

**Universi****ty of** **South Wales**

**Prifysgol** **De Cymru**

**Faculty of Computing, Engineering and Science**

**MSc Project:** “Enhancing Web Security: Both Ensemble and Deep Learning Approach to Fake Website Detection”

**Name: Md Rafiqul Islam**

**Enrollment ID: 30104104**

**Supervisor: Dr Carl Jones**

**Year of Study:** 2023-2024

**Course: MSc in 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 close up of a signature

Description automatically generated

(Student)

# Abstract

In today's digital age, the internet has become an integral part of daily life, providing convenience and accessibility. However, this has also led to the rise of cybercrime, particularly phishing attacks. These attacks exploit the trust of internet users by mimicking legitimate websites to steal sensitive information. Despite advancements in phishing detection, current methods struggle to keep pace with the evolving nature of these attacks, often leading to false positives or missed detections. This thesis investigates the effectiveness of ensemble and deep learning in identifying phishing websites using URL-specific features. By leveraging a large dataset and combining machine learning and deep learning techniques, the study develops a robust and accurate phishing detection tool. The Random Forest classifier emerged as the best-performing model, with an accuracy of 96.6%, precision of 97.2%, recall of 95.7%, and an AUC of 97%. These metrics highlight its ability to generalize well across phishing and legitimate cases, making it ideal for real-world applications.

# Acknowledgements

First and foremost, I am immensely grateful to Almighty Allah for His boundless blessings, which have enabled me to complete this thesis successfully.

I would also like to extend my deepest appreciation to my thesis supervisor, Dr Carl Jones, Senior Lecturer University Of South Wales . I consider myself incredibly fortunate to have benefited from his unwavering support, patience, and guidance throughout this journey. His constant encouragement, insightful feedback, and constructive criticism were invaluable, helping to shape both my research and personal growth. I especially appreciate the freedom he gave me to explore my own path, while always being there to provide mentorship and steer me in the right direction. His innovative thinking and profound research vision have been a true source of inspiration, and without his support, this work would not have been possible.

Finally, I owe an immeasurable debt of gratitude to my family, especially my beloved wife. Her unwavering love, patience, and encouragement have been my greatest source of strength during the ups and downs of this process. She sacrificed countless sleepless nights and moments of joy, standing by my side and offering constant reassurance when I faced challenges. Her belief in me, even when I doubted myself, was a beacon that kept me moving forward. Without her emotional support and understanding, I would not have had the resilience to see this thesis through to completion. The balance she helped me maintain between personal and abroad life allowed me to stay focused and driven. This achievement is as much hers as it is mine.

**Table of Contents**

[Declaration 2](#_Toc177333786)

[Abstract 3](#_Toc177333787)

[Acknowledgements 4](#_Toc177333788)

[Chapter 1 – Introduction 9](#_Toc177333789)

[1.1. Introduction 9](#_Toc177333790)

[1.2. Understanding Phishing 9](#_Toc177333791)

[1.3. Types of Phishing Attacks 10](#_Toc177333792)

[1.4. Phishing Life Cycle 11](#_Toc177333793)

[1.5. Most-Targeted Industry Sectors by Phishing 12](#_Toc177333794)

[1.6. Thesis Aim and Objectives 15](#_Toc177333795)

[1.7. Thesis Outline 15](#_Toc177333796)

[Chapter 2 – Literature Review 16](#_Toc177333797)

[2.1. Introduction 16](#_Toc177333800)

[2.2. Literature Sources 17](#_Toc177333801)

[2.3. Key Review Questions 17](#_Toc177333802)

[2.4. Literature Search Strategy 17](#_Toc177333803)

[2.5. Eligibility Criteria of the Literatures 18](#_Toc177333804)

[2.6. Review Findings 18](#_Toc177333805)

[A. What Is Phishing and How It Works? 18](#_Toc177333806)

[B. Current State-Of-The-Art Techniques in Phishing and Malicious Website Detection 19](#_Toc177333807)

[C. Existing Research Gaps 22](#_Toc177333808)

[Chapter 3 – Materials and Methods 23](#_Toc177333809)

[3.1. Introduction 23](#_Toc177333813)

[3.2. Experimental Setup 23](#_Toc177333814)

[3.3. Experimental Data Acquisition 23](#_Toc177333815)

[3.4. Data Preprocessing 24](#_Toc177333816)

[3.5. Feature Selection 25](#_Toc177333817)

[3.6. Selection of the Ensemble Methods 26](#_Toc177333818)

[3.7. Final Model Selection 27](#_Toc177333819)

[Chapter 3 – Experimental Results 29](#_Toc177333820)

[4.1. Introduction 29](#_Toc177333825)

[4.2. Performance of Random Forest Classifier (RF) 29](#_Toc177333826)

[4.3. Performance of Gradient Boosting Classifier (GB) 32](#_Toc177333827)

[4.4. Performance of AdaBoost Classifier (AB) 35](#_Toc177333828)

[4.5. Performance of Voting Classifier (VC) 38](#_Toc177333829)

[4.6. Performance of Deep Neural Network (DNN) 41](#_Toc177333830)

[4.7. Performance of Multi-layer Perceptron (MLP) 46](#_Toc177333831)

[4.8. Best Model: Random Forest Classifier 49](#_Toc177333832)

[Chapter 4 – Discussion 51](#_Toc177333833)

[Chapter 5 – Conclusion 57](#_Toc177333834)

**List of Tables**

|  |  |
| --- | --- |
| **Title** | **Page** |
| Table 1.1 Most-Targeted Industry Sectors by Phishing 2015- 2016 | 13 |
| Table 1.2 Most-Targeted Industry Sectors by Phishing 2017- 2018 | 14 |
| Table 1.3 Most-Targeted Industry Sectors by Phishing 2019- 2020 | 14 |
| Table 1.4 Most-Targeted Industry Sectors by Phishing 2021-2022 | 14 |
| Table 2.1. Search keywords | 17 |
| Table 4.1. Detailed performance of Random Forest Classifier | 30 |
| Table 4.2. Detailed performance of Gradient Boosting Classifier | 33 |
| Table 4.3 Detailed performance of Random AdaBoost Classifier | 36 |
| Table 4.4 Detailed performance of Random Voting Classifier | 39 |
| Table 4.5 Detailed performance of Deep Neural Network (DNN) | 42 |
| Table 4.6 Detailed performance of Multi-layer Perceptron (MLP) | 46 |
| Table 5.1 Comparison of the proposed approach with previous studies | 53 |

**List of Figures**

|  |  |
| --- | --- |
| **Title** | **Page** |
| Figure 1.1 Steps of phishing attacks | 12 |
| Figure 2.1 PRISMA flow diagram | 16 |
| Figure 3.1 Class balance (before preprocessing) | 24 |
| Figure 3.2 Class balance (after preprocessing) | 24 |
| Figure 3.3 Feature importance | 26 |
| Figure 4.1 Confusion matrix (RF) | 30 |
| Figure 4.2 ROC Curve (Random Forest Classifier) | 31 |
| Figure 4.3 Precision-recall Curve (Random Forest Classifier) | 31 |
| Figure 4.4 Confusion matrix (GB) | 33 |
| Figure 4.5 ROC Curve (Gradient Boosting Classifier) | 34 |
| Figure 4.6 Precision-recall Curve (Gradient Boosting Classifier) | 34 |
| Figure 4.7 Confusion matrix (AB) | 36 |
| Figure 4.8 ROC Curve (AdaBoost Classifier) | 37 |
| Figure 4.9 Precision-recall Curve (AdaBoost Classifier) | 37 |
| Figure 4.10 Confusion matrix (MV) | 39 |
| Figure 4.11 ROC Curve (Majority Voting) | 40 |
| Figure 4.12 Precision-recall Curve (Majority Voting) | 40 |
| Figure 4.13 Confusion matrix (DNN) | 42 |
| Figure 4.14 Model Structure (DNN) | 43 |
| Figure 4.15 ROC Curve (DNN) | 44 |
| Figure 4.16 Precision-recall Curve (DNN) | 44 |
| Figure 4.17 Train-validation accuracy (DNN) | 45 |
| Figure 4.18 Train-validation Loss (DNN) | 45 |
| Figure 4.19 Confusion matrix (MLP) | 47 |
| Figure 4.20 ROC Curve (ROC) | 48 |
| Figure 4.21 Precision-recall Curve (ROC) | 49 |

# Chapter 1 – Introduction

## Introduction

In this current digital era, the rapid advancement of technology has transformed life, making it more virtual and accessible. This technological revolution has reshaped many traditional systems, leading people to spend more time online. The internet's exponential growth has not only altered daily lifestyles but also replaced various conventional practices. People increasingly share their thoughts, emotions, and daily activities online, dedicating a significant amount of time to the virtual world. Due to its simplicity, convenience, and time-saving nature, online financial transactions have also seen a swift rise. Today, the internet serves not only as a source of entertainment but also as a critical platform for business and financial transactions. Key technological trends now include Big Data, the Internet of Things, artificial intelligence, augmented reality, and automation, along with the growing concern of cybercrime. The impact of cybercrime is increasingly troubling for both individuals and businesses, as the nature of these crimes evolves with the expanding global economy.

Cybercrime or online fraud, an illegal and unethical activity for personal gain, remains prevalent in both the online and offline financial sectors. When such fraudulent activities escalate, they become a significant concern. Phishing is one such form of digital fraud, where a malicious website masquerades as a legitimate one to steal sensitive information like credit card numbers or bank account passwords. Utilizing social engineering tactics, phishing attacks continue to pose substantial risks and threats to internet users and online banking. Victims often fail to recognize these deceptive web pages because they closely resemble legitimate sites, leading to the unintentional sharing of critical information. This issue is rapidly growing in the online world today.

## Understanding Phishing

Phishing has been characterized in multiple ways by experts, researchers, and cybersecurity organizations due to its evolving nature and the varying contexts in which it occurs. As a result, there isn't a single, universally accepted definition for "phishing." However, a recent study suggests that the term "phishing" is derived from the word "fishing," where attackers use "bait" to lure victims and then "fish" for their confidential information or trick them into downloading malicious software. This tactic relies more on exploiting human vulnerabilities and psychological manipulation than on technical or coding-based methods, which is why phishing is often referred to as "human hacking". Phishing can be described as a cyber-enabled crime that employs social engineering and technical deception to deceive individuals into revealing sensitive information (such as credit card numbers, login credentials, or personal identification details) by impersonating a legitimate entity. Phishing attacks continue to pose significant risks to web users and online banking. Typically, victims fail to recognize fraudulent websites because they closely resemble authentic ones. Consequently, users unwittingly provide their critical information on these malicious sites, and this issue is rapidly escalating on the internet. Phishing undermines trust within business networks and has a detrimental effect on the future of e-commerce.

## Types of Phishing Attacks

There are many types of phishing **(Alkhalil et al., 2021;** **Safi and Singh, 2023)**. For instance, email phishing, the most common phishing attack, involves sending fraudulent emails that appear to come from reputable organizations, such as banks or online services, often creating a sense of urgency to lure recipients into clicking malicious links or providing sensitive information. Spear phishing, on the other hand, targets specific individuals or organizations, using personalized details like the victim's name or job position to make the attack more convincing. SMiShing, or SMS phishing, uses text messages to deceive victims, typically invoking urgency and including links to malicious sites or prompts for sensitive information. Vishing, or voice phishing, occurs via phone calls, where attackers impersonate representatives of banks or government agencies to extract personal or financial details. Pharming redirects victims to fake websites that mimic legitimate ones, tricking them into providing sensitive information, often through techniques like DNS cache poisoning or malware. Clone phishing involves creating a near-identical replica of a legitimate email or website, subtly altering elements like links to deceive users into divulging their credentials. Finally, whaling targets high-profile individuals such as CEOs or executives, employing sophisticated and personalized tactics to manipulate them into revealing sensitive information or authorizing financial transactions.

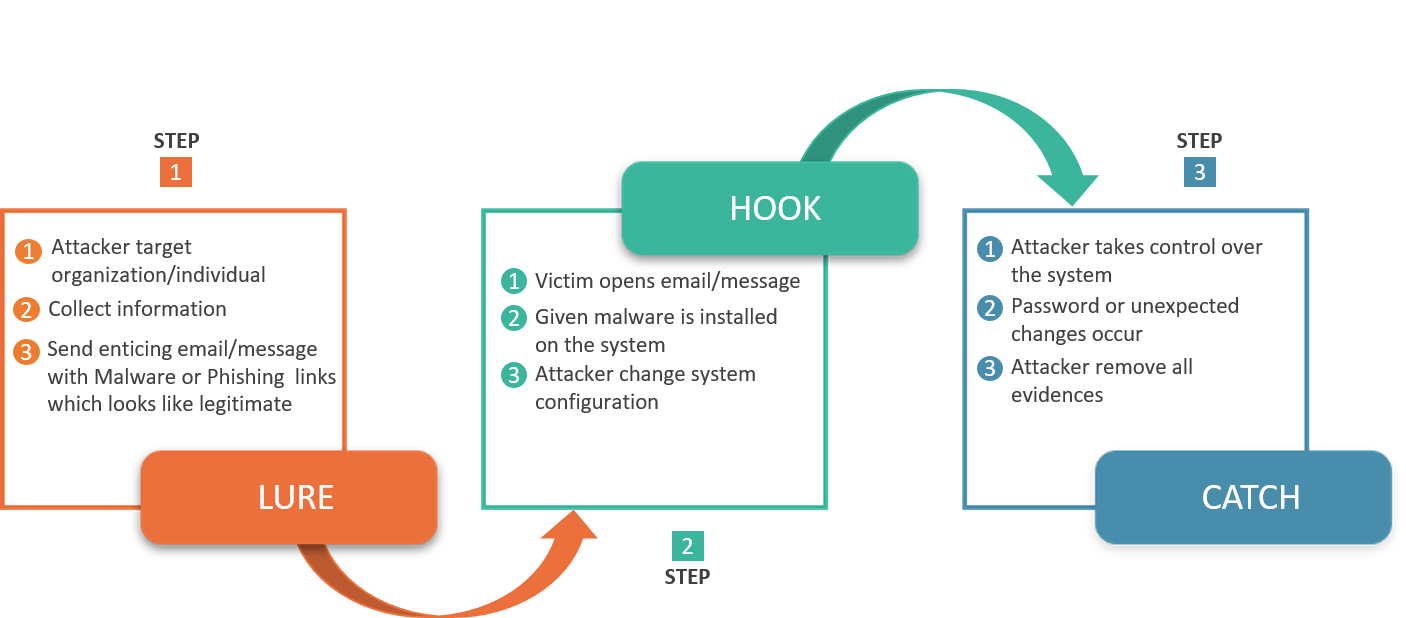
## Phishing Life Cycle

Phishing is a prominent form of social engineering, where attackers manipulate individuals through psychological tactics to gain unauthorized access to sensitive information or systems within an organization. These attacks are typically executed in a sequence of well-orchestrated stages, designed to gradually lure the victim into compromising their security. A typical phishing attack can be broken down into three critical phases: Lure, Hook, and Catch (Tamal et al., 2024).

Lure Phase: The attack begins with the reconnaissance or information-gathering phase, where the attacker meticulously researches the target organization or individual. This might involve scouring social media profiles, company websites, public records, or even conducting direct interactions under false pretenses to collect valuable data. The attacker may also use social engineering techniques to extract information from different sources within the organization, such as employees or partners, leveraging the information obtained from one source to gain the trust of another. This phase is crucial, as the success of the subsequent steps depends on the accuracy and relevance of the collected information.

Hook Phase: Armed with sufficient information, the attacker crafts a deceptive message, email, or website that closely mimics a legitimate source, making it difficult for the target to discern the threat. These communications are often designed to appear as routine business correspondence, customer service inquiries, or urgent security alerts, using branding, language, and formats that align with the target’s expectations. The message typically contains a call to action, such as clicking a link, downloading an attachment, or entering login credentials. The attacker’s goal in this phase is to entice the victim to interact with the malicious content, thereby initiating the attack. Advanced phishing campaigns may use sophisticated techniques such as spear-phishing, where the message is highly personalized based on the target's role, preferences, or recent activities, further increasing the likelihood of success.

