relu', hidden_layer_sizes=(100,), *, learning_rate='constant', solver='adam', batch_size='auto'

2.5. Evaluation Metrics

To evaluate the effectiveness of supervised machine learning (SML) classifiers and identify the optimal model, this study utilized six evaluation metrics: confusion matrix, accuracy, precision, recall, F1-score, and ROC curve. Initially, confusion matrices were created for each classifier, which is a common tool for summarizing the performance of a classification model. This matrix displays the correct and incorrect predictions using four key terms: true positive (TP), true negative (TN), false positive (FP), and false negative (FN). Following this, accuracy was calculated to determine the overall effectiveness of each classifier using the following formula:

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

However, since accuracy can be deceptive when dealing with imbalanced datasets, additional metrics were considered. Precision was used to measure the proportion of true positive predictions out of all positive predictions, calculated as:

$$Precision (P) = \frac{TP}{TP+FP}$$

This metric indicates the classifier's ability to avoid false positives. Recall, in contrast, measures the proportion of actual positive cases that were correctly identified by the classifier:

$$Recall (R) = \frac{TP}{TP+FN}$$

To account for the balance between precision and recall, the F1-score was computed, which represents the harmonic mean of the two:

$$F1\text{-score} (F1) = 2 \frac{P \cdot R}{P+R}$$

Finally, the study employed the receiver operating characteristic (ROC) curve and precision recall curve to evaluate the classifiers' diagnostic performance.

2.6. Final Model Deployment

Considering the performance of the selected classifiers, the most effective machine learning model was identified and recommended for deployment in a production environment to

help mitigate phishing attacks, such as through an Anti-Phishing Tool, Browser Extension, or Security Recommendation Tool.

Chapter 4 – Experimental Results

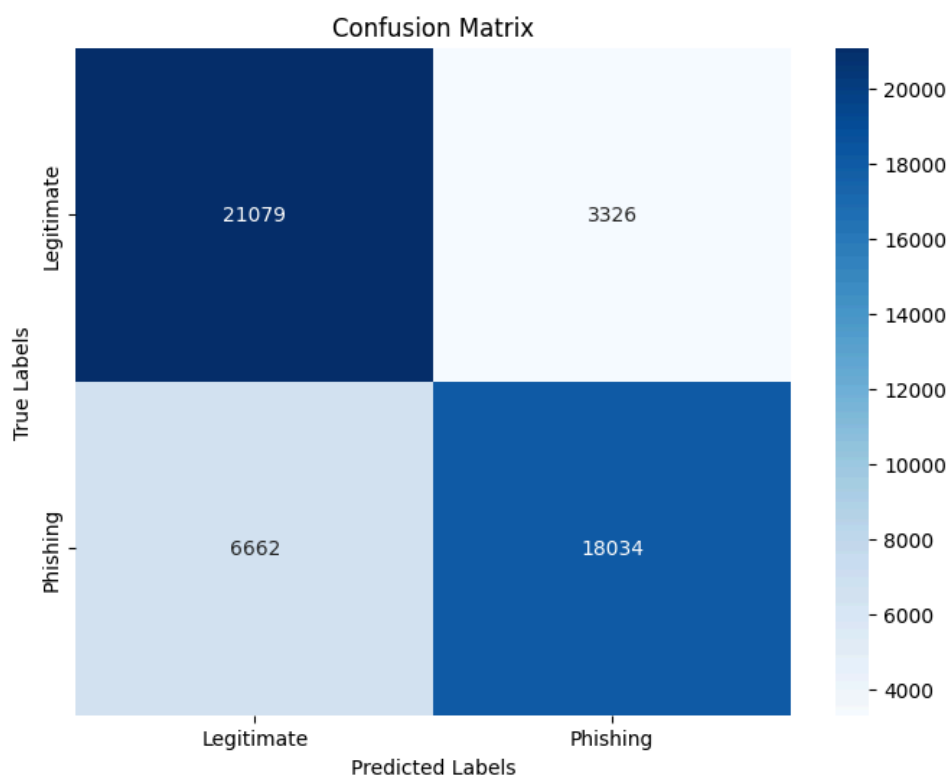
Phishing attacks remain a significant threat in the digital age, leveraging social engineering to trick unsuspecting victims into surrendering sensitive data. Unlike other targeted cyberattacks, phishing exhibits remarkable heterogeneity – impacting a wide range of targets with diverse motivations and goals. This dynamism, coupled with constantly evolving and sophisticated tactics, makes phishing detection a critical challenge for both individual users and security professionals alike. Consequently, phishing has become one of the most organized, challenging, and difficult-to-detect cyber threats of the 21st century. In light of this, this thesis aimed to address this need by proposing the development of an efficient, lightweight, data-driven, domain-based model for phishing detection utilizing state-of-the-art machine learning algorithms. Hence, this study utilized and compared 6 machine learning algorithms. The details performance of the machine learning algorithms is presented below:

4.1. Random Forest Classifier

The Random Forest Classifier performed relatively well, with an overall accuracy of 80.3% (See Table 4.1). It achieved a precision of 76% for classifying legitimate (0) instances and 84.4% for phishing (1) instances. The recall was higher for legitimate domains (86.4%) compared to phishing domains (73.0%), resulting in F1-scores of 80.8% and 78.3% for legitimate and phishing instances, respectively. The macro and weighted averages for precision, recall, and F1-score are all around 79.6-80.2%, indicating balanced performance across both classes. The training time was 33.03 seconds, and testing took 1.9 seconds, reflecting a moderate computational load. In addition to that, the Random Forest classifier demonstrates strong performance with a high number of true positives (18,034) and true negatives (21,079). This indicates that the model is effective at correctly identifying both positive and negative classes. However, it also has a moderate number of false negatives (6,662), which suggests that while it performs well overall, it still misses a notable number of positive cases (See Figure 4.1).

Table 4.1. Classification report of Random Forest Classifier

Class	Precision	Recall	F1-Score	Support	Training Time	Testing Time
0	76%	86.4%	80.8%	24405	33.03 seconds	1.9 seconds
1	84.4 %	73.0%	78.3%	24696		
5-fold CV Accuracy	80.2%					
Accuracy	80.3%			49101		
Macro avg	80.2%	79.7%	79.6%	49101		
Weighted avg	80.2%	79.7%	79.6%	49101		

**Figure 4.1** Confusion matrix (Random Forest Classifier)

4.2. Decision Tree Classifier

The Decision Tree Classifier yielded slightly lower performance compared to the Random Forest Classifier, with an overall accuracy of 79.0%. The precision for legitimate (0) and phishing (1) classifications were 74.7% and 84.8%, respectively. Recall was higher for legitimate instances at 87.1% but lower for phishing instances at 70.9%. This led to F1-scores

of 80.4% for legitimate emails and 77.2% for phishing emails. The macro and weighted averages were around 78.8-79.8%, showing a slight dip in the classifier's consistency. Training and testing times were significantly faster, at 1.6 seconds and 0.05 seconds, respectively. On the other hand, Decision Tree classifier shows slightly lower performance compared to Random Forest, with 17,514 true positives and 21,256 true negatives. Its false negative count is higher at 7,182, indicating that the Decision Tree misses more positive cases, which could affect its reliability in correctly identifying the positive class (See Figure 4.2).

