ENHANCING PHISHING DETECTION THROUGH MACHINE LEARNING

By

Zain ul Abidin

Fall 2020-NUST-MS-IS-329412

Supervisor

Dr. Hasan Tahir Butt

Department of Computing

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science in Information Security (MSIS)

In

School of Electrical Engineering and Computer Science,

National University of Sciences and Technology (NUST),

Islamabad, Pakistan

(March 2024)

THESIS ACCEPTANCE CERTIFICATE

Certified that final copy of MS/MPhil thesis entitled "Enhancing Phishing Detection through Machine Learning" written by Zain Ul abidin, (Registration No 00000329412), of SEECS has been vetted by the undersigned, found complete in all respects as per NUST Statutes/Regulations, is free of plagiarism, errors and mistakes and is accepted as partial fulfillment for award of MS/M Phil degree. It is further certified that necessary amendments as pointed out by GEC members of the scholar have also been incorporated in the said thesis.

Signature: ______________________

Name of Advisor: Dr. Hasan Tahir

Date: 31-Jan-2024

HoD/Associate Dean: ______________________

HOD Information Security
SEECS-NUST
Sector H-12, Islamabad

Date: 12-02-2024

Signature (Dean/Principal): ______________________

Dr. Muhammad Ajmal
Principal
NUST School of Electrical
Engg & Computer Science
H-12, Islamabad

Date: 12 Feb, 2024

Approval

It is certified that the contents and form of the thesis entitled "Enhancing Phishing Detection through Machine Learning" submitted by Zain Ul abidin have been found satisfactory for the requirement of the degree

Advisor : Dr. Hasan Tahir

Signature: ______________________

Date: 31-Jan-2024

Committee Member 1:Dr. Qaiser Riaz

Signature: QaiserRiaz

01-Feb-2024

Committee Member 2:Dr. Mehdi Hussain

Signature: ______________________

Date: 01-Feb-2024

Signature: zain

Date: 01-Feb-2024

DEDICATION

I dedicate this thesis to my family and faculty, who supported and motivated me to embark on this journey and has been instrumental in my ultimate success.

Certificate of Originality

I hereby declare that this submission titled "Enhancing Phishing Detection through Machine Learning" is my own work. To the best of my knowledge it contains no materials previously published or written by another person, nor material which to a substantial extent has been accepted for the award of any degree or diploma at NUST SEECS or at any other educational institute, except where due acknowledgement has been made in the thesis. Any contribution made to the research by others, with whom I have worked at NUST SEECS or elsewhere, is explicitly acknowledged in the thesis. I also declare that the intellectual content of this thesis is the product of my own work, except for the assistance from others in the project's design and conception or in style, presentation and linguistics, which has been acknowledged. I also verified the originality of contents through plagiarism software.

Student Name:Zain Ul abidin

Student Signature: Zain

FORM TH-4

National University of Sciences & Technology

MASTER THESIS WORK

We hereby recommend that the dissertation prepared under our supervision by: (Student Name & Reg. #) Zain Ul abidin [00000329412]

Titled: Enhancing Phishing Detection through Machine Learning

be accepted in partial fulfillment of the requirements for the award of Master of Science (Information Security) degree.

Examination Committee Members

1. Name: Qaiser Riaz Signature: 01-Mar-2024 11:00 AM

2. Name: Mehdi Hussain Signature: 01-Mar-2024 11:00 AM

Supervisor's name: Hasan Tahir Signature: 01-Mar-2024 2:47 PM

HoD/Associate Dean

07 Mar 2024
Date

COUNTERSINGED

07 Mar 2024
Date

Dr. Muhammad Ajmal
Principal
NUST School of Electrical
Engg & Computer Science
H-12, Islamabad

Dean/Principal

ACKNOWLEDGMENT

First and foremost, I would like to express my heartfelt gratitude to the Almighty Allah, who has bestowed upon me not only this achievement but also guided and blessed me in all the achievements of my life.

Furthermore, I extend my deepest gratitude to my research supervisor, Dr. Hasan Tahir Butt, for his invaluable guidance, expertise, and unwavering support throughout this research journey. His insightful feedback and encouragement have been pivotal in shaping the direction of my work and elevating its quality.

I would also like to extend my sincere appreciation to my family, especially my beloved parents, whose unconditional love, prayers, and support have been a constant source of inspiration and motivation. Their encouragement, patience, and unwavering belief in my abilities have been the driving force behind my success. I am truly grateful for the sacrifices they have made to provide me with the necessary resources and create a nurturing environment for me to pursue my academic goals.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

In the current digital environment, the prevalence of phishing attacks, which use social engineering to unlawfully obtain sensitive data like user credentials and personal information, is on the rise. This increase highlights the need for more advanced detection methods. Traditional phishing detection strategies are usually more effective with smaller datasets and often suffer from high computational demands due to their reliance on numerous features, limiting scalability in machine learning applications.

This research introduces a new method employing five well-known machine learning algorithms: Logistic Regression, Random Forest, Gradient Boosting, XGBoost, and LightGBM. The goal is to create a general framework for analyzing large-scale phishing data. An extensive dataset of 274,131 phishing URL entries has been compiled from sources like Kaggle, PhishTank, and OpenPhish. This dataset covers a wide range of URL categories, including Benign, Defacement, Phishing, Malware, and Spam, offering a broad foundation for the detection model.