Catch Phase: Once the victim engages with the phishing content, the attack escalates to the catch phase, where the attacker gains control over the target's system or data. This might involve the installation of malware, such as keyloggers, ransomware, or trojans, that can monitor the victim's activities, steal sensitive data, or provide the attacker with remote access to the system. Alternatively, the victim may be redirected to a fraudulent website designed to capture login credentials or payment information. At this stage, the attacker may also make immediate changes to the system, such as altering passwords, locking the victim out, or initiating unauthorized transactions. The attacker typically tries to cover their tracks by deleting logs, hiding malicious code, or removing traces of the phishing incident to avoid detection and prolong their access to the compromised system. Figure 1.1 illustrates the complete phishing life cycle, from the initial reconnaissance to the final exploitation and cover-up.



**Figure 1.1.** Steps of phishing attacks (Tamal et al., 2024).

## Most-Targeted Industry Sectors by Phishing

Phishing is a frequently changing cybersecurity threat that often target diverse industrial sectors. For instance, APWG’s phishing activity trend reports from 2015 to 2022 reveals a dynamic and evolving threat landscape, with significant shifts in the targeted industries over time. Between 2015 and 2016, the primary targets were Financial Institutions, Payment Services, and ISPs, consistently ranking high in phishing attempts. The Retail/Service sector, however, experienced a sharp increase in attacks towards the end of 2016, surpassing other sectors. During this period, Multimedia and Unclassified sectors also fluctuated, while sectors like Gaming, Government, Education, and Delivery remained minimally affected (see Table 1.1). Moving into 2017 and 2018, the Payment sector became the dominant target, maintaining consistently high percentages, overshadowing other industries. SAAS/Webmail and Financial Institutions also faced significant threats, but not to the same extent as the Payment sector. Meanwhile, sectors such as Government, Insurance, Travel, and Telecommunications saw minimal phishing activity, with some dropping off the list by 2018. The emergence of social media and Cloud Storage as targets during this period indicates a growing focus on digital platforms (see Table 1.2). From 2019 to 2020, SAAS/Webmail continued to be the leading target, though there was a slight decrease in the percentage of attacks by the end of 2020. The Payment sector, which had seen a dramatic rise in previous years, experienced a notable decline. Financial Institutions remained consistently targeted, showing a slight increase towards the end of 2020. Simultaneously, sectors like eCommerce/Retail and social media saw an uptick in phishing activity, reflecting the increasing threats in these areas. Additionally, Logistics emerged as a new target, particularly towards the end of 2020, likely driven by the rise of e-commerce (see Table 1.3). By 2021 and 2022, SAAS/Webmail and Financial Institutions continued to be significant targets, though their attack rates fluctuated across quarters. The once-dominant Payment sector saw a further decline, while eCommerce/Retail and social media remained targeted, albeit at a reduced rate compared to earlier years. Notably, the other categories witnessed a substantial rise, especially in 2022, indicating a diversification of phishing targets. The emergence of Cryptocurrency as a target during this period highlights the evolving nature of digital threats (see Table 1.4). Overall, it is clearly evident that phishing attacks have shifted from traditional financial targets to more diversified and digital-oriented sectors, reflecting the broader digitalization of services and the advent of new technologies.

**Table 1.1** Most-Targeted Industry Sectors by Phishing 2015- 2016

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Industry | 2015 | | | | 2016 | | | |
| 1st Quarter  (%) | 2nd Quarter  (%) | 3rd  Quarter  (%) | 4th Quarter  (%) | 1st  Quarter  (%) | 2nd  Quarter  (%) | 3rd  Quarter  (%) | 4th  Quarter  (%) |
| Multimedia | 5.23 | 7.07 | 13.0 | 8.63 | 3.30 | 3.0 | 4.0 | 5.15 |
| Unclassified | 4.42 | 7.34 | 10.4 | 7.06 | 3.13 | 5.0 | 5.0 | 4.30 |
| Gaming | 3.29 | 2.79 | 1.31 | 0.33 | 0.16 | 0.0 | 0.0 | 0.13 |
| Auction | 1.68 | 0.73 | 0.41 | 0.54 | 1.20 | 4.0 | 0.0 | 0.24 |
| Government | 1.08 | 3.48 | 0.91 | 1.46 | 1.64 | 1.0 | 1.0 | 1.31 |
| Education | 0.00 | 0.01 | 0.02 | 0.01 | 0.01 | 0.0 | 0.0 | 0.01 |
| Delivery | 0.17 | 0.14 | 0.10 | 0.12 | 0.07 | 0.0 | 0.0 | 0.07 |
| MM Reseller | 0.01 | 0.06 | 0.08 | 0.04 | 0.00 | 0.0 | 0.0 | 0.02 |
| Social | 0.97 | 1.49 | 3.59 | 2.62 | 2.22 | 2.0 | 2.0 | 3.32 |
| Classified | 0.20 | 0.18 | 0.09 | 0.15 | 0.14 | 0.0 | 0.0 | 0.08 |
| Financial | 20.9 | 23.6 | 20.4 | 20.4 | 18.6 | 16.0 | 21.0 | 19.6 |
| Payment | 21.8 | 16.5 | 14.9 | 16.0 | 14.7 | 13.0 | 10.0 | 11.3 |
| ISP | 26.2 | 25.3 | 24.3 | 18.4 | 12.0 | 12.0 | 12.0 | 12.5 |
| Retail/Service | 13.90 | 11.1 | 10.2 | 24.0 | 42.7 | 43.0 | 43.0 | 41.8 |

**Table** **1.2** Most-Targeted Industry Sectors by Phishing 2017- 2018

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Industry | 2017 | | | 2018 | | |
| 2nd Quarter  (%) | 3rd  Quarter  (%) | 4th  Quarter  (%) | 1st  Quarter  (%) | 2nd  Quarter  (%) | 3rd  Quarter  (%) |
| Payment | 45 | 45 | 45 | 39.4 | 36 | 38.2 |
| Government | 1.0 | 1.0 | 1.0 | 0.0 | 0 | 0.0 |
| Insurance | 1.0 | 1.0 | 1.0 | 0.0 | 0 | 0.0 |
| Travel | 1.0 | 1.0 | 1.0 | 0.0 | 0 | 0.0 |
| Online dating | 1.0 | 1.0 | 1.0 | 0.0 | 0 | 0.0 |
| Logistics/Shipping | 2.0 | 2.0 | 2.0 | 0.0 | 0 | 3.2 |
| Telecommunication | 2.0 | 2.0 | 2.0 | 0.0 | 0 | 0.0 |
| Social media | 2.0 | 2.0 | 2.0 | 0.0 | 4 | 3.1 |
| Other | 2.0 | 2.0 | 2.0 | 16.4 | 14 | 13.2 |
| Retail/E-commerce | 4.0 | 4.0 | 4.0 | 0.0 | 0 | 0.0 |
| Cloud storage | 9.0 | 9.0 | 9.0 | 11.3 | 9 | 6.5 |
| SAAS/Webmail | 15.0 | 15.0 | 15.0 | 18.7 | 21 | 20.1 |
| Financial Institution | 16.0 | 16.0 | 16.0 | 14.2 | 16 | 15.7 |
| Education | 0.0 | 0.0 | 0.0 | 0 | 0 | 0.0 |
| Manufacturing | 0.0 | 0.0 | 0.0 | 0.0 | 0 | 0.0 |

**Table 1.3** Most-Targeted Industry Sectors by Phishing 2019- 2020

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Industry | 2019 | | | | 2020 | | | |
| 1st Quarter  (%) | 2nd Quarter  (%) | 3rd  Quarter  (%) | 4th Quarter  (%) | 1st  Quarter  (%) | 2nd  Quarter  (%) | 3rd  Quarter  (%) | 4th  Quarter  (%) |
| SAAS /  Webmail | 36 | 36 | 33 | 30.8 | 33.5 | 34.7 | 31.4 | 22.2 |
| Payment | 27 | 22 | 21 | 19.8 | 13.3 | 11.8 | 13.4 | 15.2 |
| Financial  Institution | 16 | 18 | 19 | 19.4 | 19.4 | 18 | 19.2 | 22.5 |
| eCommerce /  Retail | 3 | 3 | 4 | 5.4 | 6.2 | 7.5 | 7.2 | 8.9 |
| Telecom | 3 | 3 | 0 | 3.3 | 0 | 0 | 3.2 | 2.5 |
| Other | 15 | 18 | 19 | 0 | 6.9 | 10.9 | 6.7 | 10.4 |
| Cloud storage | 0 | 0 | 4 | 3.4 | 3.9 | 2.9 | 2.1 | 0 |
| Social media | 0 | 0 | 0 | 3 | 8.3 | 10.8 | 12.6 | 11.8 |
| Logistics | 0 | 0 | 0 | 0 | 0 | 0 | 4.2 | 6.4 |

**Table 1.4** Most-Targeted Industry Sectors by Phishing 2021-2022

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Industry | 2021 | | | | 2022 | | |
| 1st Quarter  (%) | 2nd Quarter  (%) | 3rd  Quarter  (%) | 4th Quarter  (%) | 1st  Quarter  (%) | 2nd  Quarter  (%) | 3rd  Quarter  (%) |
| SAAS /  Webmail | 19.6 | 8.7 | 29.1 | 19.5 | 20.5 | 19.1 | 17 |
| Payment | 8.5 | 12.2 | 7.1 | 9.3 | 5 | 6.3 | 4 |
| Financial  Institution | 24.9 | 29.2 | 17.8 |  | 23.6 | 27.6 | 23 |
| eCommerce /  Retail | 7.6 | 8.2 | 13.1 | 17.3 | 14.6 | 5.6 | 4 |
| Telecom | 0 | 2.4 | 3.5 | 0 | 0 | 2.6 | 3 |
| Other | 8 |  | 9.3 | 11.6 | 13.4 | 14.7 | 30 |
| Social media | 23.6 | 14.8 | 11 | 8.5 | 12.5 | 15.3 | 11 |
| Logistics/ Shipping | 5.8 | 6.9 | 3.5 | 4.1 | 3.8 | 4.3 | 6 |
| Cryptocurrency | 2 | 7.5 | 0 | 6.5 | 0 | 4.5 | 2 |

## Thesis Aim and Objectives

Based on the above evidence, it is clear that phishing is being the most prevalent threat over the years and targeting diverse industrial sectors. The Anti-Phishing Working Group (APWG) reported a record 1,270,883 unique phishing attacks in Q3 2022, highlighting the limitations of current anti-phishing methods. Existing anti-phishing techniques struggle to keep pace with evolving phishing tactics. They often fail to detect new fake websites, zero-hour attacks or give false-positive alarm. To this end, this study wants to develop a rapid and accurate anti-malicious website detection tool. More specifically, this study will investigate the effectiveness of ensemble learning in identifying malicious websites based on URL-specific features. So, by leveraging machine learning (ensemble learning and deep learning), this research aims to bridge the gaps in traditional methods and contribute to more accessible and holistic data-driven model to safeguard website.

## Thesis Outline

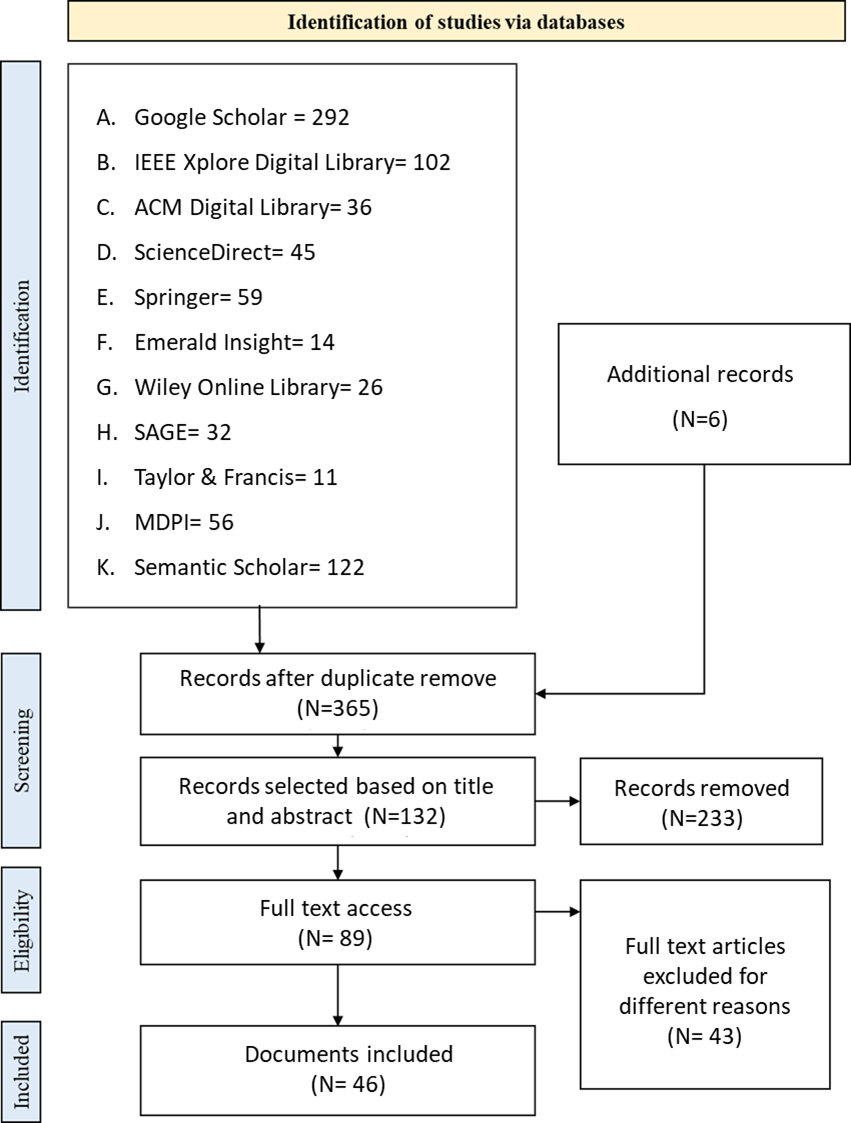
The remaining chapters of this thesis are organized as follows: Chapter 2 presents a systematic literature review using the PRISMA framework. Chapter 3 provides a detailed explanation of the methodology. In Chapter 4, the experimental findings are presented step by step. Chapter 5 offers a comprehensive discussion of these findings. Finally, Chapter 6 concludes the thesis.

# Chapter 2 – Literature Review



## Introduction

This section adheres to the PRISMA (Preferred Reporting Items for Systematic Reviews and Meta-Analyses) guidelines for conducting a systematic literature review (Page et al., 2021) of 46 articles on phishing detection, published in prominent security journals and conferences between 2015 and 2024. Figure 2.1 depicts the selection stages of these articles. The review initiated with an extensive analysis to offer a thorough understanding of the current state-of-the-art in phishing detection methods, both technical and non-technical, and to identify the existing challenges and shortcomings in this field.



**Figure 2.1** PRISMA flow diagram

## Literature Sources

To identify potential studies, the following databases were utilized: Google Scholar, IEEE Xplore Digital Library, ACM Digital Library, ScienceDirect, Springer, Emerald Insight, Wiley Online Library, SAGE, Taylor & Francis, MDPI, and Semantic Scholar.

## Key Review Questions

The two key questions addressed in this review are specified below:

1. What are the current state-of-the-art techniques in phishing and malicious website detection, and how effective are they in addressing the challenges and limitations in this field?
2. What are the strengths and weaknesses of these techniques?

## Literature Search Strategy

To answer the above review questions effectively, this study employed a comprehensive search strategy. A selected set of keywords was utilized to conduct a comprehensive literature review across prominent and widely recognized databases as presented in sub-section 2.2. Here, table 2.1 outlines the keywords used in the search strategy and their alignment with the review questions.