Table 4.2. Classification report of Decision Tree Classifier

Class	Precision	Recall	F1-Score	Support	Training Time	Testing Time
0	747%	871%	804%	24405	1.6 seconds	0.05 seconds
1	848 %	709%	772%	24696		
5-fold CV Accuracy	78.9%					
Accuracy	79.0 %			49101		
Macro avg	79.8 %	79.0 %	78.8 %	49101		
Weighted avg	79.8 %	79.0 %	78.8 %	49101		

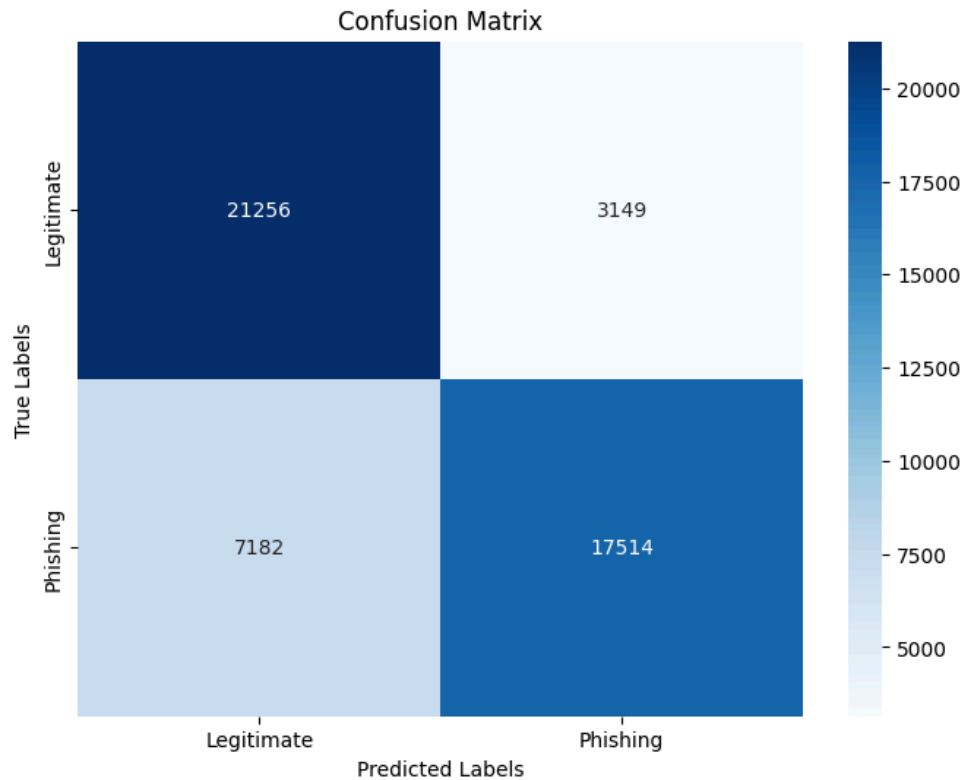


Figure 4.2 Confusion matrix (Decision Tree Classifier)

4.3. Gaussian Naive Bayes

The Gaussian Naive Bayes classifier showed poor performance, with an overall accuracy of 58.8%. The precision for legitimate (0) instances was 54.2%, while it was 93.1% for phishing (1) instances (see Table 4.3). However, recall was highly imbalanced, with 98.7% for legitimate emails and only 17.7% for phishing emails, resulting in an F1-score of 70.0% for legitimate emails but a very low 30.0% for phishing. The macro and weighted averages were also low, around 49.8-74.9%, indicating a substantial disparity in class performance. Training and testing times were extremely fast at 0.09 seconds and 0.01 seconds, respectively. In addition, the Gaussian Naive Bayes classifier struggles significantly with identifying the positive class, as evidenced by its low true positive count (4,376) and extremely high false negative rate (20,320). However, it excels in minimizing false positives, with only 322 instances, making it very conservative in predicting the positive class (See Figure 4.3).

Table 4.3. Classification report of Gaussian Naive Bayes

Class	Precision	Recall	F1-Score	Support	Training Time	Testing Time
Legitimate	0.542	0.987	0.700	24405	0.09	0.01
Phishing	0.931	0.177	0.300	25884	0.01	0.01

0	54.2%	98.7%	70.0%	24405	0.09 seconds	0.01 seconds
1	93.1%	17.7%	30.0%	24696		
5-fold CV Accuracy	59.01%					
Accuracy	58.8 %			49101		
Macro avg	73.7 %	58.2 %	749.9 %	49101		
Weighted avg	73.8%	58.0%	49.8%	49101		

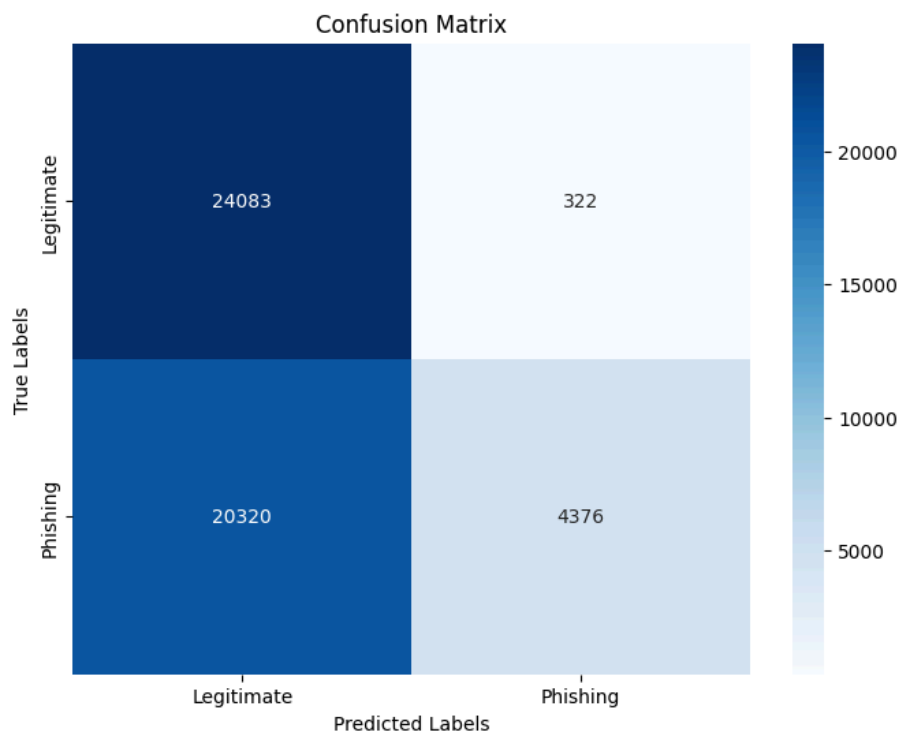


Figure 4.3 Confusion matrix (Gaussian Naive Bayes)