A thorough preprocessing of the data was conducted to correct common issues such as incorrect formats, duplicates, broken links, and domain-only URLs, ensuring the dataset's quality for machine learning. A key aspect of this approach is the use of a relatively small set of features, even with larger datasets, addressing a major limitation of previous methods.

The processed data underwent extraction, optimization, and evaluation within the proposed machine learning frameworks. The findings of this research are notable, showing that the new methodologies outperform existing techniques in detection accuracy, handling of large data volumes, and efficiency in feature use. Experimental results show especially high accuracy in phishing URL detection, with algorithms like Random Forest, Gradient Boosting, XGBoost, and LightGBM achieving up to 98% accuracy in identifying phishing URLs within the substantial 274,131 URL dataset.

Keywords: Phishing Detection, Social Engineering, Machine Learning, URL Classification, Supervised Learning, Large-scale Dataset Analysis.

CHAPTER 1

INTRODUCTION

This chapter establishes the foundation of the research, providing insight into the main theme and essence of the thesis. It navigates the reader through the document's structure, offering clarity on the forthcoming content. The chapter explores the context of phishing threats, identifies key challenges, and delineates proposed solutions. It also discusses the research's motivations, its intended contributions to the scholarly domain, expected benefits of the work, and the study's primary objectives.

1.1 Background

Phishing campaigns pose a major global cybersecurity issue. They are designed to deceive individuals into revealing important personal details, like login and financial information, by pretending to be legitimate organizations across digital mediums. The growing complexity and agility of these malevolent schemes raise serious concerns. Criminals [1] frequently impersonate entities such as banks, popular online services, or government agencies. This enhances the complexity and prevalence of such illegal activities, further complicating the task for cybersecurity professionals.

Phishing incidents result in more than just minor inconveniences; they lead to significant financial losses. For instance, the Federal Bureau of Investigation's 2021 report [2] revealed that the total financial impact on American businesses and citizens exceeded an astonishing USD 6.9 billion. This figure includes not only the direct loss of money but also the additional costs incurred in mitigating the damages after successful phishing attacks. The substantial economic impact highlights the urgent need for more effective defensive measures.

Turning to potential remedies, machine learning (ML), a branch of artificial intelligence (AI), emerges as a significant tool. Essentially, ML [3] enables computers to learn from vast data sources, allowing continuous improvements in their functioning. A notable feature of ML systems is their capacity to analyze complex aspects in digital content, ranging from emails to web platforms, to identify possible malicious intentions.

Conventional anti-phishing strategies [4] based on blacklists or basic heuristic rules, show limitations in their effectiveness. The update frequency of these methods often falls behind the emergence of new malicious domains, resulting in false positives and missed threats. This shortfall has spurred interest in ML-based solutions. Their ability to extract insights from large datasets and identify subtle behavioral patterns provides them with a proactive edge in combating new online threats.

1.1.1 Phishing Attacks

Phishing fundamentally represents a type of cyber fraud that targets individual's psychology to illicitly extract sensitive information. It presents itself in various forms, each with its own communication channel and level of sophistication [5]:

a. Vishing

This involves a phone call where the caller creates an urgent or alarming situation to manipulate you into revealing confidential information. This tactic, known as 'vishing' relies on the power of voice communication [6] [7].

b. Smishing

Smishing is a deceptive strategy where fraudsters send text messages that appear to be from trustworthy entities. The purpose of these messages is to manipulate individuals into disclosing private information or interacting with unsafe links. A typical instance of this method includes messages that urgently request users to verify banking details or misleadingly inform them of winning a lottery. These deceitful texts are crafted to exploit vulnerabilities and extract sensitive data.

c. Email Phishing

One of the most common forms, this method uses deceptive emails that look highly authentic. These emails are favored by scammers due to their anonymity, wide reach, low cost, and scalability [8]–[10].

d. Spear Phishing

This more sophisticated form targets specific groups or individuals with tailored tactics. For example, you might receive a fake email from 'HR' requesting account updates. This targeted deception is known as 'spear phishing'.

e. Whaling

'Whaling' targets high-profile individuals, like company executives, aiming to access critical business information.

f. Clone Phishing

In this scenario, receiving an email that closely resembles a previous legitimate one, but with a malicious twist – a harmful attachment or link replaces the original content. This tactic of duplicating and altering a legitimate email is known as 'clone phishing'.

g. Pharming

Differing from other methods, 'pharming' redirects you to a fraudulent website, even if you type in the correct URL. This tactic involves creating fake websites that appear legitimate to collect your data.

Recognizing these phishing methods is crucial, as their simplicity can sometimes mask their effectiveness, as shown in various studies. [11] [12] Understanding their strategies not only highlights their potential dangers but also prepares for preventive actions.

As phishing tactics become more ingenious, there's an increasing need for a comprehensive cybersecurity strategy, combining advanced technology, user education, and vigilance against unexpected or suspicious communications. This holistic approach is vital for individuals and organizations to effectively combat the persistent threat of phishing.