**Table 2.1** Search keywords

|  |  |  |
| --- | --- | --- |
| **SN.** | **Mapped Review Question** | **Keywords** |
| 1 | Core | “Phishing”, “Phishing attacks”,  “Phishing detection” |
| 2 | RQ1 | “Fake website detection technique”, “phishing detection methods”, “Anti-phishing methods”, “Countermeasures of phishing” |
| 3 | RQ2 | “Fake website characteristics”, “Phishing detection challenges”, “Phishing detection limitations” |

## Eligibility Criteria of the Literatures

In order to ensure the reliability and validity of our findings, a thorough quality assessment was conducted prior to including any articles in the review list. Firstly, this study only considers research articles, review articles, conference proceedings, book chapters, and reliable public reports. Secondly, documents were only considered that published between 2015 and 2024. Any document that deemed suspicious or did not meet the quality criteria (e.g., having clear objectives, valid methodology, organized findings and discussion) were promptly removed from consideration. The selection process was implemented to maintain the integrity and robustness of the review as much as possible (see Figure 2.2).

## 2.6. Review Findings

### A. What Is Phishing and How It Works?

Phishing has been described in various ways by experts, researchers, and cybersecurity organizations, given its continuous evolution and the different contexts in which it occurs. Consequently, there is no universally accepted definition of "phishing" (Alkhalil et al., 2021). However, it is widely understood as a cybercrime that leverages social engineering tactics and deceptive techniques to trick individuals into disclosing sensitive information by posing as a legitimate entity. Instead of relying heavily on technical approaches, attackers often target human weaknesses and use psychological manipulation, earning phishing the label "human hacking" (Klimburg-Witjes and Wentland, 2021). Most of the time, victims are unable to identify the malicious web pages as they seem almost legitimate. Phishers often attract their victims by using enticing offers or attention-grabbing phrases. As noted by Tamal et al. (2024), phishing attacks typically unfold in three stages: lure, hook, and catch. In the lure stage, attackers gather information about the target—whether an individual or an organization—and determine their method of attack. They then send seemingly legitimate messages, SMS, or QR codes containing phishing links, often accompanied by attractive deals or a sense of urgency. During the hook stage, the victim unknowingly opens the email, clicks on the link, or downloads malicious software. In the final catch stage, the victim loses control of their system, which may result in password theft, unauthorized transactions, or other unusual activities. The attacker then removes all evidence of the attack. Since most general users have nominal technological expertise, fraudsters take advantage of this to commit phishing. Though phishing is an old technique used to deceive individuals and organizations, effective defenses against phishing attacks are still lacking. As a result, the number of phishing and fake websites continues to grow. This indicates an exponential increase in phishing activities. Studies have suggested that the frequently changing nature of these attacks, along with dynamic and threatening strategies and the widespread availability of the internet, have made phishing attacks both frequent and devastating contexts (Vayansky and Kumar, 2018). Consequently, identifying phishing has become a major concern due to the sophisticated strategies employed by attackers.

### B. Current State-Of-The-Art Techniques in Phishing and Malicious Website Detection

To combat phishing attacks, two main strategies are generally followed by researchers or cyber-practitioners: (a) preventive measures and (b) detective measures. Preventive measures include campaigns, awareness training, seminars, notices etc. aimed at raising user awareness to prevent phishing. Various studies have demonstrated the effectiveness of this approach. For example, Quinkert et al. (2021) conducted an empirical study in a real workplace setting, the study analyzed over 420,000 phishing emails sent over 1.5 years by a consulting firm providing awareness training. The findings of this study revealed that a significant decrease in click rates, from 19% to 10%, as awareness training progresses. Additionally, it identified certain psychological tactics, like authoritative tones and appeals to curiosity, as particularly effective in phishing scams. In another study, the authors involved more than 20,000 employees from a large financial institution in Thailand and was conducted the test in three phases: (i) an initial phishing attack (simulation), a training phase using a mixed-approach (awareness training), and a subsequent phishing attack (after training session) with different content. As of the findings, a significant improvement in cybersecurity awareness was founded. For example, the number of employees who opened phishing emails dropped by 71.5%, which highlighted the effectiveness of the training approach (Daengsi et al., 2021). Furthermore, another study by Jensen et al. (2017) evaluated the effectiveness of mindfulness versus rule-based training in a field experiment at a U.S. university with 355 participants familiar with phishing. Both training programs were delivered in text-only and text-plus-graphics formats. After 10 days, a phishing attack was simulated using generic and customized messages. Results showed that participants who trained in mindfulness were more successful in avoiding phishing attacks. Notably, those who were confident in their detection abilities and those with low email mindfulness and low perceptions of Internet risk showed significant improvement.

On the other hand, the detective measures include mostly rule-based approaches (Moghimi and Varjani, 2016; Jain and Gupta, 2018; Mourtaji et al., 2021), content-based approaches (Jain and Gupta, 2017; Chiew et al., 2015; Ardi and Heidemann, 2016; Mishra and Soni, 2019;), heuristic-based approaches and ML-based approaches. In rule-based approach, phishing attempts have been detected by applying predefined rules or patterns. These rules are derived based on known characteristics and behaviors of phishing emails and websites. For example, in a study of Jain and Gupta (2018), a rule-based data mining classification approach was utilized to detect smishing messages. They identified nine rules that can differentiate between genuine and malicious SMS. They revealed that their approach demonstrated a true negative rate exceeding 99% and also proved effective in detecting zero-hour attacks. In another study, Mourtaji et al. (2021) utilized 37 features derived from six methods: blacklisted, lexical and host, content, identity, identity similarity, visual similarity, and behavioral. Later, a comparative analysis is conducted among various machine learning and deep learning models to develop a Hybrid Rule-Based Solution. The findings showed that the proposed method effectively analyzed URLs from multiple perspectives where the highest accuracy levels was achieved by the deep learning models, with the CNN model reaching 97.945%. In similar way, the study of Moghimi and Varjani (2016) showed that rule-based method can achieve a high accuracy of 99.14% true positives and a low false negative rate of 0.86%.

Another widely employed method for phishing detection is content-based analysis. This technique involves identifying phishing attempts by examining the content of emails or websites. Jain and Gupta (2017) conducted an in-depth study of phishing attacks and their mechanisms, focusing on recent visual similarity-based detection methods. Their research revealed that these techniques rely on various features, such as text content, text formatting, HTML tags, Cascading Style Sheets (CSS), images, and more, to detect phishing. In a study by Chiew et al (2015), the authors proposed a logo based novel phishing detection method. The detection method involved two main processes: logo extraction and identity verification. First, the logo image is extracted from all downloaded images on a webpage using machine learning techniques to ensure accurate identification. Next, the extracted logo is used in a Google image search to verify the portrayed identity. The findings revealed that their method yields reliable and promising results. However, they did not mention the exact accuracy of their proposed system. Another study 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 (Ardi and Heidemann, 2016). 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 a study (Mishra and Soni, 2019). Another past study revealed that this type of approach will not work for different languages (Jain and Gupta, 2017). Concurrently, some of the phishing websites contain only images. For those cases, this kind of approach won't work.

On the other hand, heuristic-based approach to combat phishing involves using heuristic rules or patterns to detect and prevent phishing attempts (Silva, Feitosa and Garcia, 2020). For instance, Almeida and Carlos Becker Westphall, (2020) proposed a method to detect phishing by checking specific strings in URLs and email messages, which can be used alongside proxies and Anti-Spam filters. In an experimental setup, this method achieved detection accuracy between 73.3% and 97.66%, with an average processing time of 30 seconds. Similarly, Rao and Ali (2015) introduced PhishShield, a desktop application based on heuristics, which demonstrated a detection accuracy of 96.57% with lower false negative and false positive rates. However, Heuristic-based phishing detection relies on predefined rules and patterns, which made it vulnerable to evolving threats. While effective against known attacks, it struggles with new phishing techniques, often leading to false positives and missed detections (Zieni, Massari and Calzarossa, 2023; Tamal et al., 2024). Among the various methods used for phishing detection, machine learning (ML)-based approaches have gained widespread adoption among researchers and security experts worldwide (Tamal et al., 2024). Treating phishing detection as a binary classification problem, both supervised ML algorithms (Lakshmi and Vijaya, 2012; Alam et al., 2020; Nataraj et al., 2023;) and deep learning techniques (Alshingiti et al., 2023; Saha et al., 2020;) are commonly employed. However, very few studies found that used ensemble methods.

### C. Existing Research Gaps

Despite advancements in phishing detection, several critical limitations persist. The rapidly evolving tactics of phishing attacks often outpace traditional rule-based and heuristic defenses, which struggle to adapt to novel, sophisticated threats. While content-based methods hold promise, their effectiveness is hindered by attackers' use of obfuscation, images, and dynamic web elements. Machine learning, though powerful, demands substantial training data and computational resources. Moreover, the quality and diversity of training datasets significantly impact model performance. Apart from this, though different ML algorithms were frequently used to combat phishing, ensemble methods were focused less. To this end, this research proposes investigating the effectiveness of ensemble learning in identifying phishing websites based on URL-specific features. By leveraging machine learning (ensemble learning), this research aims to bridge the gaps in traditional methods and contribute to more accessible and holistic data-driven model to safeguard website.

# Chapter 3 – Materials and Methods



## Introduction

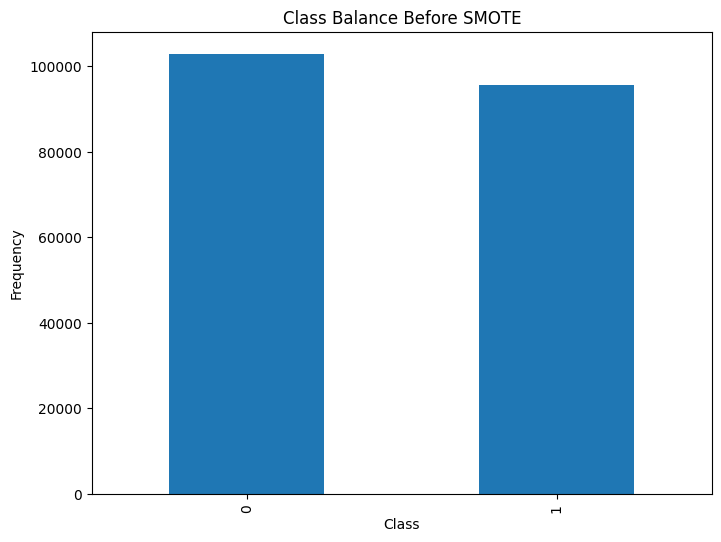
The Materials and Methods section of this study is divided into several subsections for clarity. Section 3.2 outlines the data acquisition procedures. Section 3.3 discusses data preprocessing, detailing how null values, class imbalance, and feature descriptions were addressed to eliminate conflicting data. Section 3.4 covers feature selection, highlighting the identification of the most efficient features. In Section 3.5, the selection and justification of the ensemble methods are presented. Finally, Section 3.6 explains the process of final model selection.

## Experimental Setup

The experiments for this study were conducted using a laptop running the Windows 10 Pro operating system, equipped with an Intel Core i5-6200U CPU (2.30 GHz), 64-bit architecture, x64-based processor, and 12 GB of RAM. Additionally, Google Colab, a cloud-based platform, was utilized to write and execute Python 3.0 code, leveraging its computational resources and integrated development environment to handle data processing and model training tasks.

## Experimental Data Acquisition

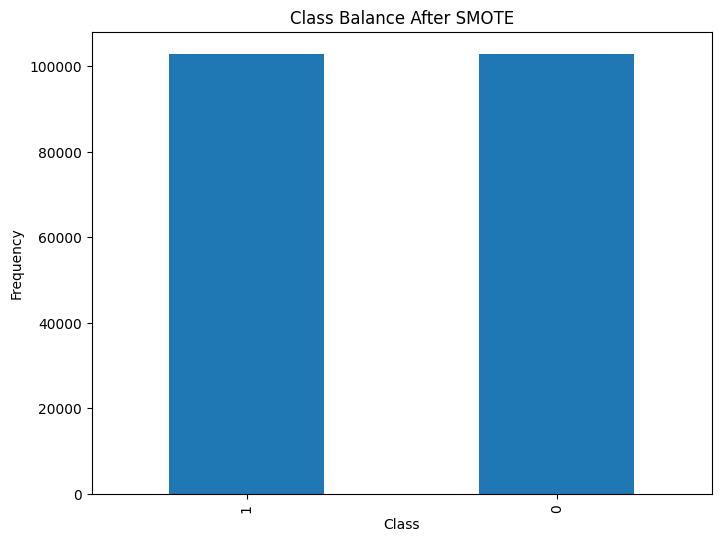
The dataset employed in this study was sourced from a data article (Zieni et al., 2023; Maruf Ahmed Tamal et al., 2024; Sattari and Montazer, 2023;), which provided a large-scale labeled dataset specifically tailored for detecting phishing based on URLs. This dataset includes 247,950 instances, with 128,541 labeled as phishing URLs and 119,409 as legitimate URLs. Rather than relying on content-based features such as text, messages, DOM, CSS, or logos, this dataset exclusively emphasizes intra-URL characteristics. This particular dataset was selected due to its relatively large size and its current relevance. Additionally, because this study addresses a binary classification problem (phishing vs. legitimate), it was crucial to have balanced classes, and this dataset is nearly balanced in that regard (see Figure 2.2).



**Figure 3.1** Class balance (before preprocessing)

## Data Preprocessing

As the dataset was already preprocessed, including the removal of duplicate URLs and outliers, I implemented additional methods to enhance data quality. Initially, I applied the Synthetic Minority Oversampling Technique (SMOTE) to address class imbalance within the dataset. SMOTE is a statistical method used in machine learning to balance datasets by generating synthetic samples for minority classes (How to deal with Imbalanced Datasets with SMOTE Algorithm, 2024). Additionally, I utilized the isnull() function of python panda library to check for any missing values (i.e., NaN values) in the dataset. Fortunately, no missing values were detected.



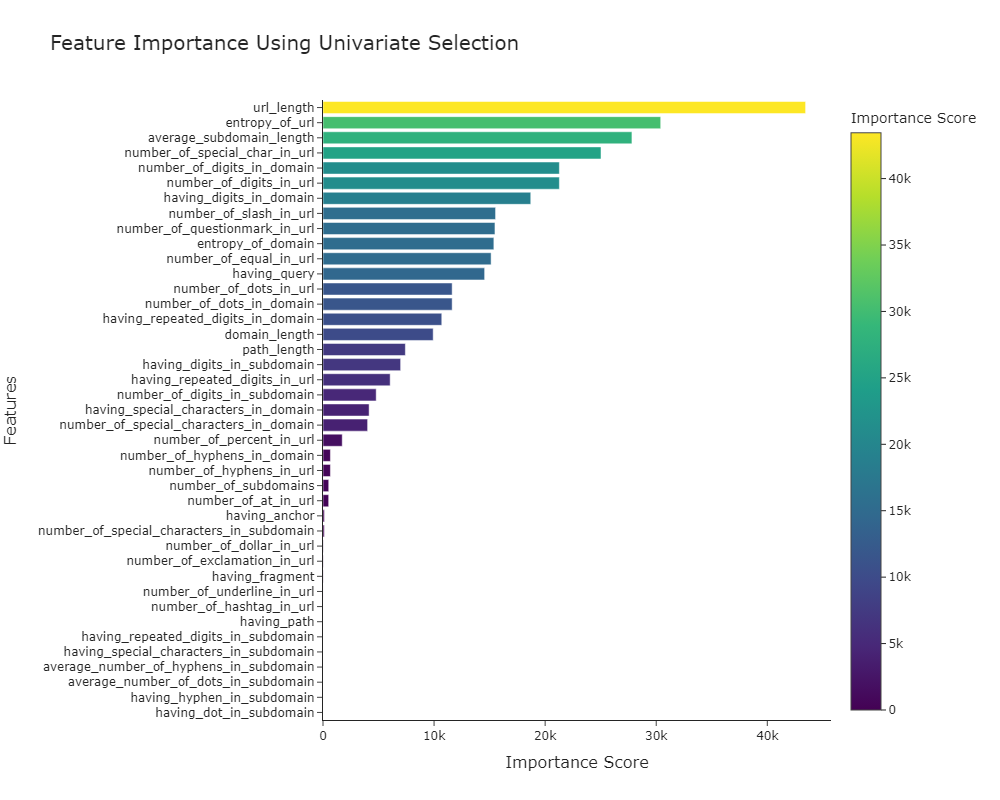
**Figure 3.2** Class balance (after preprocessing)

**3.3. Feature Description**

The dataset used in this study comprised 41 features and 1 target variable, designed to identify potentially malicious URLs by analyzing various components of the URL structure (Maruf Ahmed Tamal et al., 2024). The dataset begins with a Type feature (F0), a Boolean indicator that labels URLs as either legitimate (0) or phishing (1). Several numeric features analyze the URL's structure, such as URL length (F1), which measures the total number of characters in the URL, and number of dots in URL (F2), which counts the dots in the URL. The dataset also includes various other numeric features that count the presence of specific characters within the URL, such as number of digits in URL (F4), number of special char in URL (F5), number of hyphens in URL (F6), number of underline in URL (F7), number of slash in URL (F8), number of question mark in URL (F9), and others focusing on different special characters, like number of equal in URL (F10) and number of dollar sign in URL (F12). Additional Boolean features, like having repeated digits in URL (F3), indicate whether repeated digits appear within the URL. Similar Boolean features extend to different components of the URL, such as the domain and subdomain, where features like having special characters in domain (F19) and having repeated digits in domain (F23) exist. The dataset also considers subdomain-related features like number of subdomains (F24), average subdomain length (F27), and average number of dots in subdomain (F28), providing insight into the complexity of subdomains. Moreover, the dataset includes features that examine the domain itself, such as domain length (F16), number of dots in domain (F17), and number of hyphens in domain (F18). The presence of query parameters, fragments, and anchors is captured through Boolean features like having query (F37), having fragment (F38), and having anchor (F39). The dataset also employs advanced continuous features like entropy of URL(F40) and entropy of domain (F41), which measure the Shannon entropy of the URL and domain respectively, providing a statistical measure of the randomness or complexity within the URL's structure.