4.4. SGDClassifier

The SGDClassifier provided a moderate performance, with an overall accuracy of 69.7%. It showed a precision of 69.2% for legitimate (0) instances and 70.3% for phishing (1) instances. The recall was 70.5% for legitimate emails and 69.0% for phishing emails, resulting in F1-scores of 69.8% and 69.6%, respectively (see Table 4.4). The macro and weighted averages were consistently at 69.7%, indicating balanced performance across both classes. The classifier had moderate training time at 3.98 seconds, while testing was almost instantaneous at 0.005 seconds. In addition, SGDClassifier has a high false positive rate (7,206) and a significant false negative rate (7,665), indicating that it struggles with both

incorrectly identifying the positive class and missing positive cases. This makes it less reliable for tasks where precise classification is crucial (See Figure 4.4).

Table 4.4. Classification report of SGDClassifier

Class	Precision	Recall	F1-Score	Support	Training Time	Testing Time
0	69.2%	70.5 %	69.8%	24405	3.98 seconds	0.005 seconds
1	70.3%	69.0%	69.6%	24696		
5-fold CV Accuracy	67.4%					
Accuracy	69.7%			49101		
Macro avg	69.7 %	69.7 %	69.7 %	49101		
Weighted avg	69.7 %	69.7 %	69.7 %	49101		

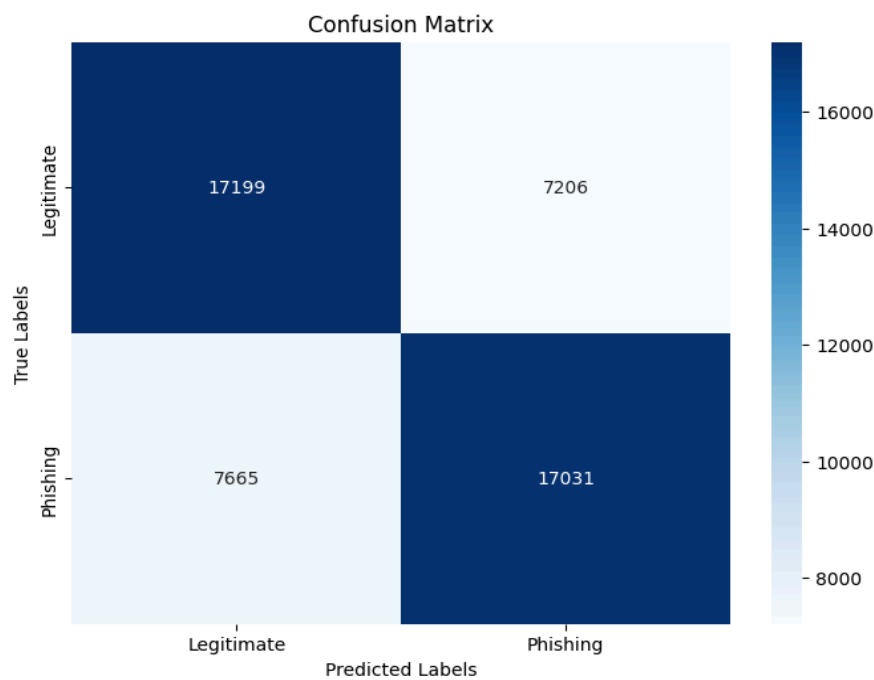


Figure 4.4 Confusion matrix (SGDClassifier)

4.5. Extra Trees Classifier

The Extra Trees Classifier demonstrated good performance with an overall accuracy of 80.0%. It achieved a precision of 75.9% for legitimate (0) instances and 80.5% for phishing (1) instances. The recall was 87.3% for legitimate emails and 79.9% for phishing emails, resulting in F1-scores of 81.2% and 79.8%, respectively. The macro and weighted averages

were both around 69.7%, suggesting consistent performance across both classes. However, training time was longer at 18.66 seconds, while testing took 1.902 seconds (see Table 4.5). Apart from this, Extra Trees classifier shows a strong performance, similar to Random Forest, with 17,915 true positives and 21,300 true negatives. Its false negative count is slightly higher at 6,781, but overall, it maintains a good balance between true positives and true negatives, making it a reliable classifier (See Figure 4.5).

Table 4.5 Classification report of Extra Trees Classifier

Class	Precision	Recall	F1-Score	Support	Training Time	Testing Time
0	75.9%	87.3 %	81.2%	24405	18.66 seconds	1.902 seconds
1	80.5%	79.9%	79.8%	24696		
5-fold CV Accuracy	80%					
Accuracy	80%			49101		
Macro avg	69.7 %	69.7 %	69.7 %	49101		
Weighted avg	69.7 %	69.7 %	69.7 %	49101		

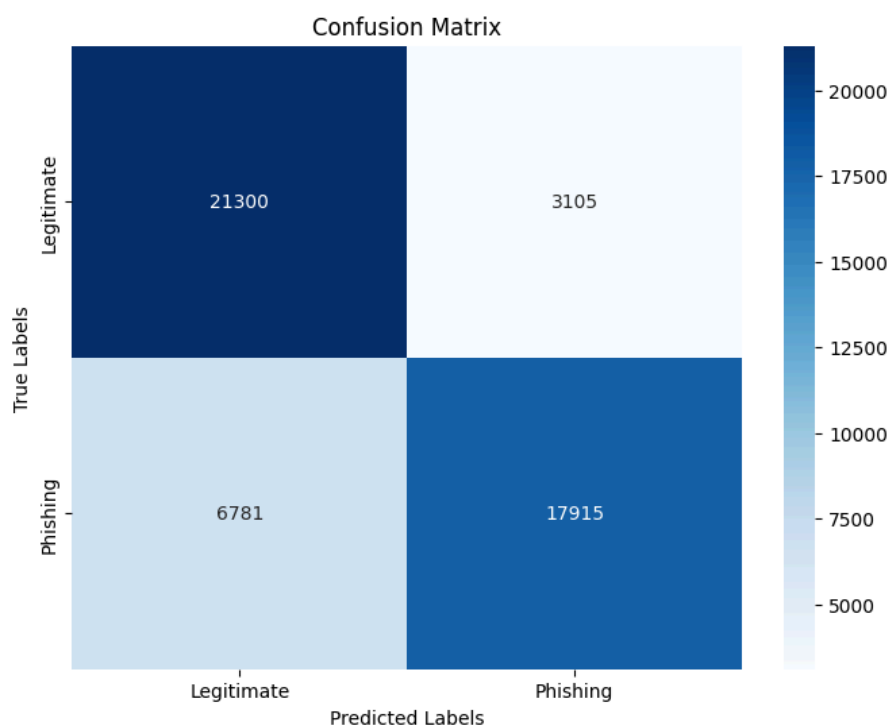


Figure 4.5 Confusion matrix (Extra Trees Classifier)