1.2 Evolution of Phishing Attacks

The digital age, while bringing unparalleled connectivity, has also given rise to sophisticated threats that capitalize on the intersections of technology and human behavior. Phishing is one such peril, morphing over time to exploit human tendencies and technical blind spots more effectively. These scams have from rudimentary website clones to nuanced cons, using clever storytelling coupled with technological ruses. And as these tricks continue to evolve, conventional defense systems sometimes find themselves playing catch-up [13]:

a. **Business Email Compromise (BEC)**

   BEC specifically targets businesses, where attackers impersonate company executives or colleagues. They concoct elaborate stories to trick employees into making wire transfers or divulging sensitive information. The deceptive realism of the attacker's email and narrative makes BEC particularly challenging to identify and counter [14] [15].

b. **Ransomware Attacks**

   These attacks represent a severe escalation in cyber threats. Attackers encrypt victims' data, blocking access, and then demand a ransom, often in untraceable cryptocurrency, for its release. Typically, an innocuous-looking phishing email is used to launch these ransomware attacks, trapping the victim in a digital hostage situation.

c. **Cryptocurrency Cons**

   In the frenzy of digital currency, fraudsters have found fertile ground for scams. They entice potential investors with promises of profitable cryptocurrency investments. The scams range from selling fake digital currencies to tricking individuals into transferring legitimate cryptocurrencies into the fraudster's wallet. The anonymity and high stakes involved in cryptocurrency transactions make these scams particularly complex and risky.

The most effective defense against such cyber-attacks is awareness. Keeping informed about the latest phishing tactics, carefully verifying emails before responding to them, and practicing strong cyber hygiene are crucial strategies in the digital world. These steps are key in staying one step ahead in this ever-evolving landscape of online threats.

1.3 Machine Learning for Phishing Detection

Machine learning presents a promising approach for detecting phishing attempts. Leveraging machine learning's capacity to analyze complex patterns in large datasets and adapt to emerging threats positions it as an ideal tool for identifying phishing. As part of artificial intelligence, machine learning is increasingly recognized as a solution to the escalating phishing problem. [16]. This technology enables computers to learn from extensive data and iteratively enhance their capabilities without explicit programming. The paper explores the application of machine learning in strengthening phishing detection by analyzing features, identifying patterns, and adjusting to novel threats. Despite certain challenges, incorporating machine learning into cyber security strategies shows significant potential in diminishing phishing's impact and curtailing financial losses [17].

This introduction sets the stage for a detailed exploration of phishing attacks, their consequences, and the crucial role machine learning plays in combating this ever-evolving cyber security menace. The following sections explore the complexities of phishing, the potential of machine learning in addressing these challenges, and the important considerations and hurdles associated with this advanced approach in cyber security.

Machine learning offers a variety of methods to detect and counteract phishing attacks. Here are some notable applications:

a. **Email Analysis**

   Machine learning can scrutinize various email attributes, such as the sender's address, subject, and content, to identify potential red flags. For example, algorithms might flag emails from unknown sources, those containing specific keywords, or those with unusual attachments [18].

b. **Website Analysis**

   Machine learning can be used to inspect various website characteristics, like its URL, HTML structure, and content, to spot potential anomalies. For instance, algorithms might flag websites with URLs resembling legitimate sites, those containing specific terms, or those with unusual design elements [19].

c. **User Behavior Analysis**

   Machine learning can monitor user actions, like the sites users visit, the links they click on, and the information they enter online, to identify potential risks. For example, algorithms might flag users frequently visiting phishing sites or entering sensitive information on questionable platform.

1.4 Thesis Motivation

The motivation of research is rooted in the growing challenge of social engineering and phishing attacks, which pose significant threats to individuals, businesses, and the overall security of digital platforms. In an era where technological interconnectivity is the norm, safeguarding online privacy and data protection has become crucial. The advent of the digital age, while bringing revolutionary changes, has also introduced numerous security challenges.

This study aims to contribute to the field of cyber security by investigating the effects of phishing attacks. It involves analyzing historical and current data related to phishing, intending to highlight the severity of this threat and underline the urgent need for effective defense strategies. The rapid increase in internet users and the proliferation of online platforms have complicated the task of identifying reliable sources, leaving users more vulnerable to sophisticated phishing tactics.

Moreover, this study embarks on a journey to scrutinize existing anti-phishing tools. While commendable efforts like Spoof Guard, Netcraft Anti-Phishing Toolbar, and Google safe browsing have emerged to defeat phishing sites, the ever-adapting strategies of cyber adversaries necessitate a rigorous evaluation of these tools' efficacy and their inherent shortcomings. Such an evaluation accentuates the dire need for cutting-edge, real-time anti-phishing interventions.

Venturing further, this research is poised to navigate the realm of machine learning in anti-phishing, with a keen interest in models anchored in URL analysis for detecting phishing attempts. In this context, the paper [20] "Classification of Malicious Websites Using Machine Learning Based on URL Characteristics" by Muon Ha et al. emerges as a pivotal reference. Their study meticulously evaluates machine learning classification algorithms' prowess in pinpointing malicious websites through URL analysis. By harnessing a robust dataset of URLs and focusing on a spectrum of malicious websites, including phishing, they achieved an impressive accuracy rate of 95.68% using the Random Forest algorithm. Their practical application, spanning web applications and browser extensions, underscores the real-world viability of such machine learning-driven solutions. Their methodology and findings resonate with the proposed research direction, emphasizing the potential of machine learning, especially the Random Forest algorithm, in cybersecurity.

This analysis aims to examine the strengths and weaknesses of current models to discover innovative ways to improve and advance phishing detection's accuracy and speed. The primary goal is to develop strong anti-phishing tools capable of proactively countering phishing threats. By addressing gaps in existing research, proposing improvements, and revealing various attack methods, the objective is to enhance online security measures and protect users from the financial and privacy risks associated with phishing.