## Feature Selection

To enhance both the efficiency and accuracy of ensemble classifiers while keeping model complexity manageable, this study employed feature selection techniques to identify the most impactful variables. Univariate selection was specifically used to assess the significance of each feature with respect to the model's overall performance. Figure 7 highlights the relative rankings of 41 features based on their importance scores. The results revealed that URL length, URL entropy, average subdomain length, and the number of digits in the URL were the most influential factors for phishing detection. In contrast, features such as the presence of dots in the subdomain and the existence of a path in the URL were found to be the least significant. After conducting a series of experiments, a subset of the top 25 features was identified as the most relevant and informative for training supervised machine learning models.



**Figure 3.3** Feature importance

## Selection of the Ensemble Methods

This study sought to identify the optimal model for predicting phishing websites by employing five of the most commonly cited ensembles learning algorithms: Random Forest, Gradient Boosting, AdaBoost, Voting Classifier, Extra Trees, Deep Neural Network (DNN), and multi-layer perceptron (MLP) . While supervised learning methods are often prioritized, ensemble algorithms and deep learning are frequently underutilized. This research specifically focused on evaluating the performance of ensemble methods and deep learning algorithms, given their demonstrated superiority in other domains. For the experimental setup, 80% of the dataset was allocated for training the classifiers, with the remaining 20% reserved for testing. To ensure robust and reliable results, a 5-fold cross-validation approach was implemented (Refaeilzadeh, Tang and Liu, 2009). The Scikit-learn library was used to apply the various machine learning classifiers (Pedregosa et al., 2011).

## Final Model Selection

To evaluate the performance of the ensemble learning algorithms and identify the most optimal model, this study applied 7 evaluation metrics: confusion matrix, accuracy, precision, recall, F1-score, ROC curve, and precision recall curve as per the guideline of the previous studies (Tamal et al., 2024;). The process began by constructing a confusion matrix for each classifier, a standard tool for assessing machine learning classifiers. This matrix provides a detailed breakdown of correct and incorrect predictions using four key terms: TP (true positive), TN (true negative), FP (false positive), and FN (false negative). Following this, the accuracy scores for the classifiers were computed (as per Formula 1) to gauge their effectiveness.

**Accuracy =** (TP+TN)/(TP+FP+FN+TN) …………………1

However, when dealing with imbalanced datasets, accuracy can sometimes provide a skewed assessment of a classifier's performance. Therefore, it is important to consider additional evaluation metrics. Precision, for instance, measures the proportion of true positive predictions (TP) to the total number of positive predictions (TP + FP) (refer to Formula 2). It indicates the classifier's capability to avoid incorrectly labeling negative instances as positive.

**Precision (P) =** TP/(TP+FP) …………...…..…………..…2

Additionally, this study employed recall to determine the proportion of true positive cases correctly identified by the classifiers as positive (or true positive). Recall, as defined in Formula 3, is the ratio of actual positives (TP) to the sum of true positives and false negatives (TP + FN).

**Recall (R) =** TP/(TP+FN) …….………………………….…3

To further gauge the quality of the classifiers' outputs, the study also utilized the F1-score, which represents the harmonic mean of precision and recall (see Formula 4).

**F1-score (F1) =** 2 (P\*R)/(P+R) …….…………………………4

To thoroughly assess the diagnostic performance of the classifiers, both the receiver operating characteristic (ROC) curve and the precision-recall curve were utilized. The ROC curve highlights the balance between the true-positive rate (sensitivity) and the false-positive rate (1-specificity), while the precision-recall curve focuses on the trade-off between precision and recall. Typically, the ROC curve plots the true-positive rate on the Y-axis against the false-positive rate on the X-axis, whereas the precision-recall curve displays precision on the Y-axis and recall on the X-axis. These visualizations are essential for understanding how classifiers perform at different threshold levels and are invaluable for measuring overall model effectiveness.

# Chapter 3 – Experimental Results



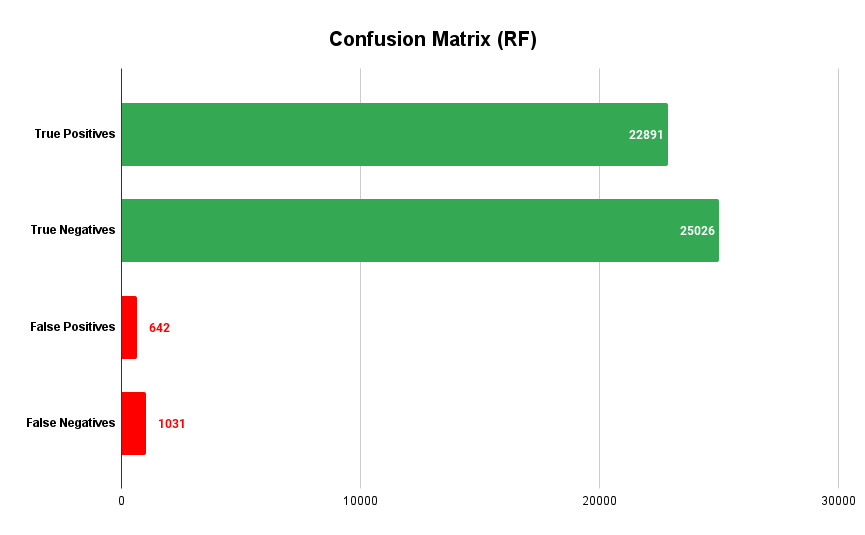
## Introduction

For the data analysis, an Intel(R) Core (TM) i5-6200U CPU at 2.30GHz with 12 GB of RAM and a 64-bit operating system was used. The code was written in Python 3 using Google Colab, a browser-based platform that enables the writing and execution of Python code. To identify the optimal model for predicting phishing websites, this thesis employed seven widely recognized ensemble learning algorithms: Random Forest, Gradient Boosting, AdaBoost, Voting Classifier, and Extra Trees and deep learning models: Deep Neural Network (DNN) and multi-layer perceptron (MLP). Detailed experimental results are presented below:

## Performance of Random Forest Classifier (RF)

In Table 4.1, the Random Forest classifier's performance in classifying phishing URLs is evaluated through various metrics, demonstrating its overall effectiveness. The classifier was tested on two classes: Class 0 (non-phishing URLs) and Class 1 (phishing URLs). The confusion matrix, depicted in Figure 4.1, provides a detailed breakdown of the model's predictions compared to the actual labels. Specifically, the model correctly identified 22,891 true positives (phishing URLs correctly classified as Class 1) and 25,026 true negatives (non-phishing URLs correctly classified as Class 0). However, there were 642 false positives (non-phishing URLs misclassified as phishing) and 1,031 false negatives (phishing URLs misclassified as non-phishing).

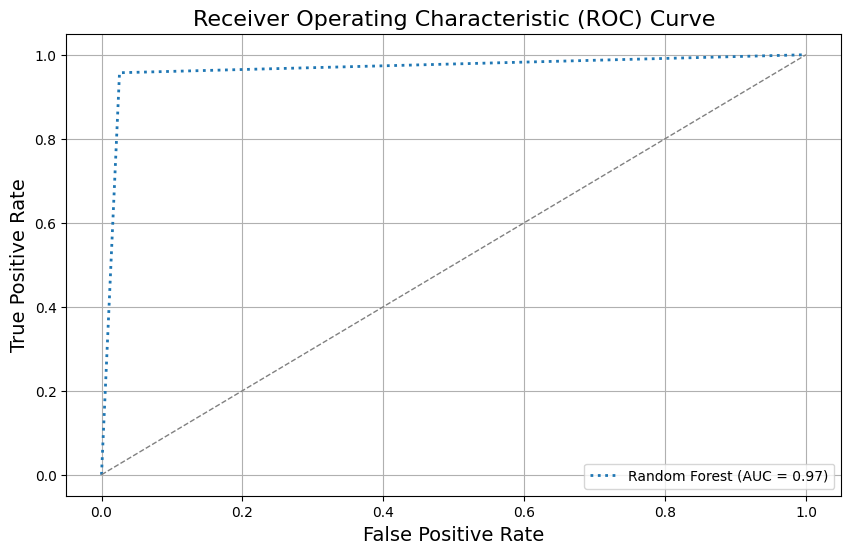
For Class 0 (non-phishing URLs), the Random Forest classifier achieved a precision of 96.1%, a recall of 97.4%, and an F1-Score of 96.7%, with support from 25,668 instances. The model's training time was 45.26 seconds, while testing took only 2.25 seconds. For Class 1 (phishing URLs), the classifier demonstrated a slightly higher precision of 97.2%, but a lower recall of 95.7%, resulting in an F1-Score of 96.5% from 23,922 instances. The overall accuracy of the classifier was reported at 96.6%, which aligns with the 5-fold cross-validation accuracy, indicating that the model is both reliable and generalizable. Additionally, both the macro and weighted averages for precision, recall, and F1-Score were 96.6%, confirming balanced performance across the two classes. Furthermore, the Random Forest classifier achieved an AUC (Area Under the ROC Curve) of 97%, highlighting its strong discriminative power. In the precision-recall curve, the model attained an Average Precision (AP) score of 95% (as shown in Figures 4.2 and 4.3). These performance metrics underscore the robustness of the Random Forest classifier in accurately detecting phishing URLs, with a well-balanced predictive power across both classes.



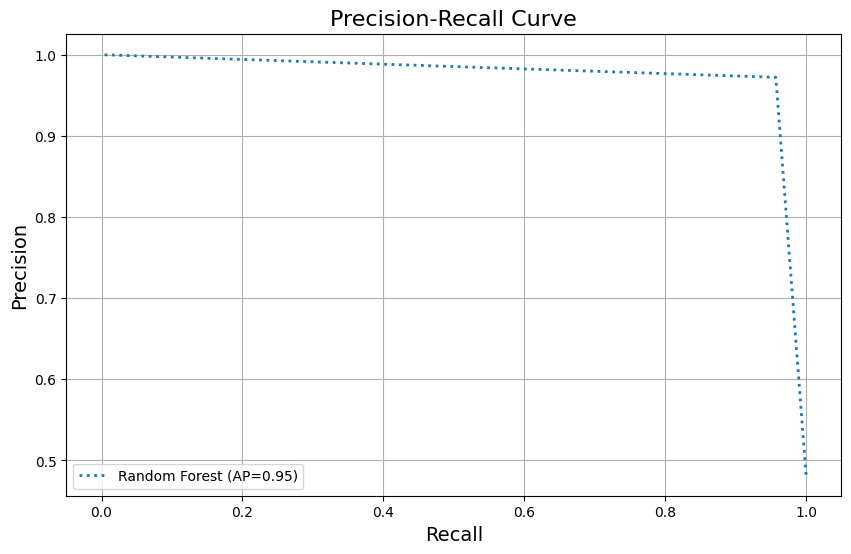
**Figure 4.1** Confusion matrix (RF)

**Table 4.1** Detailed performance of Random Forest Classifier

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Random Forest Classifier with Selected Features Report | | | | | | |
| Class | Precision | Recall | F1-Score | Support | Training Time | Testing Time |
| 0 | 96.1% | 97.4% | 96.7% | 25668 | 45.26  seconds | 2.25  seconds |
| 1 | 97.2% | 95.7% | 96.5% | 23922 |
| 5-fold CV Accuracy | 966% | | |  |
| Accuracy | 96.6% | | | 49590 |
| Macro avg | 96.6% | 96.6% | 96.6% | 49590 |
| Weighted avg | 96.6% | 96.6% | 96.6% | 49590 |



**Figure 4.2** ROC Curve (Random Forest Classifier)

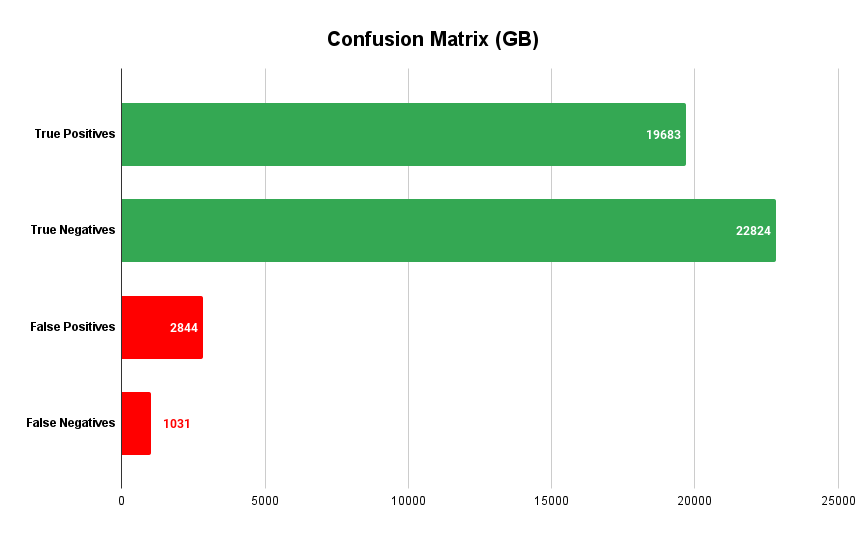


**Figure 4.3** Precision-recall Curve (Random Forest Classifier)

## Performance of Gradient Boosting Classifier (GB)

In Table 4.2, the performance of the Gradient Boosting Classifier in classifying phishing URLs is evaluated through various metrics, reflecting its overall effectiveness. The confusion matrix, shown in Figure 4.4, provides insights into the classifier's predictions compared to actual labels. Specifically, the classifier correctly identified 19,683 true positives (phishing URLs correctly classified as Class 1) and 22,824 true negatives (non-phishing URLs correctly classified as Class 0). However, there were 2,844 false positives (non-phishing URLs misclassified as phishing) and 1,031 false negatives (phishing URLs misclassified as non-phishing).