4.6. Multilayer Perceptron (MLP)

The Multilayer Perceptron (MLP) classifier showed moderate performance with an overall accuracy of 72.5%. It achieved a precision of 69.1% for legitimate (0) instances and 77.2% for phishing (1) instances. The recall was 80.8% for legitimate emails and 64.3% for phishing emails, resulting in F1-scores of 74.5% and 70.1%, respectively. The macro and weighted averages were slightly below 70%, indicating that the classifier performed better on legitimate emails. Training time was 3.3 seconds, while testing was very fast at 0.005 seconds (Table 4.6). MLP (Multi-Layer Perceptron) classifier performs moderately well with 15,876 true positives and 19,710 true negatives. However, it also has a significant number of false negatives (8,820), indicating that while it performs well overall, it may still miss a notable number of positive cases (See Figure 4.6).

Table 4.6 Classification report of Logistic Regression

Class	Precision	Recall	F1-Score	Support	Training Time	Testing Time
0	69.1%	80.8%	74.5%	24405	3.3 seconds	0.005 seconds
1	77.2%	64.3%	70.1%	24696		
5-fold CV Accuracy	72.5%					
Accuracy	72.5%			49101		
Macro avg	70.5%	69.8%	69.4%	49101		
Weighted avg	70.6%	69.7%	69.4%	49101		

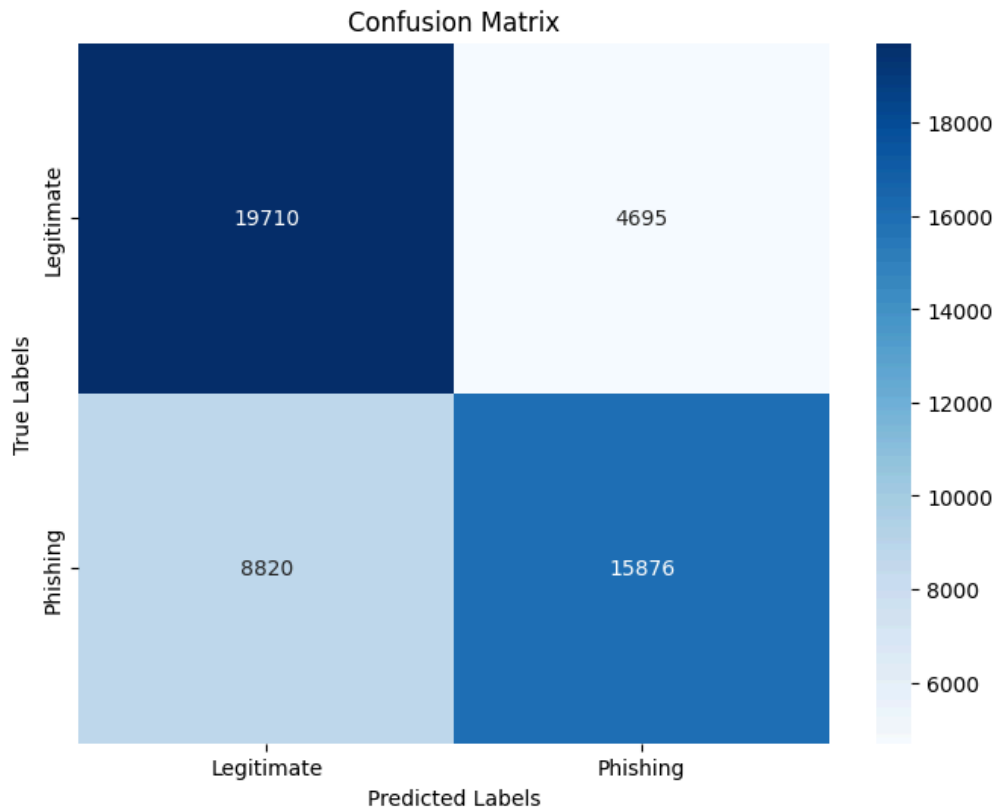


Figure 4.6 Confusion matrix (MLP)

4.7. ROC Curve analysis

The ROC (Receiver Operating Characteristic) Curve shown in Figure 4.7 illustrates the performance of multiple binary classifiers by plotting the True Positive Rate (TPR) against the False Positive Rate (FPR) at various threshold settings. Each dotted line represents a different classifier, and the legend indicates the classifier names alongside their respective Area Under the Curve (AUC) scores, which quantify overall performance. Higher AUC values suggest better discriminative ability, with the ideal AUC being 1.0. In this graph, the Random Forest and Extra Trees Classifiers achieve the highest AUC scores of 0.80, indicating they are the most effective at distinguishing between positive and negative classes. The Decision Tree follows closely with an AUC of 0.79, while the MLP Classifier performs moderately well with an AUC of 0.73. The SGDClassifier achieves a lower AUC of 0.70, suggesting it is less effective in comparison. In contrast, Gaussian Naive Bayes has an AUC of 0.58, indicating much lower performance, with a limited ability to differentiate between classes. No classifiers in this graph perform as poorly as random guessing, which would correspond to an AUC of 0.50. The variation in these AUC scores highlights the differing effectiveness of the classifiers, with

Random Forest and Extra Trees being the most reliable models for the task, while Gaussian Naive Bayes struggles to provide accurate predictions.