Inspired by L. Bustio et. al.'s [21] groundbreaking research on developing a URL feature set to enhance phishing detection in resource-constrained IoT environments, this study seeks to reinterpret and adapt their anti-phishing framework for identifying fraudulent websites. The focus is on creating a flexible, comprehensive, and scalable machine learning model for phishing detection, prioritizing a concise but effective feature set suitable for large datasets. The goal is to leverage a rich dataset and an optimized set of features to attain exceptional classification accuracy using traditional machine learning techniques, thereby surpassing the standards established by prior research.

1.5 Research Objectives

Research objectives are the specific, targeted goals set by a researcher for their study. These objectives form the core of the research, providing direction and purpose. Depending on the study's subject, these objectives can vary but must be clear, measurable, achievable, and relevant. They assist in data collection, result interpretation, and conclusion formation. In this investigation, the emphasis is on an in-depth exploration, analysis, and creation of effective solutions for a specified issue using stringent scientific methods. The main goal is to develop a comprehensive and effective approach to tackle phishing, exploring various aspects to fulfill these key objectives:

a. **Developing an Advanced Features Set of URL**

   The goal here is to create an advanced collection of URL features that accurately reflect key aspects of web interactions. These features will be finely tuned to enhance their relevance and effectiveness, thus improving the accuracy of any analysis or detection tools that utilize them [22]–[24].

b. **Leveraging a Comprehensive Dataset for Thorough Assessment:**

   This objective involves using a large dataset, consisting of 274,131 instances, to thoroughly test the effectiveness of the proposed models. This comprehensive dataset is critical for ensuring that the findings are robust and meaningful.

c. **Contrasting Efficiency across a Spectrum of Machine Learning Paradigms**

   The study aims to compare how the developed URL features perform across various machine learning algorithms. This will help identify the strengths and weaknesses of each algorithm and determine which ones are most compatible with the URL features.

d. **Assessing Precision through Diverse Scoring Mechanisms and Matrices**

   The final goal is to rigorously assess the models' precision using different scoring methods and matrices. This multifaceted evaluation will provide a comprehensive view of the model's performance, highlighting areas of strength and those needing improvement.

1.6 Research Questions

The next section outlines the research questions set up for the implementation of this study:

a. **What necessitates this research?**

   Over the past three decades, the field of Information Technology has experienced swift progress. This growth, especially in web applications and their user base, has significantly heightened website security threats. Phishing attacks have emerged as a primary challenge in this context. Despite the development of various machine-learning-based phishing detection systems, they frequently fail to cover all potential threats comprehensively. Therefore, it's crucial to persistently evaluate, refine, and reassess current methods. This research thoroughly examines previous approaches in this domain and proposes a new, agile, and effective strategy to combat phishing.

b. **Why is this study crucial, and what constitutes its research process?**

   This research introduces a technique to discern and categorize phishing websites through a machine-learning lens. This strategy aids in the prompt identification of phishing sites, safeguarding users from potential threats. The research process encompasses the subsequent stages:

   1. Amassing a substantial dataset.
   2. Extracting URL Features (lexical, significance, among others).

3. Refining the extracted feature set.
4. Conducting experiments and documenting results.
5. Analyzing and contrasting the outcomes.
6. Drafting the thesis report.

c. What objectives does this research aim to fulfill?

The primary goals of this investigation include:

- Crafting an agile and expandable mechanism for URL phishing detection.
- Identifying the optimal combination of features that can achieve the highest levels of accuracy in processing extensive datasets is essential (274,131 values).

d. How does this research differ from existing methodologies?

While numerous strategies have been proposed in the past, this study emphasizes a unique combination of agility and potency. By leveraging advanced machine learning techniques and focusing on a comprehensive feature set, it aims to bridge the gaps left by previous methods, offering a more holistic and efficient solution.

e. What potential impact might the findings of this study have on the broader IT community?

The outcomes of this research could revolutionize the way phishing attacks are detected and prevented. By providing a more efficient and scalable solution, it has the potential to set new standards in web security, benefiting both businesses and individual users alike.

1.7 Problem Statement

The escalating threat of phishing attacks poses a significant risk to individuals, businesses, and the wider digital landscape. As the internet continues to grow, distinguishing between legitimate and malicious websites becomes increasingly challenging, leaving users more susceptible to these deceptive attacks. Although tools like Spoof Guard, Netcraft's toolbar, and Google's Safe Browsing exist, cybercriminals are often ahead, devising ways to circumvent these protections. Current machine learning models, particularly those analyzing URLs, lack the desired level of foolproof effectiveness. They require further refinement to enhance their precision and response speed.

The primary objective of this study is to develop a robust, quick-response anti-phishing tool capable of detecting and stopping phishing attempts in real-time. This tool aims to overcome the limitations of existing defenses and adapt to the evolving tactics of cyber attackers. A significant emphasis is placed on improving feature selection and utilizing machine learning to increase detection speed and accuracy. To evaluate and validate the new models, we will use a large dataset of 274,131 entries. The goal is to establish a flexible, adaptable, and scalable machine learning solution that effectively identifies phishing attempts, using a streamlined set of URL-based features to provide a strong defense against phishing.

The research challenge can be summarized as follows: While many existing tools detect phishing with reasonable accuracy, they often rely on either excessive features or small datasets. We need a streamlined, universal method for phishing detection—one that uses a minimal yet effective set of features on a large dataset, all while aiming for high detection rates.