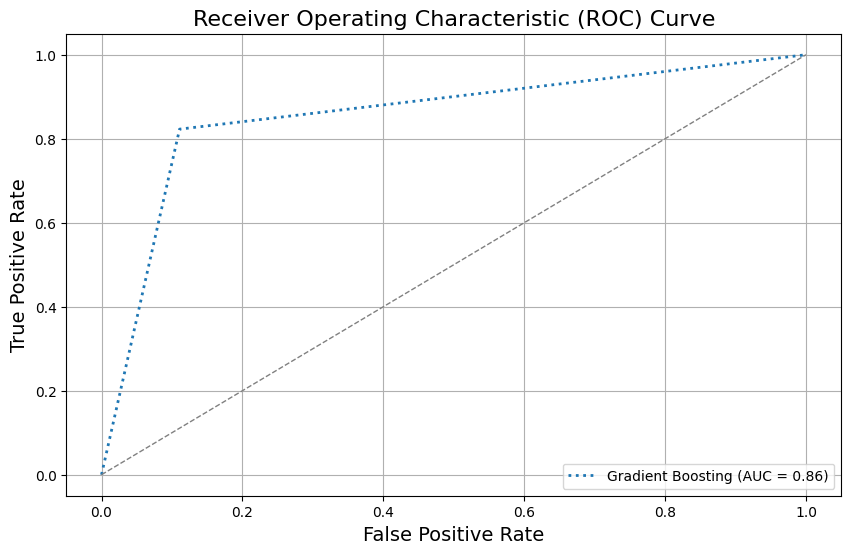
For Class 0 (non-phishing URLs), the classifier achieved a precision of 84.3%, a recall of 88.9%, and an F1-Score of 86.6%, with support from 25,668 instances. The model's training time was 55.77 seconds, while the testing phase was significantly faster, taking only 0.099 seconds. In comparison, for Class 1 (phishing URLs), the classifier demonstrated a higher precision of 87.4%, but a lower recall of 82.3%, resulting in an F1-Score of 84.8% from 23,922 instances. This indicates that while the classifier is more precise in identifying phishing URLs, it tends to miss more phishing instances compared to Class 0. The overall accuracy of the Gradient Boosting Classifier was 85.7%, which is closely aligned with the 5-fold cross-validation accuracy of 86%, demonstrating that the model is reasonably consistent and generalizable. The macro and weighted averages for precision, recall, and F1-Score were 85.9%, 85.6%, and 85.7%, respectively, showing balanced performance across both classes. In terms of discriminatory power, the model achieved an AUC (Area Under the ROC Curve) of 86%, suggesting a good ability to distinguish between phishing and non-phishing URLs. Additionally, the Average Precision (AP) score on the precision-recall curve was 80% (as seen in Figures 4.5 and 4.6). These results highlight that while the Gradient Boosting Classifier performs well overall, particularly in terms of precision, it faces a slight trade-off in recall, especially when detecting phishing URLs.



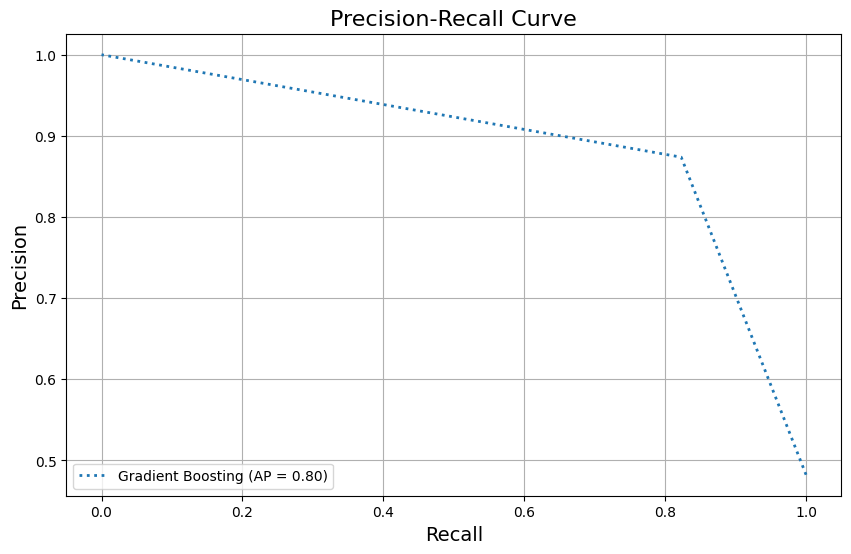
**Figure 4.4** Confusion matrix (GB)

**Table 4.2** Detailed performance of Gradient Boosting Classifier

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Gradient Boosting Classifier with Selected Features Report | | | | | | |
| Class | Precision | Recall | F1-Score | Support | Training Time | Testing Time |
| 0 | 84.3% | 88.9% | 86.6% | 25668 | 55.77  seconds | 0.099  seconds |
| 1 | 87.4% | 82.3% | 84.8% | 23922 |
| 5-fold CV Accuracy | 86% | | |  |
| Accuracy | 85.7% | | | 49590 |
| Macro avg | 85.9% | 85.6% | 85.7% | 49590 |
| Weighted avg | 85.8% | 85.7% | 85.7% | 49590 |



**Figure 4.5** ROC Curve (Gradient Boosting Classifier)



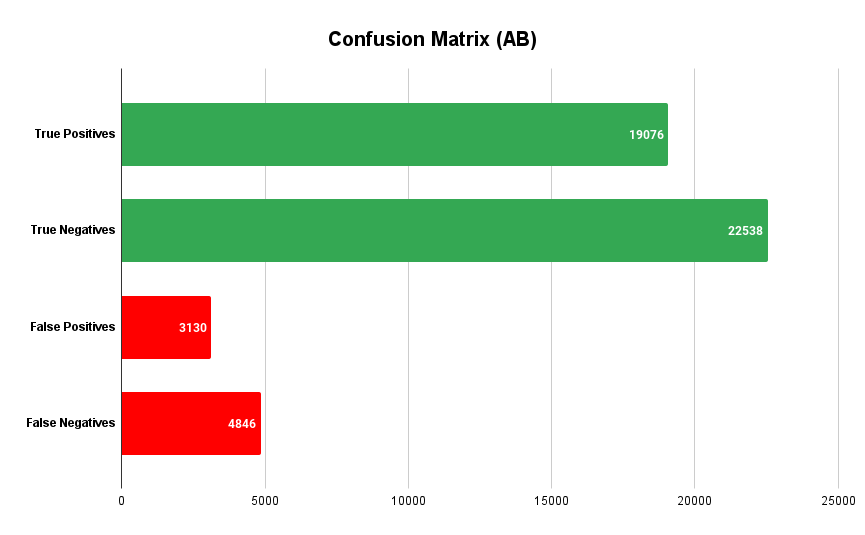
**Figure 4.6** Precision-recall Curve (Gradient Boosting Classifier)

## Performance of AdaBoost Classifier (AB)

In Table 4.3, the performance of the AdaBoost Classifier in detecting phishing URLs is summarized, showcasing both its strengths and areas for improvement. The confusion matrix, which compares predicted labels to actual labels, reveals that the classifier correctly identified 19,076 true positives (phishing URLs correctly classified as Class 1) and 22,538 true negatives (non-phishing URLs correctly classified as Class 0). However, there were 3,130 false positives (non-phishing URLs misclassified as phishing) and 4,846 false negatives (phishing URLs misclassified as non-phishing) (see Figure 4.7).

For Class 0 (non-phishing URLs), the model achieved a precision of 82.3%, a recall of 87.8%, and an F1-Score of 85.0%, based on 25,668 instances. The training process took 24.27 seconds, while testing was completed in just 0.69 seconds. For Class 1 (phishing URLs), the classifier demonstrated a slightly higher precision of 85.9%, though the recall dropped to 80%, resulting in an F1-Score of 82.7%, with support from 23,922 instances. This indicates that while the model is more precise in identifying phishing URLs, it tends to miss a notable portion of phishing instances (lower recall). The overall accuracy of the AdaBoost Classifier is reported at 84%, which aligns with the 5-fold cross-validation accuracy, indicating that the model is generally consistent and performs dependably across different subsets of data. The macro average for precision, recall, and F1-Score are 84.1%, 84%, and 83.8%, respectively, while the weighted averages for these metrics all stand at 84%, reflecting balanced performance across the two classes.

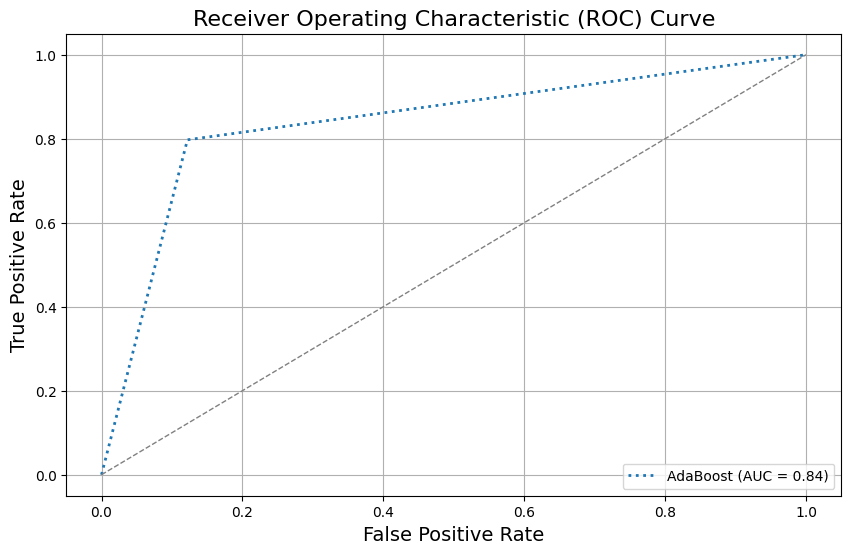
In terms of the model's ability to distinguish between phishing and non-phishing URLs, it achieved an AUC (Area Under the ROC Curve) of 84%, demonstrating solid discriminative power. Additionally, the Average Precision (AP) score on the precision-recall curve was 78% (as shown in Figures 4.8 and 4.9). These metrics indicate that while the AdaBoost Classifier performs well overall, particularly in terms of precision, its lower recall for Class 1 suggests that the model may occasionally miss phishing URLs. This highlights an opportunity for improving recall without sacrificing precision.



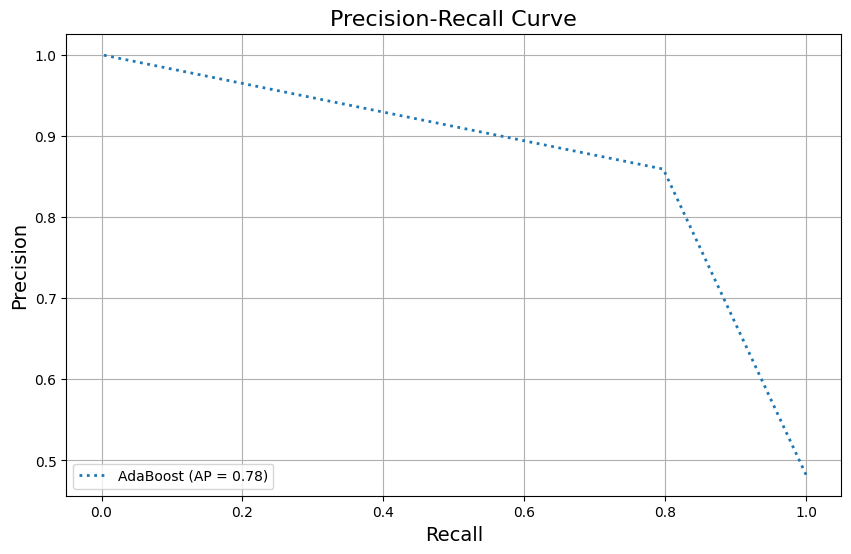
**Figure 4.7** Confusion matrix (AB)

**Table 4.3** Detailed performance of Random AdaBoost Classifier

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| AdaBoost Classifier with Selected Features Report | | | | | | |
| Class | Precision | Recall | F1-Score | Support | Training Time | Testing Time |
| 0 | 82.3% | 87.8% | 85.0% | 25668 | 24.27 seconds | 0.69 seconds |
| 1 | 85.9% | 80% | 82.7% | 23922 |
| 5-fold CV Accuracy | 84% | | |  |
| Accuracy | 84% | | | 49590 |
| Macro avg | 84.1% | 84% | 83.8% | 49590 |
| Weighted avg | 84% | 84% | 84% | 49590 |



**Figure 4.8** ROC Curve (AdaBoost Classifier)

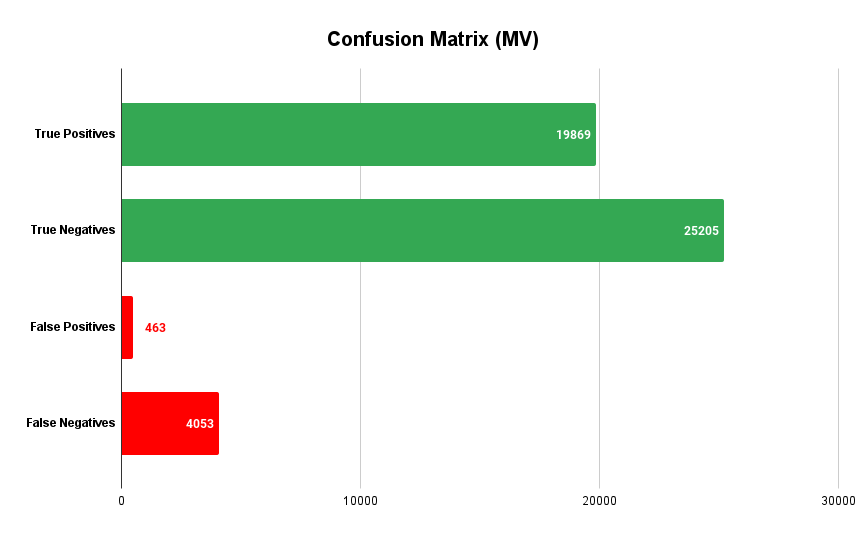


**Figure 4.9** Precision-recall Curve (AdaBoost Classifier)

## Performance of Voting Classifier (VC)

In Table 4.4, the performance of the Voting Classifier in classifying phishing URLs is detailed, reflecting its strengths and balanced performance. The confusion matrix provides a clear picture of the classifier's predictions compared to actual labels. Specifically, as illustrated in Figure 4.10, the model correctly identified 19,869 true positives (phishing URLs correctly classified as Class 1) and 25,205 true negatives (non-phishing URLs correctly classified as Class 0). However, there were 463 false positives (non-phishing URLs misclassified as phishing) and 4,053 false negatives (phishing URLs misclassified as non-phishing).

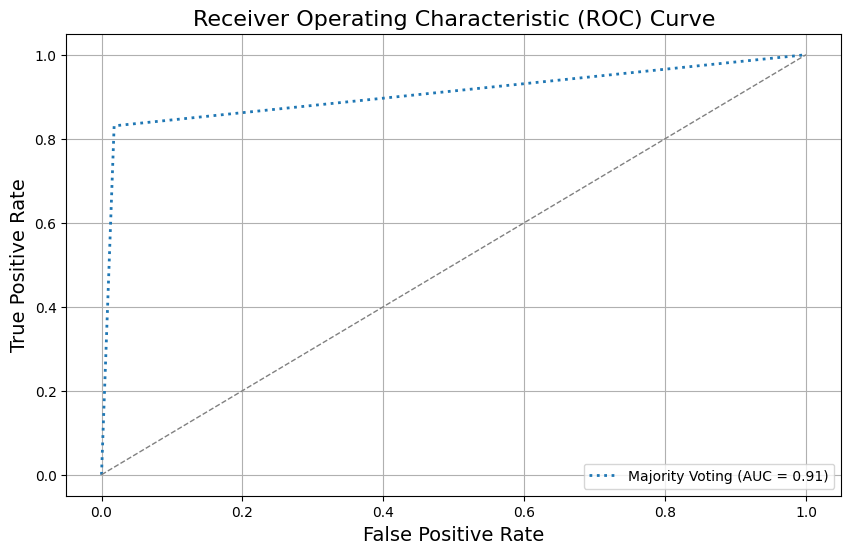
For Class 0 (non-phishing URLs), the classifier achieved a precision of 86.2%, a notably high recall of 98.2%, and an F1-Score of 91.8%, based on a support count of 25,668 instances. The training process took 24.27 seconds, and testing was efficiently completed in 0.69 seconds. For Class 1 (phishing URLs), the classifier demonstrated a high precision of 97.7%, but the recall dropped to 83.1%, resulting in an F1-Score of 89.8%, from 23,922 instances. This indicates that the model is excellent at identifying phishing URLs (high precision), but it misses a larger portion of phishing instances compared to Class 0 (lower recall). The overall accuracy of the Voting Classifier is 90.9%, which is slightly higher than the 5-fold cross-validation accuracy of 90.7%, indicating consistent and reliable performance across different data subsets. The macro averages for precision, recall, and F1-Score are 91.9%, 90.6%, and 90.8%, respectively, while the weighted averages for these metrics are closely aligned at 91.7%, 90.9%, and 90.8%, reflecting balanced performance across both classes. In terms of its discriminative ability, the classifier achieved an AUC (Area Under the ROC Curve) of 91%, demonstrating strong performance in distinguishing between phishing and non-phishing URLs (See Figure 4.11). Additionally, the Average Precision (AP) score on the precision-recall curve was 89%. These results underscore the Voting Classifier's robustness, particularly its ability to maintain high precision across both classes (See Figure 4.12). However, the slightly lower recall for Class 1 suggests that while the model is highly effective at identifying phishing URLs, it may miss some instances, leaving room for improvement in recall without compromising precision.