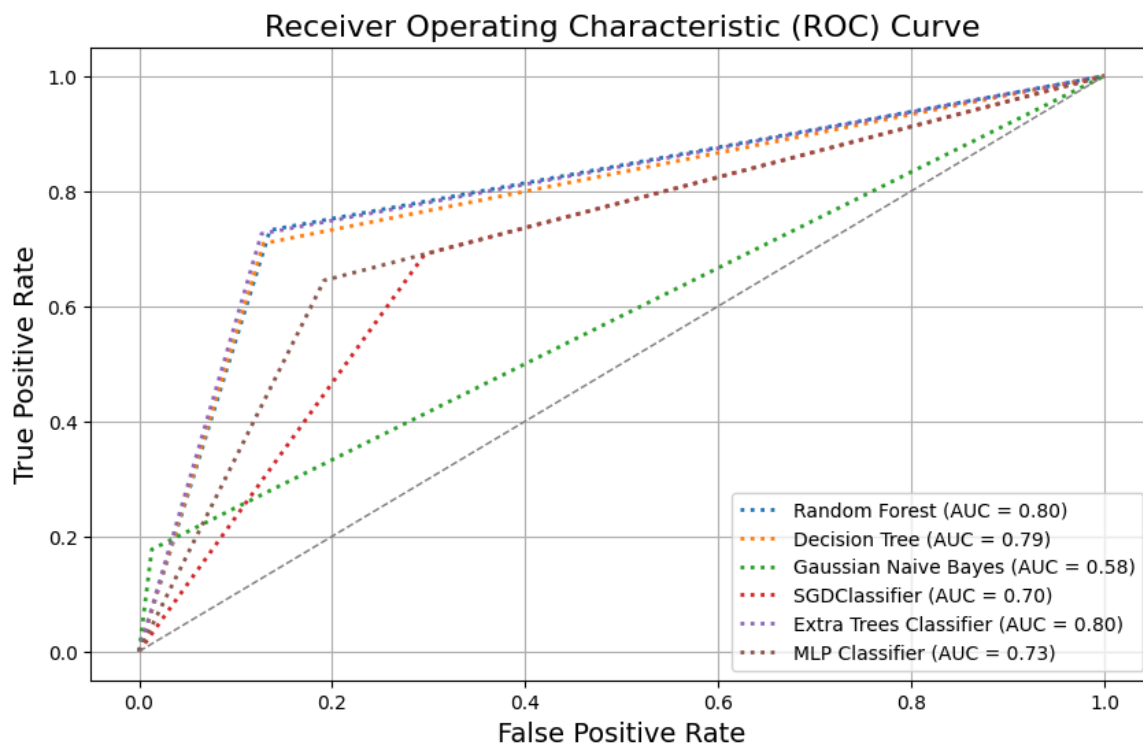


Figure 4.7. ROC Curve Analysis

4.8. Precision-recall Curve analysis

The Precision-Recall (PR) curve provided illustrates the performance of six different classifiers: Random Forest, Decision Tree, Gaussian Naive Bayes, SGDClassifier, Extra Trees Classifier, and MLP Classifier (see Figure 4.8). Precision, displayed on the y-axis, reflects the accuracy of the positive predictions made by each model, while recall, shown on the x-axis, indicates the model's ability to capture all relevant positive instances. Among the classifiers, Random Forest and Extra Trees perform the best, as their curves stay closer to the top-right corner of the graph, which represents a favorable balance between precision and recall. Decision Tree and MLP Classifier show moderate performance, with their precision starting to drop more significantly as recall increases. The SGDClassifier has a lower curve, reflecting struggles in maintaining precision as recall rises, and the Gaussian Naive Bayes classifier shows the poorest performance, with its curve being the farthest from the ideal top-right corner. Overall, Random Forest and Extra Trees maintain high precision across different

levels of recall, making them the best-performing models in this comparison, while the other models exhibit trade-offs between precision and recall to varying degrees.

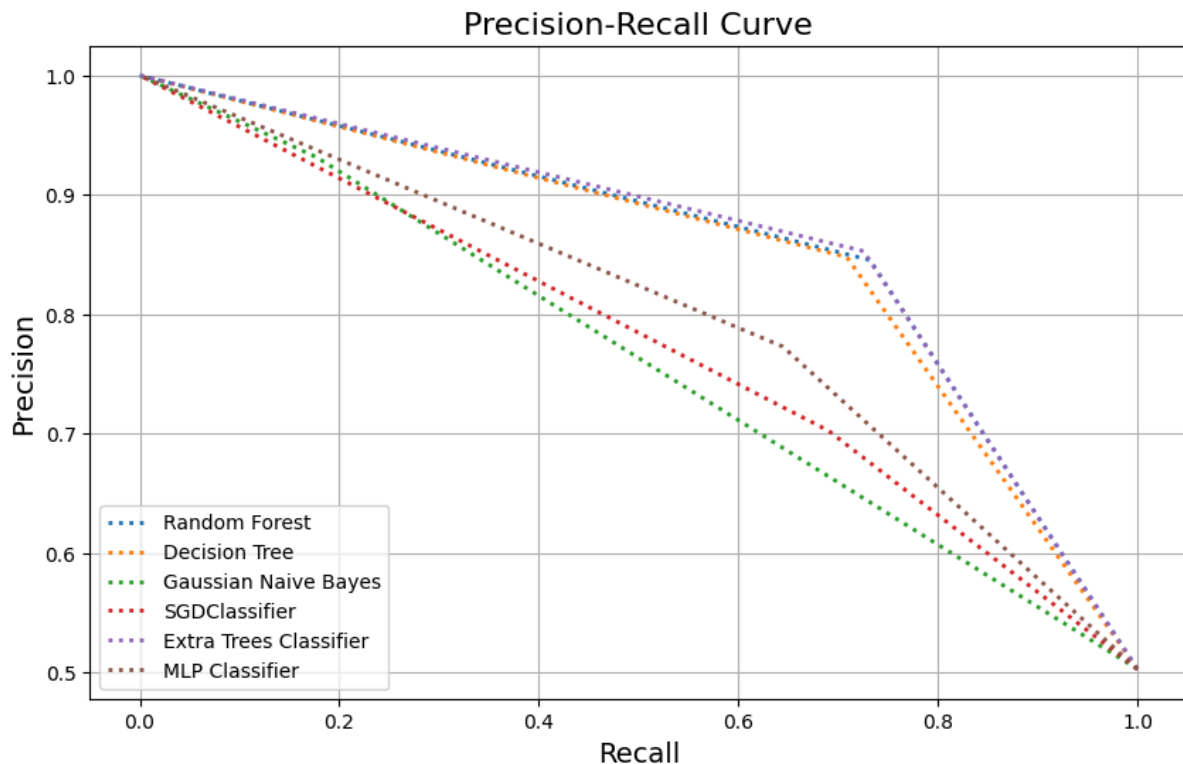


Figure 4.8 Precision-Recall Curve Analysis

4.9. Performance Comparison Among the Classifiers

The classifiers exhibit varying performance profiles, with the Random Forest and Extra Trees Classifiers standing out as the most reliable models. Both achieved high overall accuracies (80.3% and 80.0%, respectively) and demonstrated strong performance in terms of precision, recall, and F1-scores, particularly for phishing instances. The Random Forest had an edge with its high precision for phishing (84.4%) and balanced recall (73.0%), resulting in F1-scores of 80.8% and 78.3% for legitimate and phishing instances, respectively. The Extra Trees Classifier mirrored this performance with precision of 80.5% for phishing and recall of 79.9%, leading to similar F1-scores. Both classifiers also showed high true positive and true negative counts, though they struggled with a moderate number of false negatives. In contrast, the Decision Tree Classifier, while also performing well, showed slightly lower accuracy (79.0%) and had a higher false negative count (7,182), which impacts its reliability in identifying positive cases. The Gaussian Naive Bayes Classifier demonstrated the poorest performance with an overall accuracy of 58.8%, characterized by a significant imbalance in

precision and recall, especially for phishing instances, leading to a low AUC score of 0.58. Its low true positive count and high false negative rate further indicate its struggle with identifying phishing cases effectively. The SGDClassifier, with a moderate accuracy of 69.7%, exhibited balanced performance across precision and recall but had a high false positive and false negative rate, suggesting difficulties in classification reliability. The Multilayer Perceptron (MLP) achieved moderate performance with an accuracy of 72.5%, showing good precision for phishing but higher false negatives, highlighting its limitations in capturing all positive instances. The ROC and Precision-Recall Curve analyses reinforce these findings, with Random Forest and Extra Trees showing superior discriminative ability and balanced performance, while Gaussian Naive Bayes and SGDClassifier lag behind, struggling with precision-recall balance and overall effectiveness.