1.8 Thesis Solution and Contribution

In the research titled "Enhancing Phishing Detection through Machine Learning," significant advancements have been made to the benchmark methodology, achieving accuracy levels comparable to the benchmark with a dataset about twice its size. Data was collected in five categories: Benign, Malware, Spam, Phishing, and Defacement, from a variety of online sources. The focus was on URL analysis, extracting and refining features related to length, frequency, rate, linguistic patterns (using NLP), and overall significance, ultimately identifying 24 key features. Four different machine learning algorithms were utilized, leveraging Python and its comprehensive Machine Learning libraries. The techniques tested included Logistic Regression, Random Forest, Gradient Boosting, XGBoost, and LightGBM.

The key advancements of methodology include:

- **a.** Attaining a phishing detection accuracy that matches the established benchmark, even after doubling the size of the dataset.
- **b.** Implementing an advanced method for identifying phishing in web URLs, which employs a carefully selected group of features in supervised machine learning frameworks.
- **c.** Undertaking an in-depth assessment, comparing the effectiveness and precision of five different algorithms used on a large dataset.

1.9 Thesis Organization

The structure of this thesis is organized as follows: Chapter 2 presents the foundational elements of the study, emphasizing recent literature and significant contributions in anti-phishing detection. Chapter 3 details the research methodology, including a comprehensive description of the proposed framework, the chosen feature set, and the dataset used. Chapter 4 focuses on the experimental design and procedures, highlighting the effectiveness of the approach through the results. Chapter 5 conducts a thorough analysis of these results, comparing them with benchmark methods and current techniques. Chapter 6 concludes the research, providing insights and recommendations for future investigations.

1.10 Summary

This chapter serves as an introduction to the entire document and the research undertaken within this study. It underscores the importance of the research and the value added through its findings. Topics such as phishing attacks, the diverse methods employed to identify them, the specific challenges tackled, and the solutions put forth have been touched upon. Additionally, the goals, breadth, and unique contributions of this study are outlined. The following chapter is a review of relevant literature.

CHAPTER 2

LITERATURE REVIEW

This section delves into prior research, providing a critical analysis of the methodologies used in previous studies. The discussion covers the importance of URL elements, examines well-known machine learning models, and looks at various classification strategies. It also offers an overview of the latest advancements in phishing detection. The analysis identifies potential gaps in these studies and suggests possible areas for enhancement. To conclude, the section outlines the focus of the current research, reflecting on established methods and detailing how this approach intends to contribute to the field.

2.1 Anatomy and Significance of a URL

A Uniform Resource Locator (URL) [25] can be defined as a sequence of alphanumeric and symbolic characters, serving the pivotal role of delineating the precise address or spatial coordinates within the vast expanse of the internet for a given digital resource. Its primary purpose is to operate as an exclusive and distinct marker, facilitating the retrieval of webpages, textual documents, graphic representations, multimedia elements, and diverse digital artifacts that populate the online domain.

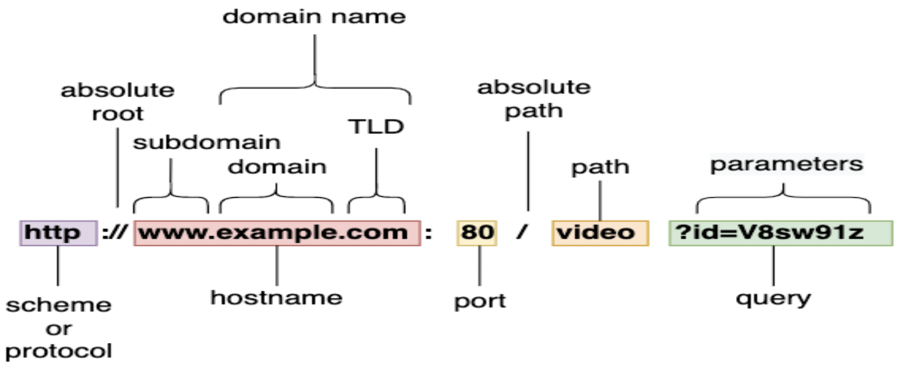


Figure 1: Anatomy of URL

A URL (Uniform Resource Locator) is composed of several components, each serving a specific function in directing a browser to a particular resource on the internet. The main elements of a URL [26] include:

a. Protocol

Positioned at the outset of a URL, elements such as "http://" or "https://" delineate the communication protocol utilized for resource access. These protocols, inclusive of HTTP (Hypertext Transfer Protocol) and HTTPS (HTTP Secure), serve as standardized conduits for retrieving web pages, encompassing a range of digital assets.

b. Domain Name

In a URL such as "https://www.example.com," the domain name, here "example.com," assumes a pivotal role in identifying the specific web entity or server entrusted with hosting the resource. This hierarchical nomenclature encompasses the top-level domain (TLD), indicative of organizational type or country affiliation (e.g., .com, .org, .gov, .uk), and cascades into the second-level domain (e.g., "example"), occasionally augmented by further subdomains (e.g., "www").

c. Path

The path, which immediately follows the domain name and is segregated by forward slashes ("/"), serves as a pointer to the particular location or directory within the web server where the desired resource is located. In the URL "https://www.example.com/path/to/resource.html," the path is "/path/to/resource.html."

d. Query Parameters

Optionally appended to a URL, query parameters are instrumental in transmitting supplementary information to the server. They are demarcated by a question mark ("?") within the URL and are further separated by ampersands ("&"). Comprising key-value pairs, with keys and values differentiated by the equal sign ("="), query parameters enable nuanced customization. For instance, in the URL "https://www.example.com/search?q=example&page=1," the query parameters are "q=example" and "page=1."