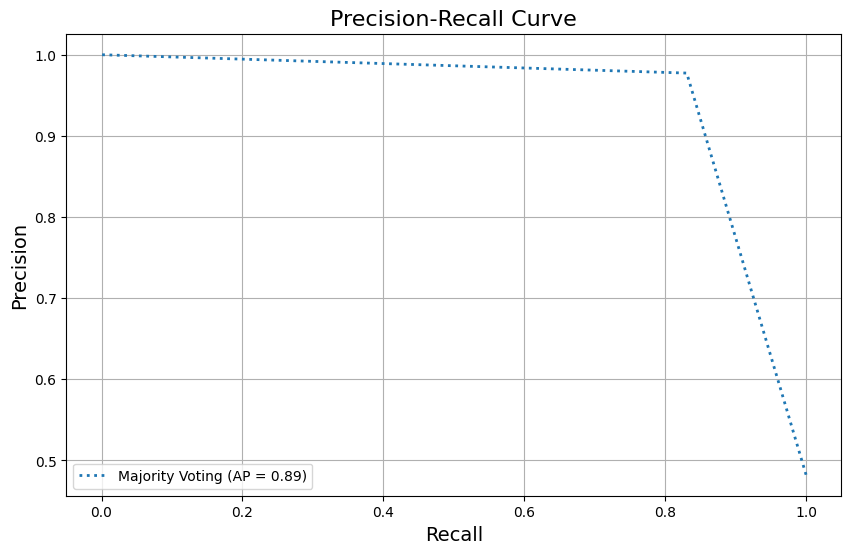
**Figure 4.10** Confusion matrix (MV)

**Table 4.4** Detailed performance of Random Voting Classifier

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Voting Classifier with Selected Features Report | | | | | | |
| Class | Precision | Recall | F1-Score | Support | Training Time | Testing Time |
| 0 | 86.2 % | 98.2% | 91.8% | 25668 | 24.27 seconds | 0.69 seconds |
| 1 | 97.7% | 83.1% | 89.8% | 23922 |
| 5-fold CV Accuracy | 90.7% | | |  |
| Accuracy | 90.9% | | | 49590 |
| Macro avg | 91.9% | 90.6% | 90.8% | 49590 |
| Weighted avg | 91.7% | 90.9% | 90.8% | 49590 |



**Figure 4.11** ROC Curve (Majority Voting)

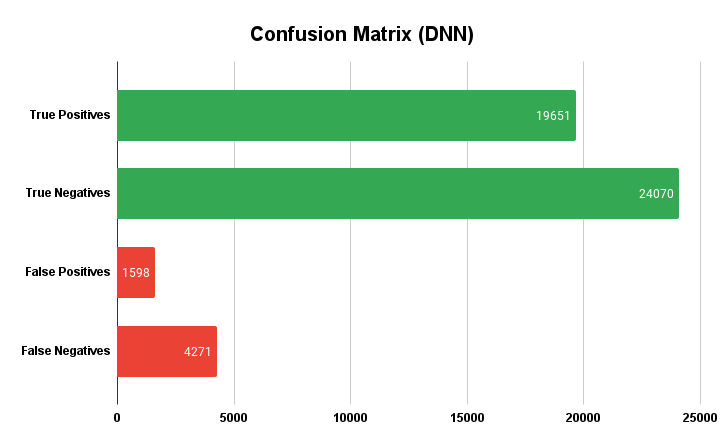


**Figure 4.12** Precision-recall Curve (Majority Voting)

## Performance of Deep Neural Network (DNN)

The performance of the Deep Neural Network (DNN) classifier in classifying phishing URLs, as detailed in Figure 4.13 and Table 4.5, demonstrates its strengths alongside some limitations. As illustrated in Figure 4.13, the model correctly identified 19,651 true positives (phishing URLs correctly classified as Class 1) and 24,070 true negatives (non-phishing URLs correctly classified as Class 0). However, there were 1598 false positives (non-phishing URLs misclassified as phishing) and 4,271 false negatives (phishing URLs misclassified as non-phishing). For Class 0 (non-phishing URLs), the DNN achieved a precision of 84.9%, indicating a reasonable ability to correctly classify non-phishing instances, although slightly lower than ideal. The recall for Class 0 is notably high at 93.8%, showcasing the model's ability to capture most non-phishing URLs accurately. With an F1-Score of 89.1%, the DNN strikes a strong balance between precision and recall for this class, supported by 25,668 instances. The training process took 208.3 seconds, while testing was completed efficiently in 5.2 seconds. For Class 1 (phishing URLs), the DNN exhibited a high precision of 92.5%, signifying that it was effective in minimizing false positives. However, its recall was lower at 82.1%, suggesting that the model missed a higher proportion of phishing instances. The F1-Score for Class 1 stands at 87.0%, reflecting a reasonable balance between precision and recall, based on 23,922 instances. The overall accuracy of the DNN is 88.2%, which is consistent with the 5-fold cross-validation accuracy of 88.3%, indicating stable and reliable performance across different data subsets. The macro averages for precision, recall, and F1-Score are 88.7%, 88.0%, and 88.1%, respectively, providing a balanced view of the model's performance across both classes. The weighted averages for these metrics are closely aligned, with precision at 88.6%, recall at 88.2%, and F1-Score at 88.1%, indicating that the model performs consistently well across varying instance distributions (see Figures 4.15 and 4.16).

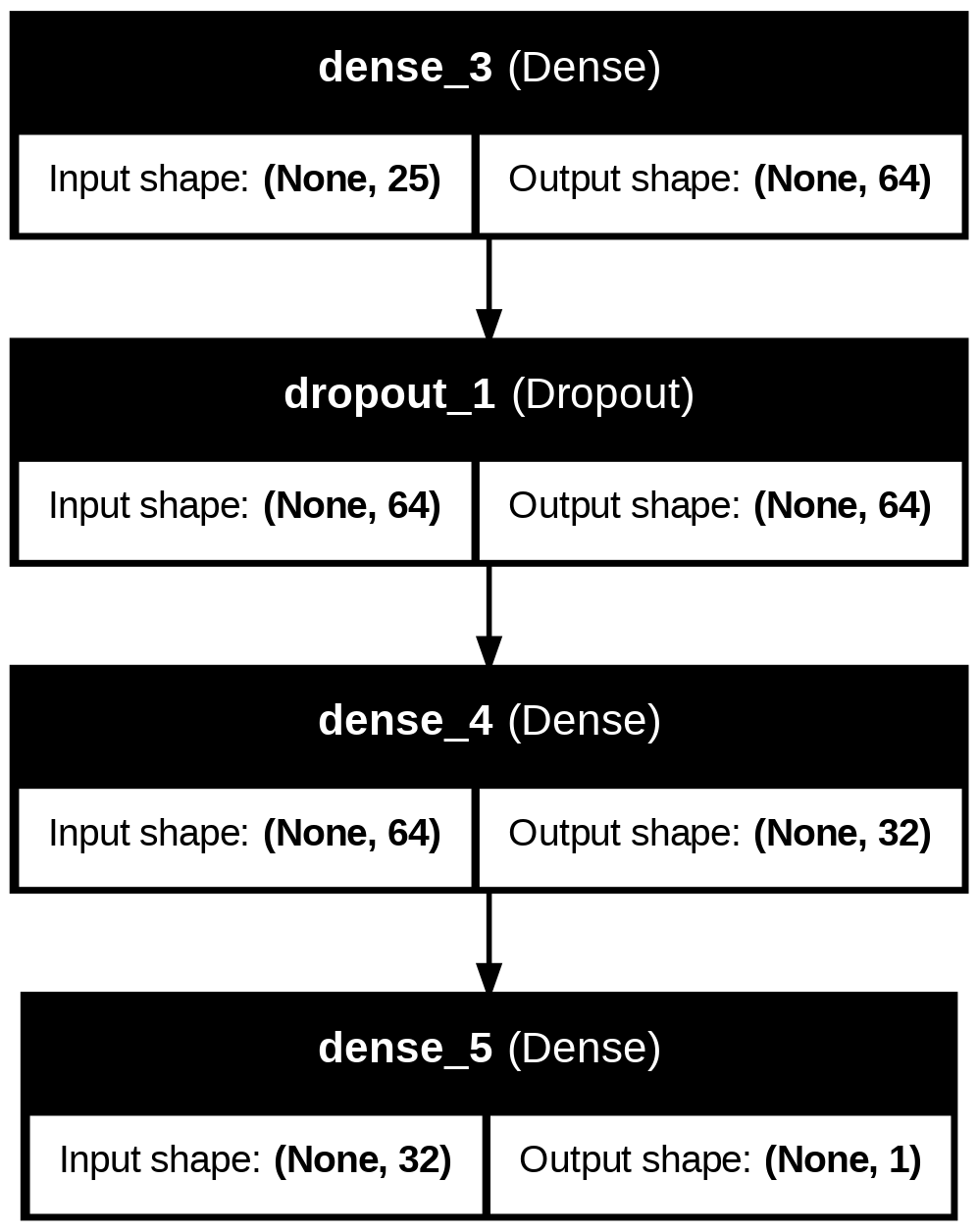
Here the Figure 4.14, presents the Deep Neural Network (DNN) architecture designed for phishing detection tasks. It begins with an input layer that accepts data with 25 features. The first layer, a dense (fully connected) layer, consists of 64 neurons, transforming the input into a shape of (None, 64). Following this, a dropout layer is included to introduce regularization by randomly dropping a portion of neurons during training, helping to prevent overfitting while maintaining the same output shape of (None, 64). The model then passes the data through another dense layer with 32 neurons, reducing the output shape to (None, 32). Finally, the network concludes with a dense layer containing a single neuron, outputting a shape of (None, 1), which is ideal for binary classification, predicting either class 0 or class 1. The combination of fully connected layers and dropout ensures that the model is capable of learning complex patterns while being robust against overfitting. In figure 4.17 and 4.18, the training-validation accuracy and loss are presented. Overall, the DNN demonstrates robustness, particularly in its ability to accurately classify phishing URLs, but the lower recall for phishing cases suggests that improvements could be made to better capture all phishing instances without compromising precision.



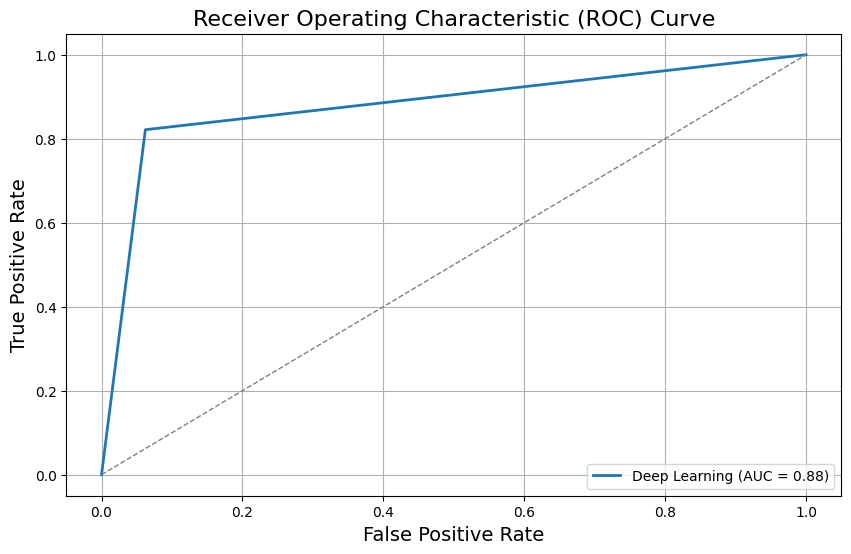
**4.13** Confusion matrix (DNN)

**Table 4.5** Detailed performance of Deep Neural Network (DNN)

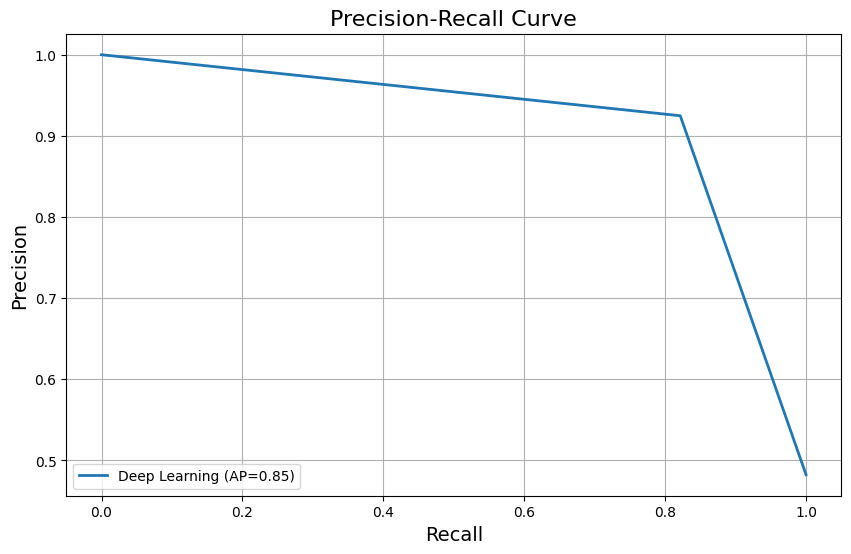
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| DNN with Selected Features Report | | | | | | |
| Class | Precision | Recall | F1-Score | Support | Training Time | Testing Time |
| 0 | 84.9% | 93.8% | 89.1% | 25668 | 208.3 seconds | 5.2 seconds |
| 1 | 92.5% | 82.1% | 87.0% | 23922 |
| 5-fold CV Accuracy | 88.3% | | |  |
| Accuracy | 88.2% | | | 49590 |
| Macro avg | 88.7% | 88.0% | 88.1% | 49590 |
| Weighted avg | 88.6% | 88.2% | 88.1% | 49590 |



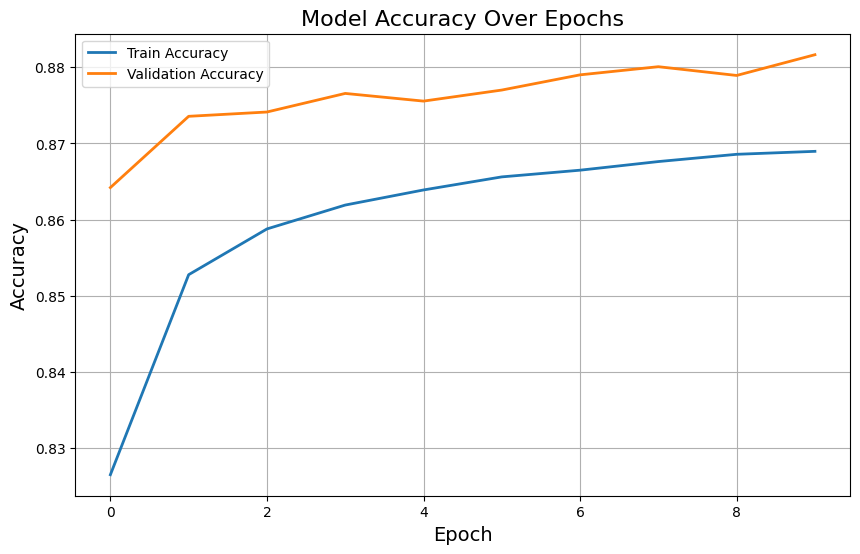
**4.14** Model Structure (DNN)