4.10. The final model (best performer)

Among the classifiers evaluated, the Random Forest Classifier emerges as the most effective model for phishing domain detection. It achieves the highest overall accuracy at 80.3%, demonstrating strong performance in distinguishing between legitimate and phishing domains. With precision scores of 76% for legitimate domains and 84.4% for phishing domains, and recall rates of 86.4% and 73.0%, respectively, the Random Forest classifier maintains a good balance between detecting true positives and minimizing false positives and negatives. This balance is further highlighted by its F1-scores of 80.8% for legitimate and 78.3% for phishing instances. Additionally, its high AUC score of 0.80 in ROC analysis indicates superior discriminative ability compared to other models, and its precision-recall curve remains closest to the ideal top-right corner, reflecting high precision across varying levels of recall. While its training and testing times are moderate, they are reasonable given its performance. In comparison, other classifiers such as the Decision Tree and Extra Trees Classifiers show slightly lower accuracy or higher false negative rates, and models like Gaussian Naive Bayes and SGDClassifier exhibit poorer overall performance with significant limitations. Therefore, the Random Forest Classifier stands out as the best model due to its robust performance across various metrics, making it the most reliable choice for accurately identifying phishing domains.

Chapter 5 – Discussion

3.1. Introduction

Phishing attacks have emerged as one of the most organized, challenging, and difficult-to-detect cyber threats in the 21st century. The heterogeneity and ever-evolving tactics of phishing make detection a critical challenge. In addressing this issue, this study aimed to develop an efficient, lightweight, data-driven, domain-based predictive model for phishing detection using state-of-the-art machine learning (ML) algorithms. The study compared six supervised ML algorithms—Random Forest (RF), Decision Tree (DT), Gaussian Naive Bayes (GNB), SGDClassifier (SGD), Extra Trees (ET), and Multilayer Perceptron (MLP)—with a large primary dataset and several evaluation metrics, offering valuable insights into the strengths and weaknesses of each algorithm in the context of phishing detection.

3.2. Comparison with Existing Studies

The Random Forest Classifier (RF) demonstrated strong performance with an overall accuracy of 80.3%, balanced precision and recall, and relatively low false positive and false negative rates. Its ability to effectively identify both legitimate and phishing domains,

coupled with a moderate computational load, makes it a reliable classifier. However, the notable number of false negatives (6,662) indicates that further optimization is required to improve the model's ability to correctly identify all phishing cases. The Extra Trees (ET) classifier exhibited similar results to the Random Forest, with a slightly lower number of false negatives and a balanced performance across both classes, confirming that ensemble-based methods are well-suited for phishing detection.

In contrast, the Decision Tree (DT) classifier, while fast in terms of training and testing times, showed slightly lower performance with a higher number of false negatives (7,182). This highlights its relative inefficiency in comparison to ensemble methods like RF and ET. The Gaussian Naive Bayes (GNB) classifier, with an accuracy of 58.8%, was the poorest performer among the six algorithms. The high precision for phishing instances (93.1%) was overshadowed by an imbalanced recall, especially for phishing domains (17.7%), which resulted in a low F1-score for phishing detection (30.0%). This shows that GNB is not suitable for phishing detection tasks where false negatives need to be minimized.

The Multilayer Perceptron (MLP) classifier and the SGDClassifier (SGD) exhibited moderate performance, with overall accuracies of 72.5% and 69.7%, respectively. Although both classifiers performed well in terms of processing speed, they struggled with a significant number of false positives and false negatives, making them less reliable for accurate phishing detection. These results suggest that while neural network-based models like MLP have potential, further optimization is required to enhance their practical applicability. Comparing the results of this study with previous works provides further insight into the strengths and limitations of various phishing detection methods. For example, the hybrid approach used by Korkmaz et al. (2022), which combined decision trees, random forests, and neural networks, achieved a notably high accuracy rate of 98.37%. However, their study utilized a smaller, imbalanced secondary dataset, which may have influenced the performance metrics. In contrast, the current study employed a large primary dataset and utilized a variety of performance metrics, including precision, recall, and F1-score, to provide a more comprehensive evaluation of the algorithms. The use of 5-fold cross-validation further ensured the reliability of the results, addressing limitations in some previous studies where cross-validation was not employed.

Similarly, Barraclough, Fehringer, and Woodward (2021) developed a methodology using hybrid approaches (e.g., ANFIS and heuristic-based techniques), achieving high accuracy. However, their approach did not account for computational complexity, an important factor for practical, real-time applications. The current study, by contrast, explicitly evaluated training and testing times for each algorithm, providing valuable insights into their computational efficiency. This is crucial for real-time phishing detection systems, where low latency is essential. In studies like Mughaid et al. (2022), which focused on practical applications of ML-based phishing detection, a small dataset size was a recurring limitation. By using a large, diverse dataset, the current study offers more generalizable results, particularly for URL domain-based phishing detection. However, it is worth noting that despite the robustness of the dataset and the use of multiple ML algorithms, the accuracy in this study was not as high as in some prior works, such as Korkmaz et al. (2022) or Dutta (2021). This discrepancy could be attributed to the specific characteristics of the dataset and the algorithms used, as well as the complexity of phishing detection in a real-world scenario.

Overall, this study's approach, leveraging a domain-based predictive model and multiple ML classifiers, demonstrated a balanced trade-off between accuracy and computational efficiency. While some algorithms (such as RF and ET) proved to be effective and reliable, others (like GNB) were less suitable for this task. These findings provide valuable contributions to the field of phishing detection by offering a more holistic evaluation of ML algorithms, addressing the limitations of prior studies in terms of dataset size, computational complexity, and evaluation metrics. Future work could focus on improving the performance of neural network-based models and exploring hybrid approaches to further enhance phishing detection accuracy and efficiency. The detailed comparison is presented in table 5.1.