The significance of the URL's structural composition cannot be understated, as it constitutes a foundational element in the context of resource retrieval and organization within the vast terrain of the World Wide Web. A judiciously crafted URL not only imparts lucidity and significance to the resource's location but also enhances the navigational experience for users. Furthermore, it facilitates the indexing and comprehension of web content by search engines, thereby contributing to the realm of search engine optimization (SEO). Concurrently, the URL's structural elegance exerts a tangible influence on website usability, where concise, descriptive URLs enhance user-friendliness, ease of sharing, and recall.

2.2 Classification and Machine Learning Algorithms

Machine learning algorithms [27]–[29] have emerged as pivotal tools in the domain of phishing detection, capitalizing on their inherent capabilities to learn from vast datasets, discern intricate patterns, and adapt to the ever-evolving landscape of cyber threats. Their application in phishing detection is geared towards the creation of sophisticated, automated systems that can proficiently identify and categorize phishing endeavors.

Phishing is a deceptive technique wherein adversaries masquerade as trustworthy entities, aiming to mislead individuals into divulging confidential information. This could range from login credentials to financial details. The integration of machine learning [30]–[32] into cyber security frameworks augments traditional protective measures, bolstering an organization's resilience against phishing onslaughts and thereby reducing the likelihood of users succumbing to such malevolent schemes.

In the context of this study, several machine learning algorithms are explored for their efficacy in phishing detection:

a. Logistic Regression

At its core, Logistic Regression is a statistical method designed for binary classification tasks. Unlike linear regression, which predicts a continuous outcome, Logistic Regression predicts the probability that a given instance belongs to a particular category. The algorithm employs the logistic function to squeeze the output of a linear equation between 0 and 1. The coefficients of the linear equation are derived from the training data using maximum likelihood estimation. One of its strengths is interpretability; each feature's coefficient indicates its importance and direction of association with the target variable. In the context of phishing detection, Logistic Regression can be employed to estimate the likelihood that a given URL or email is malicious based on various features.

b. Random Forest

Random Forest is an ensemble learning technique that amalgamates the predictions of multiple decision trees to produce a more accurate and stable outcome. Each tree in the forest is constructed using a subset of the training data, chosen with replacement (bootstrapping). Additionally, when splitting nodes, only a random subset of features is considered, ensuring tree diversity. This method counteracts the tendency of individual trees to over fit the data. The final prediction is an aggregation, typically the mode (for classification) or mean (for regression) of the predictions of all trees. Its inherent ability to rank the importance of features and its robustness against overfitting make Random Forest a valuable tool in phishing detection.

c. Gradient Boosting

Gradient Boosting is a sequential ensemble method where trees are added one at a time, and existing trees in the model are not changed. Each tree corrects the errors of its predecessor. The term "gradient" in its name stems from the fact that the algorithm uses gradient descent to minimize the loss. By adjusting the model in the direction of the steepest decrease of the loss function, Gradient Boosting iteratively refines its predictions. This meticulous, step-by-step refinement process often results in high accuracy, albeit at the cost of increased computational time.

d. XGBoost (Extreme Gradient Boosting)

An optimized distributed gradient boosting library, XGBoost is designed to be highly efficient, flexible, and portable. It not only supports the traditional gradient boosting algorithm but also allows for regularization, which helps in reducing overfitting. Furthermore, XGBoost offers several advanced features, such as handling missing values and in-built cross-validation. Its ability to automatically optimize its parameters and its robustness against outliers and non-linear relationships make it particularly suited for complex datasets, such as those encountered in phishing detection.

e. **LightGBM (Light Gradient Boosting Machine)**

An evolution of gradient boosting, LightGBM stands out for its efficiency and speed. It employs a novel technique of Gradient-based One-Side Sampling (GOSS) to filter out the data instances for finding a split value, and Exclusive Feature Bundling (EFB) to reduce the dimensionality of categorical features. This results in significant speed-ups, especially on larger datasets. Moreover, LightGBM is designed to distribute and run on distributed systems, making it scalable for massive datasets. Its ability to handle large-scale data without compromising on accuracy makes it a potent tool in the arsenal against phishing attacks.

2.3 Advantages of Machine Learning for Phishing Detection

Machine learning presents several superiorities in comparison to conventional phishing detection techniques [33]–[35]. These include:

a. **Precision -** through rigorous training, machine learning models have the potential to attain remarkable precision in identifying phishing incursions.
b. **Flexibility -** one of the salient features of machine learning models is their inherent ability to acclimate to emerging threats. This adaptability stems from their capacity to assimilate new information and refine their predictive models in response.
c. **Expandability -** a notable advantage of machine learning models is their capability to process vast datasets. This becomes particularly crucial in the context of the ever-escalating frequency of phishing attempts.