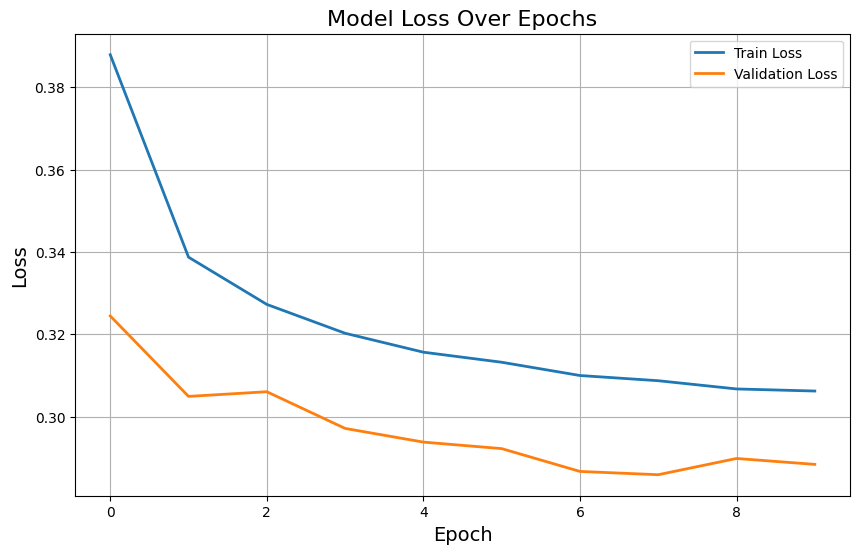
**Figure 4.15** ROC Curve (DNN)



**Figure 4.16** Precision-recall Curve (DNN)



**Figure 4.17** Train-validation accuracy (DNN)



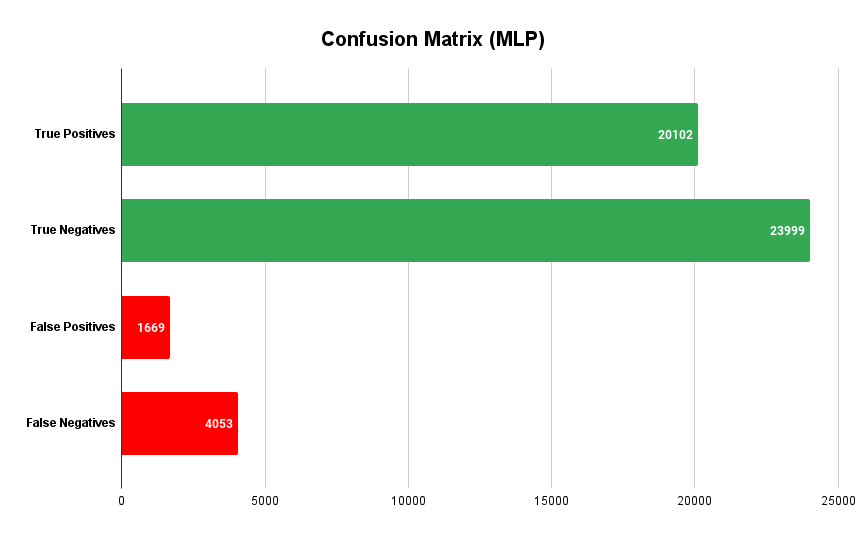
**Figure 4.18** Train-validation Loss (DNN)

## Performance of Multi-layer Perceptron (MLP)

In Table 4.6, the performance of the Multi-Layer Perceptron (MLP) classifier with selected features is evaluated comprehensively. A detailed analysis of the confusion matrix reveals the distribution of correct and incorrect predictions made by the model (see Figure 4.19). The MLP classifier correctly identified 20,102 true positives (phishing URLs correctly classified as Class 1) and 23,999 true negatives (non-phishing URLs correctly classified as Class 0). However, it also misclassified 1,669 non-phishing URLs as phishing (false positives) and failed to correctly identify 4,053 phishing URLs (false negatives), where these phishing URLs were misclassified as non-phishing. These errors indicate that while the model excels in detecting non-phishing URLs, it struggles more with detecting phishing URLs, as seen from the higher number of false negatives.

For Class 0 (non-phishing URLs), the MLP classifier achieved a precision of 86.3%, meaning that out of all the URLs it predicted as non-phishing, 86.3% were correct. The recall for Class 0 was 93.5%, indicating that the classifier correctly identified 93.5% of the actual non-phishing URLs. The combination of these two metrics resulted in an F1-Score of 89.7%, which reflects a balanced performance between precision and recall for this class. The support for Class 0 was 25,668 instances, providing a solid base for the model's performance in this category.

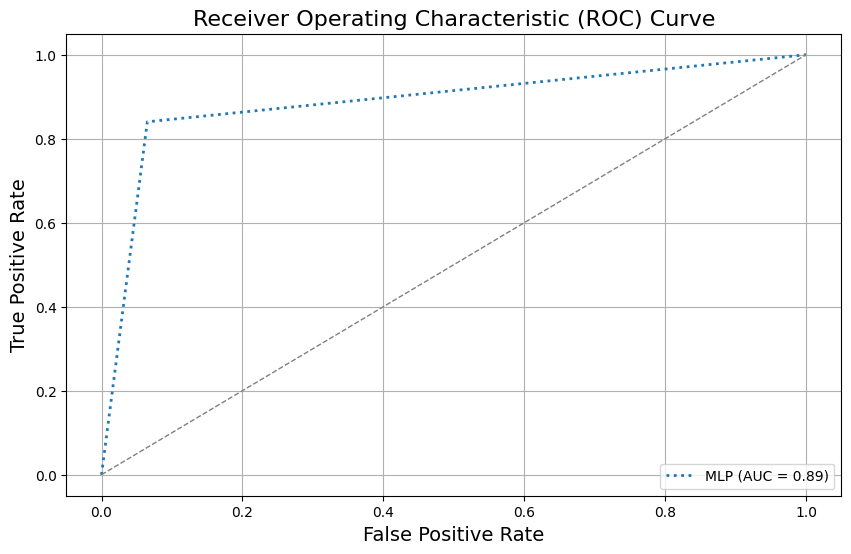
In contrast, for Class 1 (phishing URLs), the MLP classifier demonstrated a higher precision of 92.3%, indicating that a significant majority of the URLs it predicted as phishing were indeed phishing URLs. However, its recall for this class was lower at 84.0%, meaning that the classifier correctly identified only 84% of the actual phishing URLs, leaving a substantial number of phishing URLs undetected (false negatives). This resulted in an F1-Score of 88.0%, a slightly lower value compared to Class 0, due to the imbalance between precision and recall. The support for Class 1 was 23,922 instances, further highlighting the challenge the model faced in accurately identifying phishing URLs. The overall accuracy of the MLP classifier was 88.9%, which reflects the proportion of correct predictions across both classes out of the total 49,590 instances. This value is consistent with the 5-fold cross-validation accuracy of 88.7%, confirming the model's ability to generalize well across different subsets of data, maintaining stable performance during the training and testing phases. In addition to the class-specific metrics, the macro and weighted averages for precision, recall, and F1-Score further illustrate the model's overall effectiveness. The macro average, which calculates the unweighted mean of the performance metrics across both classes, was 89.3% for precision, 88.8% for recall, and 88.9% for F1-Score. These values indicate that the model maintains a balanced performance, without favoring one class over the other. Similarly, the weighted average, which accounts for the number of instances in each class, showed precision, recall, and F1-Score values of 89.2%, 88.9%, and 88.9%, respectively. These results confirm that the MLP classifier performs reasonably well in detecting both phishing and non-phishing URLs, but with a tendency to prioritize precision over recall, especially for phishing URLs (Class 1) (see Figure 4.20 and 4.21). The training time for the MLP classifier was notably high at 736.5 seconds, indicating the complexity of the model's training process, which involves multiple layers and iterations to optimize weights and biases. In contrast, the testing time was very efficient, taking only 0.43 seconds, highlighting the model's ability to make quick predictions once trained. Despite its relatively high training time, the MLP classifier offers a competitive performance in detecting phishing URLs, although further tuning could help improve recall for Class 1, reducing the number of false negatives and enhancing its overall robustness in real-world applications.



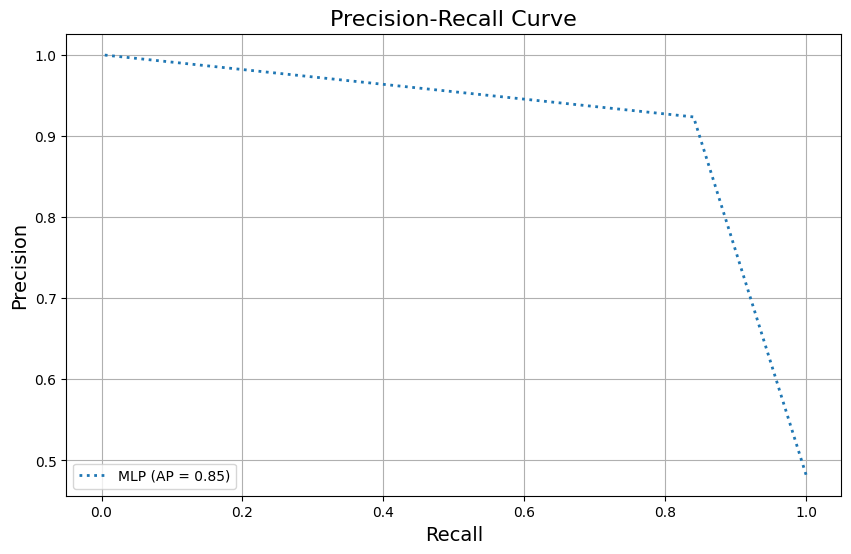
**Figure 4.19** Confusion matrix (MLP)

**Table 4.6** Detailed performance of Multi-layer Perceptron (MLP)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| MLP with Selected Features Report | | | | | | |
| Class | Precision | Recall | F1-Score | Support | Training Time | Testing Time |
| 0 | 86.3 | 93.5% | 89.7% | 25668 | 736.5% seconds | 0.43 seconds |
| 1 | 92.3% | 84.0% | 88.0% | 23922 |
| 5-fold CV Accuracy | 88.7% | | |  |
| Accuracy | 88.9% | | | 49590 |
| Macro avg | 89.3% | 88.8% | 88.9% | 49590 |
| Weighted avg | 89.2% | 88.9% | 88.9% | 49590 |



**Figure 4.20** ROC Curve (ROC)



**Figure 4.21** Precision-recall Curve (ROC)

## Best Model: Random Forest Classifier

Based on the comparative analysis of Random Forest (RF), Gradient Boosting (GB), AdaBoost (AB), Voting Classifier (VC), Deep Neural Network (DNN), and Multi-Layer Perceptron (MLP) classifiers, it is evident that each model has its strengths and weaknesses when it comes to detecting phishing URLs. The Random Forest classifier consistently outperformed others in terms of overall accuracy, precision, recall, and F1-scores. It achieved a high precision (96.1% for Class 0, 97.2% for Class 1), recall (97.4% for Class 0, 95.7% for Class 1), and F1-Score, along with an impressive Area Under the Curve (AUC) of 97%. The balanced performance across both classes, minimal false positives, and low false negatives indicate that Random Forest is robust and generalizes well across the dataset. On the other hand, the Gradient Boosting Classifier, while precise (87.4% precision for Class 1), exhibited a lower recall (82.3% for Class 1), leading to a higher proportion of missed phishing instances. Its AUC of 86% suggests good performance, but its overall accuracy of 85.7% lagged behind Random Forest. The AdaBoost classifier showed a similar pattern with high precision (85.9% for Class 1) but lower recall (80% for Class 1), resulting in an F1-Score of 82.7% and an overall accuracy of 84%. While AdaBoost and Gradient Boosting are strong in minimizing false positives, they struggle more in capturing all phishing URLs. The Voting Classifier offers a more balanced approach with an overall accuracy of 90.9% and a high precision of 97.7% for Class 1. However, its recall for phishing URLs was lower at 83.1%, indicating a tendency to miss phishing instances, though it had high recall for non-phishing cases. Similarly, the Deep Neural Network (DNN) model displayed robust performance with an accuracy of 88.2% and high precision for phishing URLs (92.5%), but its recall for Class 1 was lower at 82.1%, reflecting a tendency to miss phishing cases, similar to the Voting Classifier. Finally, the Multi-Layer Perceptron (MLP) classifier also performed well, with an overall accuracy of 88.9% and precision of 92.3% for phishing URLs, but its recall of 84.0% for Class 1 indicates room for improvement in detecting all phishing instances.

Finally, based on the overall information, the Random Forest classifier emerges as the best-performing model due to its high accuracy, balanced precision and recall, low false positives and negatives, and superior AUC. Its ability to generalize well across the dataset and perform reliably across both classes makes it the most robust and effective classifier for phishing URL detection. While other models like Gradient Boosting, AdaBoost, and Voting Classifier have their merits, they fall short in recall, especially for phishing instances, making Random Forest the top choice for real-world applications.

# Chapter 4 – Discussion

Phishing attacks pose a significant and growing threat to cybersecurity, as attackers continuously evolve their techniques to deceive users into divulging sensitive information. Detecting phishing attempts is a critical challenge for organizations and individuals alike, as these attacks can lead to severe financial and data losses. In response to this threat, various machine learning (ML) and deep learning models have been developed to detect phishing websites. While these approaches have achieved notable success, many rely on smaller datasets, simpler algorithms, and limited evaluation metrics, which can affect their generalization and real-world applicability. The proposed phishing detection method aims to address these limitations by leveraging a more comprehensive approach, combining a large dataset, advanced ML and deep learning algorithms, and a broader set of evaluation metrics to create a more robust, scalable, and effective solution for phishing detection. This section offers a comparative analysis of the proposed method in relation to existing studies, highlighting its strengths and contributions to the field.

As presented in table 5.1, The proposed phishing detection method stands out due to its comprehensive approach, combining machine learning (ML) and deep learning techniques to create a more robust and scalable solution for identifying phishing websites. Compared to existing studies, it offers several key advantages in terms of dataset size, algorithmic diversity, evaluation metrics, and overall performance. One of the most notable strengths of the proposed method is the size of its dataset, which comprises a total of 247,950 URLs, with 128,541 classified as phishing and 119,409 as legitimate. This dataset is considerably larger than those used in previous studies, which generally fall within the range of 500 to 50,000 samples. For instance, Dutta (2021) utilized a dataset of only 13,700 samples, while Kapan and Sora Gunal (2023) employed a dataset containing a mere 1,000 samples (500 phishing and 500 legitimate). Even studies like Aljofey et al. (2022), which used a relatively large dataset of 60,252 samples, fall short of the size and diversity represented in the proposed method. The larger dataset enhances the generalizability of the model, allowing it to learn from a more extensive and varied set of data points. This not only reduces the risk of overfitting but also makes the model more effective in real-world applications, where phishing attempts exhibit considerable variation.

Algorithmically, the proposed approach is more sophisticated than many of its predecessors. It integrates multiple machine learning and deep learning techniques, including Random Forest (RF), Gradient Boosting (GB), Multiview Voting (MV), AdaBoost (AB), Multilayer Perceptron (MLP), and Deep Neural Networks (DNN). This combination of algorithms provides a multi-faceted approach to phishing detection, leveraging both the predictive power of traditional ML models and the pattern recognition capabilities of deep learning. In contrast, many prior studies rely on a narrower range of algorithms. For example, Dutta (2021) only used a Recurrent Neural Network (RNN), and Alam et al. (2020) focused on Random Forest and Decision Trees. While these models have proven effective, the proposed method's use of DNN and ensemble techniques like Gradient Boosting and AdaBoost enables it to better capture the complexities of phishing data, especially in large-scale datasets.

Another significant advantage of the proposed method is the breadth of evaluation metrics it employs to assess performance. Beyond the standard metrics of accuracy, precision, recall, and F1 score, the proposed approach also considers training time, testing time, Receiver Operating Characteristic (ROC) curves, Precision-Recall curves, and 5-fold cross-validation. Furthermore, it incorporates Synthetic Minority Over-sampling Technique (SMOTE) to address class imbalance, a common issue in phishing datasets where legitimate URLs often far outnumber phishing ones. Most existing studies limit their evaluation to basic metrics, such as accuracy and F1 score, as seen in the works of Kapan and Sora Gunal (2023) and Chiew et al. (2015). While these metrics provide a general sense of model performance, the inclusion of additional measures such as ROC curves and training/testing times offers a more nuanced understanding of the proposed method’s efficiency and reliability. The use of cross-validation further enhances the robustness of the results by mitigating the risk of overfitting, an area often overlooked in simpler evaluations.