Table 5.1. Comparison with the existing studies

Refere nce	Study Objective	Algorith s Used	Strength and Weakness	Type of Approa ch

(Korkmaz et al., 2022)	To improve the accuracy of phishing detection systems while minimizing the number of false positives.	6 algorithms (DT, RF, ANN, LSTM, DNN, GAN)	<ol style="list-style-type: none"> 1. High accuracy rate (98.37%) on a realistic dataset. 2. Improved detection system by minimizing false positives. 3. Dataset size is comparatively small and it was a secondary dataset. 4. Dataset was imbalanced 	Hybrid Approach (URL and Content based)
(Barracough, Fehringer and Woodward, 2021)	To develop a novel methodology for detecting phishing websites.	5 algorithms ANFIS, NB, PART, J48, JRip	<ol style="list-style-type: none"> 1. High accuracy 2. Dataset size is comparatively small. 2. Dataset was imbalanced and was not handled properly. 3. Computational complexity was not mentioned 	Hybrid Approach (blacklist, web content-heuristic-based)
(Mughaid et al., 2022)	To explore machine learning techniques to detect phishing attacks, particularly phishing emails, which have become increasingly prevalent.	7 algorithms (SVM, BDT, LR, AP, NN, DF)	<ol style="list-style-type: none"> 1. High accuracy rates for phishing detection. 2. Emphasis on practical application with real-world data. 3. Dataset size small. 4. Computational complexity was not mentioned. 	ML-based approach
(Dutta, 2021)	To propose and evaluate a URL detection technique based on machine learning approaches, specifically using a recurrent neural network (RNN) method, to identify phishing URLs	RNN—LSTM	<ol style="list-style-type: none"> 1. The proposed method outperforms recent approaches in malicious URL detection. 2. The details on computational efficiency, scalability, and real-time application are not provided. 	ML-based approach

(Jain and Gupta, 2017)	To provide a comprehensive analysis of phishing attacks and their exploitation, with a focus on recent visual similarity-based approaches for phishing detection	Not mentioned	<p>1. Utilizes visual similarity-based techniques, which can be effective in detecting phishing websites that closely resemble legitimate ones.</p> <p>2. Comprehensive analysis of recent approaches, providing a better understanding of the current solution space.</p> <p>3. The details on computational efficiency, scalability, and real-time application are not provided</p>	Visual similarity-based
houq Alnemari and Majid Alshammari, 2023)	To develop and compare four machine learning models for detecting phishing domains.	4 algorithms ANN SVM, DT, RF	<p>1. The study employed robust cross-validation methods (5-fold and 10-fold) to ensure the reliability of the results</p> <p>2. The accuracy was high Different evaluation metrics were used to find out the best model.</p> <p>3. Used secondary dataset</p> <p>4. The dataset was small</p>	ML-based
(Tamal et al., 2024)	To develop a more robust, effective, sophisticated, and reliable solution for phishing detection through optimal feature vectorization and supervised machine learning classifiers.	Optimal Feature Vectorization Algorithm (OFVA) for feature extraction	<p>1. High accuracy, precision, and AUC achieved Comprehensive evaluation of multiple algorithms.</p> <p>2. The study might have faced challenges related to the complexity of hyperparameter tuning and the computational cost associated with evaluating multiple classifiers.</p>	ML-based

Proposed Approach	To design and develop of an efficient, lightweight, data-driven, URL domain-based predictive model for phishing detection utilizing state-of-the-art supervised machine learning algorithms	6 supervised ML algorithms (RF, DT, GNB, SGD, ET, and MLP)	<ol style="list-style-type: none"> 1. Used large primary dataset 2. Used 6 ML algorithms from different ML families 3. Used many performance evaluation metrics to evaluate and compare ML algorithms 4. 5-fold cross-validation was utilized 5. A new feature generation process was followed 6. However, accuracy was not too good 	ML and Domain-based approach
-------------------	---	--	--	------------------------------

Chapter 5 – Conclusion

Phishing has become an increasingly sophisticated and pervasive threat in the digital age, posing significant risks to individuals and organizations alike. This form of cyber-attack involves deceptive attempts to acquire sensitive information, such as login credentials and financial details, by masquerading as a trustworthy entity. The consequences of successful phishing attacks can be severe, including financial loss, identity theft, and unauthorized access to confidential data. Unlike other targeted cyberattacks, phishing exhibits remarkable heterogeneity – impacting a wide range of targets with diverse motivations and goals. This dynamism, coupled with constantly evolving and sophisticated tactics, makes phishing detection a critical challenge for both individual users and security professionals alike. Consequently, phishing has become one of the most organized, challenging, and difficult-to-detect cyber threats of the 21st century. Unfortunately, most current detection techniques are static and inflexible, relying heavily on predefined patterns and rules. This rigidity stands in stark contrast to the ever-changing strategies employed by phishers. This significant disparity between the dynamic nature of phishing attempts and the static nature of existing detection methods underscores the urgent need for innovative strategies to combat this growing threat. In light of this, this thesis addressed this need by proposing the

development of an efficient, lightweight, data-driven, domain-based model for phishing detection utilizing state-of-the-art machine learning algorithms. The study's findings revealed that the Random Forest Classifier, Extra Trees Classifier, and Decision Tree Classifier were the most effective models. Among these, the Random Forest Classifier emerged as the best-performing model, achieving an overall accuracy of 80.3% with balanced precision and recall metrics. It demonstrated a strong ability to detect both legitimate and phishing emails, reflected by high F1-scores for both classes, while maintaining reasonable training and testing times. This balance between detection accuracy and computational efficiency makes Random Forest the most reliable model for phishing detection in this study. The Extra Trees Classifier also delivered robust performance, closely mirroring the Random Forest Classifier in accuracy and overall effectiveness. While its training time was longer, its efficient testing time and high precision and recall rates suggest that it remains a strong candidate when longer training times are acceptable. Though the Decision Tree Classifier exhibited slightly lower performance than the top two models, it still proved effective with a high detection rate, especially in environments with limited computational resources, owing to its rapid training and testing times. However, it struggled with a higher number of false negatives, indicating a need for improvement in detecting phishing cases. Overall, the study demonstrated that ensemble classifiers like Random Forest and Extra Trees consistently outperformed other models, offering a well-balanced solution for phishing detection. These results underscore the importance of using dynamic, data-driven approaches to effectively counter the ever-changing tactics employed in phishing attacks. Future research can focus on refining these models to further enhance detection rates and address the challenges of real-time phishing detection in various contexts