2.4 Limitations of Traditional Machine Learning Approaches

Traditional phishing detection mechanisms [36]–[38] such as blacklists and rule-based systems, have been the cornerstone of cyber security for a considerable duration. However, as cyber threats evolve, the limitations of these methods become increasingly pronounced.

a. **Dependence on Static Data**
   - **Blacklists:** These are essentially databases of known malicious URLs. The primary challenge with blacklists is their static nature. They are effective only against previously identified threats. As phishing attackers frequently generate new malicious websites, blacklists often lag behind, failing to recognize these new threats [39].
   - **Rule-based Systems:** These systems operate based on predefined rules set by cyber security experts. While they can be effective against known patterns of phishing attacks, their static nature makes them less adept at identifying novel or slightly modified phishing strategies.

b. **Inefficiency against Zero-Hour Attacks**
   - Phishing attackers are becoming increasingly sophisticated, often launching what are termed as "zero-hour" attacks. These are attacks that are launched immediately after a new vulnerability is discovered, giving defenders zero hours to prepare or defend against them. Traditional methods like blacklists or heuristic-based approaches struggle against such attacks because they rely on prior knowledge of threats.

c. **False Positives and False Negatives**
   - One of the significant challenges with rule-based systems is their propensity to generate false positives and false negatives. A false positive occurs when a legitimate website is incorrectly flagged as malicious, leading to potential disruption for users and loss of trust in the detection system. Conversely, a false negative happens when a malicious site goes undetected, posing a direct threat to unsuspecting users [40].

d. **Lack of Real-time Adaptability**
   - The dynamic landscape of cyber threats necessitates real-time adaptability in detection systems. Traditional methods, due to their reliance on static data or predefined rules, often fail to adapt swiftly to emerging threats.

2.5 Challenges of Machine Learning for Phishing Detection

Here are some of the challenges that need to be addressed when using Machine Learning for phishing detection [41] [42]:

a. **Data Availability**
   - **Issue**

Machine Learning algorithms thrive on vast amounts of data. Their performance, especially in tasks like phishing detection, is directly proportional to the quality and quantity of the training data they are provided with. However, obtaining large datasets of labeled phishing and legitimate data can be both challenging and costly.

- **Implication**

  Without adequate data, ML models might not be trained sufficiently to recognize the nuanced patterns of phishing attacks, leading to decreased accuracy and reliability.

b. Model Complexity

- **Issue**

  As ML models are trained to recognize intricate patterns in data, they can become exceedingly complex. This complexity can make it challenging to interpret the model's predictions and discern which features are pivotal for detecting phishing attacks.

- **Implication**

  A lack of interpretability can hinder the trustworthiness of the model and make it difficult for cyber security experts to understand and validate the model's decisions.

c. Adversarial Attacks

- **Issue**

  Adversarial attacks are designed to deceive ML models by introducing subtle perturbations in the input data. These perturbations, though often imperceptible to humans, can cause the model to make incorrect predictions.

- **Implication**

  Such attacks can severely undermine the reliability of ML-based phishing detection systems, making them vulnerable to sophisticated cyber threats.

d. Evolving Nature of Phishing Attacks

- **Issue**

  Cyber attackers are in a constant race with defenders, always innovating and devising new phishing techniques. This dynamic nature of threats means that ML models need frequent updates to remain effective.

- **Implication**

  Without regular updates, even the most advanced ML models can become outdated and less effective over time.

e. **Targeted Phishing Attacks**
   - **Issue:** Some phishing attacks are tailored for specific individuals or organizations, making them harder to detect as they might not follow common patterns.
   - **Implication:** Models trained on general phishing patterns might struggle to detect these targeted attacks, leading to potential security breaches.

f. **Reliance on Social Engineering:**
   - **Issue**

     Many phishing attacks employ social engineering techniques, manipulating individuals into divulging confidential information. Detecting such attacks based solely on email or website features can be challenging.
   - **Implication**

     Relying solely on technical features might not be sufficient. There's a need for models that can also understand and detect social engineering tactics.

2.6 Related Work

In the scholarly article "Classification of Malicious Websites Using Machine Learning Based on URL Characteristics" by [20] Muon Ha and colleagues, the authors highlight into the realm of machine learning to discern malicious websites based on URL attributes. The study leverages a comprehensive dataset of URLs to train the classification model. The researchers categorize various malicious website types, such as phishing, defacement, and web-spam, and continually update their dataset with recent URL data to refine the classification accuracy. The methodology encompasses the categorization of website URLs into five distinct groups, from which 20 primary features are extracted. Subsequent machine learning techniques are employed to train classifiers. The study's findings reveal that the Random Forest algorithm stands out in terms of efficiency, boasting an impressive accuracy rate of 95.68% in pinpointing malicious websites. The researchers further validate their model's performance using renowned machine learning algorithms through a 10-fold cross-validation process. They employ performance metrics like precision, recall, accuracy, and F1-Score for evaluation. On a practical front, the team developed software tools in the form of web applications and browser extensions to detect and alert users about malicious websites. This dual approach, integrating the Random Forest method with a blacklist check, amplifies the software's efficiency. The process involves an initial check of the input URL against a database of known malicious URLs, followed by verification via the machine learning model. In summation, this research offers profound insights into the potential of machine learning algorithms in fortifying cyber security, particularly in the realm of malicious website detection. The outcomes not only underscore the prowess of the Random Forest algorithm but also highlight the successful real-world application of such algorithms. Future endeavors will pivot towards exploring diverse feature sets and expanding datasets to enhance the precision of malicious website detection.