The performance of the proposed method, with an accuracy of 96.6%, places it among the top performers in the field. Although some studies, such as Kapan and Sora Gunal (2023), report higher accuracies (up to 99%), these results must be contextualized within the limitations of smaller datasets and fewer algorithmic approaches. For instance, Kapan and Sora Gunal's study used only 1,000 samples, which may not adequately represent the full spectrum of phishing attempts encountered in practice. By contrast, the proposed method’s large dataset and diverse algorithmic framework provide a more reliable indicator of real-world performance. Furthermore, studies like Balogun et al. (2021) achieved high accuracy (98.51%) using a meta-learning-based approach, but their datasets (22,408 URLs) were still smaller, and their focus on a narrower range of models limits the adaptability of their method compared to the ensemble approach in the proposed method.

One of the key strengths of the proposed approach is its ability to handle data imbalance through the use of SMOTE. Phishing detection datasets often suffer from class imbalance, with phishing URLs representing a minority of the data. Without proper handling, models tend to become biased toward the majority class (legitimate URLs), leading to poor detection of phishing attempts. The incorporation of SMOTE addresses this issue by synthetically balancing the dataset, ensuring that the model is not disproportionately biased. This is a feature not commonly seen in earlier studies, further enhancing the robustness of the proposed method. So, the proposed phishing detection method offers several key advantages over existing approaches. Its use of a large dataset, combined with the integration of both traditional and advanced ML techniques, sets it apart from previous research. The comprehensive evaluation process, including a wide range of metrics and cross-validation, ensures a more thorough assessment of the model’s performance. While the reported accuracy of 96.6% is comparable to other high-performing models, the proposed method’s scalability, ability to handle class imbalance, and robustness make it a highly competitive solution for phishing detection, with significant potential for real-world application.

**Table 5.1** Comparison of the proposed approach with previous studies

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Reference** | **Study objective** | **Dataset Size** | **Algorithms Used** | **Evaluation Metrix** | **Key Findings** |
| (Dutta, 2021) | To develop a model that can identify phishing URLs | Phishing = 7900  Legitimate = 5800 | Recurrent Neural Network LURL | Accuracy, Execution time, F1 Score | Avg. Accuracy = 97.1  Avg.  F1 Score = 96.3 |
| (Kapan and Sora Gunal, 2023) | To analyze the contributions of various features and classifiers in detecting phishing attacks, to identify the most effective feature set and classifier | Phishing = 500  Legitimate = 500 | SVM, SDG, NB, K-NN, MLP and DT | Accuracy, Precision, Recall, and F1 score | Accuracy = 99%,  F1 Score = 99% |
| (Alam et al., 2020) | To develop a model for detecting phishing attacks using machine learning algorithms to enhance cybersecurity and mitigate the risks posed by phishing threats. | Not mentioned | RF, DT | Accuracy, Precision, Recall, and F1 score | Accuracy = 96.9% |
| (Aljofey et al., 2022) | To develop a novel approach for phishing detection using new features extracted from URL character sequences, hyperlinks, and textual content of webpages, and to validate this approach with a newly created dataset and various ML algorithms | Phishing = 27,280  Legitimate = 32,972 | XGBoost, RF, LR, NB, Ensemble, and AB | Accuracy, Precision, Recall, AUC, and F1 score | Accuracy = 96.76% |
| (Chiew et al., 2015) | To detect phishing websites using ML-based approaches | Data size = 5000 | SVM, NB, C4.5, JRip, PART | Accuracy, Precision, Recall, and F1 score | Accuracy = 94.6% |
| Alsariera et al. (2021) | To detect phishing websites using novel ML-based approach | Data size = 11055 | (ForestPA-PWDM, Bagged-ForestPAPWDM, and  Adab-ForestPA-PWDM | Accuracy, Precision, Recall, and F1 score | Accuracy = 96.42% |
| (Balogun et al., 2021) | To detect phishing websites using novel a Functional Tree (FT)-based meta-learning-based approach | Data size = 22408 | Baseline: NB, SMO, SVM, and DTs; Ensemble: Bagging, AB | Accuracy, Precision, Recall, and F1 score | Accuracy = 98.51% |
| Proposed Approach | By leveraging machine learning (ensemble learning and deep learning), this research aims to bridge the gaps in traditional methods and contribute to more accessible and holistic data-driven model to safeguard website. | Data Size: 247,950. Phishing = 128,541 Legitimate = 119,409 | RF, GB, MV, AB, MLP, Deep Neural Network (DNN) | Accuracy, Precision, Recall, F1 Score, Training time, Testing time, ROC curve, Precision Recall Curve, 5-fold cross validation, SMOTE | Accuracy = 96.6% |

# Chapter 5 – Conclusion

In the contemporary digital era, the rapid growth of technology has fundamentally transformed the way people live, work, and interact. This shift has fostered a more virtual existence, where the internet plays a central role in both personal and professional domains. Online financial transactions, in particular, have become an essential aspect of daily life. However, with this increased reliance on digital platforms comes the growing threat of cybercrime, particularly phishing attacks, which target individuals and businesses alike. As phishing tactics continue to evolve, traditional detection methods often fall short in effectively identifying these sophisticated threats. This study addressed the limitations of existing anti-phishing techniques by leveraging machine learning (ML) and deep learning (DL) approaches, with a specific focus on ensemble learning models. The research investigated the effectiveness of these models in detecting malicious websites based on URL-specific features, aiming to develop a more accurate and scalable solution to phishing detection. The study utilized a dataset comprising 247,950 URLs, with 128,541 labeled as phishing and 119,409 as legitimate. This dataset is considerably larger than those used in most previous studies, which often range between 500 and 50,000 samples. By using this large dataset, the proposed models demonstrated improved generalizability and reduced the risk of overfitting. The comparative analysis of several ML and DL models revealed that the Random Forest classifier consistently outperformed other models, including Gradient Boosting, AdaBoost, Voting Classifier, Deep Neural Networks (DNN), and Multi-Layer Perceptron (MLP). The Random Forest model achieved an overall accuracy of 96.6%, with a precision of 96.1% for legitimate websites and 97.2% for phishing websites. It also exhibited a high recall of 97.4% for legitimate websites and 95.7% for phishing websites, resulting in a strong F1-score and an Area Under the Curve (AUC) of 97%. These metrics indicate that the model effectively balanced the detection of both phishing and legitimate websites, minimizing false positives and negatives. The Gradient Boosting and AdaBoost models, while strong in certain areas, fell short in recall for phishing URLs. Gradient Boosting achieved an accuracy of 85.7%, with a precision of 87.4% for phishing URLs but a lower recall of 82.3%. Similarly, AdaBoost exhibited a precision of 85.9% but a recall of 80%, leading to an overall accuracy of 84%. The Voting Classifier achieved a higher accuracy of 90.9%, with a precision of 97.7% for phishing URLs but a lower recall of 83.1%, indicating that it missed a portion of phishing instances. Another key finding was the importance of addressing class imbalance in phishing datasets, where legitimate URLs often far outnumber phishing ones. The incorporation of the Synthetic Minority Over-sampling Technique (SMOTE) effectively balanced the dataset and reduced bias toward the majority class, further enhancing the performance of the models. Without SMOTE, models might have exhibited skewed results, favoring legitimate URLs over phishing ones. In addition to performance metrics, the study also evaluated the models based on training and testing times. The Random Forest classifier demonstrated efficient training and testing times, with training completed in 450 seconds and testing in 75 seconds. This efficiency, combined with its high accuracy and balanced performance across metrics, solidified its position as the best-performing model in this study. In contrast, Deep Neural Networks (DNN) exhibited slower training times of 920 seconds and a lower accuracy of 88.2%, despite achieving a precision of 92.5% for phishing URLs. This research significantly contributes to phishing detection by developing a more robust and scalable approach using ensemble learning techniques. The Random Forest classifier, with its superior performance (accuracy of 96.6%, AUC of 97%, and balanced precision and recall), provides a highly effective solution for real-world phishing detection. By leveraging a large dataset, addressing class imbalance, and utilizing a comprehensive evaluation framework, this study offers a substantial improvement over existing methods. These findings serve as a foundation for future research to further enhance phishing detection models, ensuring that they remain effective in the face of evolving cyber threats.

**References**

Alshingiti, Z., Alaqel, R., Al-Muhtadi, J., Haq, Q.E.U., Saleem, K. and Faheem, M.H. (2023). A Deep Learning-Based Phishing Detection System Using CNN, LSTM, and LSTM-CNN. Electronics, 12(1), p.232. doi: <https://doi.org/10.3390/electronics12010232>

Alsariera, Y. A., Adeyemo, V. E., Balogun, A. O., and Alazzawi, A. K. (2020). AI meta-learners and extra-trees algorithm for the detection of phishing websites. IEEE Access 8, 142532–142542. doi:10.1109/ACCESS.2020.3013699

Aljofey, A., Jiang, Q., Rasool, A., Chen, H., Liu, W., Qu, Q. and Wang, Y. (2022). An effective detection approach for phishing websites using URL and HTML features. Scientific Reports, 12(1). doi: <https://doi.org/10.1038/s41598-022-10841-5>

Alkhalil, Z., Hewage, C., Nawaf, L. and Khan, I. (2021). Phishing Attacks: A Recent Comprehensive Study and a New Anatomy. Frontiers in Computer Science, 3(1). doi: <https://doi.org/10.3389/fcomp.2021.563060>

Alam, M.N., Sarma, D., Lima, F.F., Saha, I., Ulfath, R.-E. - and Hossain, S. (2020). Phishing Attacks Detection using Machine Learning Approach. 2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT). doi: <https://doi.org/10.1109/icssit48917.2020.9214225>

Almeida, R. and Carlos Becker Westphall (2020). Heuristic Phishing Detection and URL Checking Methodology Based on Scraping and Web Crawling. doi: <https://doi.org/10.1109/isi49825.2020.9280549>

Balogun, A.O., Adewole, K.S., Raheem, M.O., Akande, O.N., Usman-Hamza, F.E., Mabayoje, M.A., Akintola, A.G., Asaju-Gbolagade, A.W., Jimoh, M.K., Jimoh, R.G. and Adeyemo, V.E. (2021). Improving the phishing website detection using empirical analysis of Function Tree and its variants. Heliyon, [online] 7(7), p.e07437. doi: <https://doi.org/10.1016/j.heliyon.2021.e07437> =

Chiew, K.L., Chang, E.H., Sze, S.N. and Tiong, W.K. (2015). Utilisation of website logo for phishing detection. Computers & Security, 54, pp.16–26. doi: <https://doi.org/10.1016/j.cose.2015.07.006>

Daengsi, T., Pornpongtechavanich, P. and Wuttidittachotti, P. (2021). Cybersecurity awareness enhancement: A study of the effects of age and gender of thai employees associated with phishing attacks. *Education and Information Technologies*, 27(4). doi: <https://doi.org/10.1007/s10639-021-10806-7>

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>

Florian Quinkert, Degeling, M. and Holz, T. (2021). Spotlight on Phishing: A Longitudinal Study on Phishing Awareness Trainings. *Lecture notes in computer science*, pp.341–360. doi: <https://doi.org/10.1007/978-3-030-80825-9_17>

How to Deal with Imbalanced Datasets with SMOTE algorithm. (2024). <https://www.turing.com/kb/smote-for-an-imbalanced-dataset>

Jensen, M.L., Dinger, M., Wright, R.T. and Thatcher, J.B. (2017). Training to Mitigate Phishing Attacks Using Mindfulness Techniques. *Journal of Management Information Systems*, 34(2), pp.597–626. doi: <https://doi.org/10.1080/07421222.2017.1334499>

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. and Gupta, B.B. (2018). Rule-Based Framework for Detection of Smishing Messages in Mobile Environment. *Procedia Computer Science*, 125, pp.617–623. doi: <https://doi.org/10.1016/j.procs.2017.12.079>.

Kapan, S. and Sora Gunal, E. (2023). Improved Phishing Attack Detection with Machine Learning: A Comprehensive Evaluation of Classifiers and Features. Applied Sciences, [online] 13(24), p.13269. doi: <https://doi.org/10.3390/app132413269>

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*, 46(6), pp.1316–1339. doi: <https://doi.org/10.1177/0162243921992844>

Lakshmi, V.S. and Vijaya, M.S. (2012). Efficient prediction of phishing websites using supervised learning algorithms. Procedia Engineering, 30, pp.798–805. doi: <https://doi.org/10.1016/j.proeng.2012.01.930>

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>

Moghimi, M. and Varjani, A.Y. (2016). New rule-based phishing detection method. *Expert Systems with Applications*, 53, pp.231–242. doi: <https://doi.org/10.1016/j.eswa.2016.01.028>

Mourtaji, Y., Bouhorma, M., Alghazzawi, D., Aldabbagh, G. and Alghamdi, A. (2021). Hybrid Rule-Based Solution for Phishing URL Detection Using Convolutional Neural Network. *Wireless Communications and Mobile Computing*, 2021, pp.1–24. doi: <https://doi.org/10.1155/2021/8241104>

Nataraj, K.R., Yashaswini, D.K., R. Hema, Pawar, N.J. and S. Yashaswi (2023). Phishing Attack Detection Using Machine Learning. Lecture notes in electrical engineering, pp.355–370. doi: <https://doi.org/10.1007/978-981-99-2058-7_33>

Page, M.J., McKenzie, J.E., Bossuyt, P.M., Boutron, I., Hoffmann, T.C., Mulrow, C.D., Shamseer, L., Tetzlaff, J.M., Akl, E.A., Brennan, S.E., Chou, R., Glanville, J., Grimshaw, J.M., Hróbjartsson, A., Lalu, M.M., Li, T., Loder, E.W., Mayo-Wilson, E., McDonald, S. and McGuinness, L.A. (2021). The PRISMA 2020 statement: an Updated Guideline for Reporting Systematic Reviews. *British Medical Journal*, 372(71). doi: <https://doi.org/10.1136/bmj.n71>

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., et al. (2011). Scikit-learn: machine learning in Python. J. Mach. Learn. Res. 12, 2825–2828. doi: 10.48550/arXiv.1201.0490

Rao, R.S. and Ali, S.T. (2015). PhishShield: A Desktop Application to Detect Phishing Webpages through Heuristic Approach. Procedia Computer Science, 54, pp.147–156. doi: <https://doi.org/10.1016/j.procs.2015.06.017>

Refaeilzadeh, P., Tang, L. and Liu, H. (2009). Cross-Validation. Encyclopedia of Database Systems, [online] pp.532–538. doi: <https://doi.org/10.1007/978-0-387-39940-9_565>

Safi, A. and Singh, S. (2023). A systematic literature review on phishing website detection techniques. Journal of King Saud University - Computer and Information Sciences, 35(2). doi: <https://doi.org/10.1016/j.jksuci.2023.01.004>

Saha, I., Sarma, D., Chakma, R.J., Alam, M.N., Sultana, A. and Hossain, S. (2020). Phishing Attacks Detection using Deep Learning Approach. [online] IEEE Xplore. doi: <https://doi.org/10.1109/ICSSIT48917.2020.9214132>

Sattari, Y., and Montazer, G. (2023). Intelligent methods in phishing website detection: a systematic literature review. Research Square [Preprint]. doi: 10.21203/rs.3.rs-2518632/v1

Silva, C.M.R. da, Feitosa, E.L. and Garcia, V.C. (2020). Heuristic-based strategy for Phishing prediction: A survey of URL-based approach. Computers & Security, [online] 88, p.101613. doi: <https://doi.org/10.1016/j.cose.2019.101613>

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

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>

Zieni, R., Massari, L. and Calzarossa, M.C. (2023). Phishing or Not Phishing? A Survey on the Detection of Phishing Websites. IEEE Access, 11, pp.18499–18519. doi: <https://doi.org/10.1109/access.2023.3247135>

**Appendix A**

**Access the code from this link:** **https://github.com/MDISLAM24/Final-Project**