References

- APWG and Phishing Activity Trends Reports (2023). Apwg.org. Available online: <https://apwg.org/trendsreports> (16/07/2024).
- Adewole, K.S., Akintola, A.G., Shakirat Aderonke Salihu, Насир Фарук and Rasheed Gbenga Jimoh (2019). Hybrid Rule-Based Model for Phishing URLs Detection. Springer eBooks, pp.119–135. doi: https://doi.org/10.1007/978-3-030-23943-5_9
- Alkhalil, Z., Hewage, C., Nawaf, L. and Khan, I. (2021) “*Phishing Attacks: A Recent Comprehensive Study and a New Anatomy*,” *Frontiers in Computer Science*, vol. 3 [Online]. DOI: <https://doi.org/10.3389/fcomp.2021.563060>
- Al-Haija, Q.A. and Badawi, A.A. (2021). URL-based Phishing Websites Detection via Machine Learning. 2021 International Conference on Data Analytics for Business and Industry (ICDABI). doi: <https://doi.org/10.1109/icdabi53623.2021.9655851>
- Ardi, C. and Heidemann, J. (2016). AuntieTuna: Personalized Content-based Phishing Detection. Proceedings 2016 Workshop on Usable Security. [online] doi: <https://doi.org/10.14722/usec.2016.23012>
- Barraclough, P.A., Fehringer, G. and Woodward, J. (2021). Intelligent cyber-phishing detection for online. *Computers & Security*, 104, p.102123. doi: <https://doi.org/10.1016/j.cose.2020.102123>
- Chen, J.-L., Ma, Y.-W. and Huang, K.-L. (2020). Intelligent Visual Similarity-Based Phishing Websites Detection. *Symmetry*, 12(10), p.1681. doi: <https://doi.org/10.3390/sym12101681>
- Dutta, A.K. (2021). Detecting phishing websites using machine learning technique. *PLOS ONE*, [online] 16(10), p.e0258361. doi: <https://doi.org/10.1371/journal.pone.0258361>

- Hambali, M.A., Oladele, T.O. and Adewole, K.S. (2020). Microarray cancer feature selection: Review, challenges and research directions. *International Journal of Cognitive Computing in Engineering*, 1, pp.78–97. doi: <https://doi.org/10.1016/j.ijcce.2020.11.001>
- Jain, A.K. and Gupta, B.B. (2017). Phishing Detection: Analysis of Visual Similarity Based Approaches. *Security and Communication Networks*, [online] 2017, pp.1–20. doi: <https://doi.org/10.1155/2017/5421046>
- Jain, A.K., Parashar, S., Katare, P. and Sharma, I. (2020). PhishSKaPe: A Content based Approach to Escape Phishing Attacks. *Procedia Computer Science*, 171, pp.1102–1109. doi: <https://doi.org/10.1016/j.procs.2020.04.118>
- Jeeva, S.C. and Rajsingh, E.B. (2016). Intelligent phishing url detection using association rule mining. *Human-centric Computing and Information Sciences*, 6(1). doi: <https://doi.org/10.1186/s13673-016-0064-3>
- Jha, A.K., Muthalagu, R. and Pawar, P.M. (2023). Intelligent phishing website detection using machine learning. *Multimedia Tools and Applications*. doi: <https://doi.org/10.1007/s11042-023-14731-4>
- Kumar, M., Cheemaladinne Kondaiah, Alwyn Roshan Pais and Routhu Srinivasa Rao (2023). Machine learning models for phishing detection from TLS traffic. *Cluster Computing*, 26(5), pp.3263–3277. doi: <https://doi.org/10.1007/s10586-023-04042-6>
- Korkmaz, M., Kocyigit, E., Sahingoz, O.K. and Diri, B. (2022). A Hybrid Phishing Detection System Using Deep Learning-based URL and Content Analysis. *Elektronika ir Elektrotechnika*, 28(5), pp.80–89. doi: <https://doi.org/10.5755/j02.eie.31197>
- Klimburg-Witjes, N. and Wentland, A. (2021) “*Hacking Humans? Social Engineering and the Construction of the ‘Deficient User’ in Cybersecurity Discourses*,” *Science, Technology & Human Values/Science, Technology, & Human Values*, vol. 46, no. 6, pp. 1316–1339 [Online]. DOI: <https://doi.org/10.1177/0162243921992844>
- Mughaid, A., AlZu’bi, S., Hnaif, A., Taamneh, S., Alnajjar, A. and Elsoud, E.A. (2022). An intelligent cyber security phishing detection system using deep learning techniques. *Cluster Computing*, 25. doi:<https://doi.org/10.1007/s10586-022-03604-4>.
- Mishra, S. and Soni, D. (2019). A Content-Based Approach for Detecting Smishing in Mobile Environment. *SSRN Electronic Journal*. doi: <https://doi.org/10.2139/ssrn.3356256>
- Nanda, M. and Goel, S. (2024). URL based phishing attack detection using BiLSTM-gated highway attention block convolutional neural network. *Multimedia Tools and Applications*. doi: <https://doi.org/10.1007/s11042-023-17993-0>
- Odeh, A., Keshta, I. and Abdelfattah, E. (2021). Machine Learning Techniques for Detection of Website Phishing: A Review for Promises and Challenges. 2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC). doi: <https://doi.org/10.1109/ccwc51732.2021.9375997>

www.openphish.com. (2024) OpenPhish - Phishing Intelligence. [online] Available at: <https://www.openphish.com/>

Pandey, P. and Mishra, N. (2023). Phish-Sight: a new approach for phishing detection using dominant colors on web pages and machine learning. doi: <https://doi.org/10.1007/s10207-023-00672-4>

Pereira, J. (2020). *The 2020 Official Annual Cybercrime Report - Herjavec Group* [Online]. Available online: <https://www.herjavecgroup.com/the-2019-official-annual-cybercrime-report/> (16/07/2024).

Sahingoz, O.K., Buber, E., Demir, O. and Diri, B. (2019). Machine learning based phishing detection from URLs. *Expert Systems with Applications*, 117, pp.345–357. doi: <https://doi.org/10.1016/j.eswa.2018.09.029>

Statista (2024). *Worldwide digital population 2024* [Online]. Available online: <https://www.statista.com/statistics/617136/digital-population-worldwide/> (16/07/2024).

Shouq Alnemari and Majid Alshammari (2023). Detecting Phishing Domains Using Machine Learning. *Applied sciences*, 13(8), pp.4649–4649. doi: <https://doi.org/10.3390/app13084649>

Tamal, M. A., Islam, M. K., Bhuiyan, T., Sattar, A. and Prince, N. U. (2024) “*Unveiling suspicious phishing attacks: enhancing detection with an optimal feature vectorization algorithm and supervised machine learning*,” *Frontiers in Computer Science*, vol. 6. DOI: <https://doi.org/10.3389/fcomp.2024.1428013>

Ubing, A.A., Kamilia, S., Abdullah, A., Jhanjhi, N. and Supramaniam, M. (2019). Phishing Website Detection: An Improved Accuracy through Feature Selection and Ensemble Learning. *International Journal of Advanced Computer Science and Applications*, 10(1). doi: <https://doi.org/10.14569/ijacsa.2019.0100133>

Vayansky, I. and Kumar, S. (2018). Phishing – challenges and solutions. *Computer Fraud & Security*, 2018(1), pp.15–20. doi: [https://doi.org/10.1016/s1361-3723\(18\)30007-1](https://doi.org/10.1016/s1361-3723(18)30007-1)

www.domcop.com. (2024). Buy Expired Domains: Moz, Majestic, SEMrush, Estibot, SimilarWeb & more. [online] Available at: <https://www.domcop.com>