In the research paper [43] "Phishing Website Detection Using Machine Learning Classifiers Optimized by Feature Selection", the authors introduce an innovative methodology to identify phishing websites employing machine learning strategies. The crux of their approach revolves around three classifiers: K-Nearest Neighbor (KNN), Decision Tree, and Random Forest. These classifiers undergo optimization using feature selection techniques derived from Weka, a renowned machine learning software suite. In the digital realm, the looming threat of security breaches, with phishing websites at the forefront, is undeniable. Such deceptive websites masquerade as legitimate entities, aiming to harvest confidential user data, making their differentiation from genuine sites a challenging task. Addressing this conundrum, the authors embark on the journey of classifying phishing websites. Their strategy, rooted in machine learning classifiers, achieves a remarkable accuracy rate of 100%, simultaneously reducing the feature count to a mere seven. Additionally, the model-building time witnesses a significant reduction, especially evident in the Random Forest classifier. This research stands as a testament to the field, showcasing a highly precise and efficient methodology for detecting phishing websites. The substantial reduction in feature count and model-building time further accentuates the solution's practicality. This scholarly work lays a robust foundation for subsequent studies, aiming to bolster online security measures and counteract phishing threats through machine learning methodologies.

In the article [44] "An Adversarial Attack Analysis on Malicious Advertisement URL Detection Framework", the authors present a cutting-edge technique to identify malicious advertisement URLs harnessing machine learning strategies. The cyber security landscape identifies malicious advertisement URLs as a critical concern, often acting as conduits for cyber-attacks. This necessitates a robust solution, both in the industrial and academic spheres. The authors critique existing malicious URL detection methodologies, highlighting their shortcomings in addressing unseen features and generalizing to test datasets. To bridge this gap, they introduce a unique set of lexical and web-scrapped features, leveraging machine learning techniques to establish a system adept at detecting fraudulent advertisement URLs. The proposed feature set encompasses six distinct types, meticulously addressing the obfuscation challenges in fraudulent URL classification. The detection, prediction, and classification tasks employ twelve uniquely formatted datasets, each with distinct statistical attributes. The prediction analysis further extends to mismatched and unlabeled datasets. Four machine learning techniques, namely Random Forest, Gradient Boost, XGBoost, and AdaBoost, undergo rigorous performance analysis for the detection phase. The proposed methodology boasts a minimal false negative rate of 0.0037, while maintaining an impressive accuracy rate of 99.63%. The authors also introduce an innovative unsupervised data clustering technique using the K-Means algorithm for visual analysis. Furthermore, they scrutinize the vulnerabilities of decision tree-based models under a limited knowledge attack scenario, incorporating the exploratory attack, and implement the Zeroth Order Optimization adversarial attack on the detection models. This scholarly work stands as a beacon in the field, offering an efficient and highly precise methodology for detecting malicious advertisement URLs. It also sheds light on the vulnerabilities of decision tree-based models under adversarial attacks, paving the way for future research endeavors aiming to bolster the resilience of machine learning models in cyber security applications.

In the research paper [45] "An Assessment of Lexical, Network, and Content-Based Features for Detecting Malicious URLs Using Machine Learning and Deep Learning Models", the authors embark on a journey to explore diverse feature types for the detection of malicious URLs, employing both machine learning and deep learning techniques. The escalating threat landscape, characterized by malicious URLs often used as vectors for malware or phishing campaigns, underscores the significance of this research endeavor. The paper's primary focus revolves around the evaluation of lexical, network, and content-based features in the context of URL classification tasks. The authors posit that these feature types are pivotal in discerning between benign and malicious URLs. Lexical features encompass elements derived from the URL string, network features originate from host-based data, and content-based features are extracted from the webpage content. The authors employ a plethora of machine learning and deep learning models to assess these features. While the initial sections of the paper do not explore into the specifics of the models used, the results unequivocally demonstrate the efficacy of the proposed approach. This research endeavor makes a monumental contribution by evaluating the influence of varied feature types on the performance metrics of URL classification models. The insights gleaned from this study can significantly aid in the development of robust systems adept at detecting malicious URLs, thereby fortifying cyber security measures.

In the scholarly article [46] "Analysis of the Performance Impact of Fine-Tuned Machine Learning Model for Phishing URL Detection", the authors study how to adjust machine learning models to better recognize phishing URLs. They focus on making these models more accurate for identifying these harmful web links. With the digital landscape witnessing a surge in phishing campaigns, exploiting individuals' propensity to divulge personal information online, the research presented in this paper stands as a cornerstone in fortifying cyber security measures. The authors highlight the modus operandi of phishing campaigns, often initiated via emails, wherein attackers employ social engineering tactics to entice the target to engage with the embedded phishing link. Such deceptive campaigns can be weaponised for a myriad of malicious intents. This research distinguishes itself by zeroing in on the performance ramifications of fine-tuning machine learning models, specifically tailored for the task of phishing URL detection. While the initial sections of the paper do not divulge the specifics of the machine learning models employed or the intricacies of the fine-tuning process, the authors' methodology promises to shed light on optimizing these models for niche cyber security tasks. This scholarly endeavor stands as a seminal reference for future research in the domain, emphasizing the enhancement of machine learning model performance in the realm of phishing URL detection. The meticulous exploration of fine-tuning techniques, specifically tailored for phishing URL detection, offers a fresh perspective, poised to significantly influence cyber security research and practices.

In the research paper [47] "APuML: An Efficient Approach to Detect Mobile Phishing Webpages using Machine Learning", the authors introduce APuML, a groundbreaking methodology designed to detect malicious mobile webpages. Given the ubiquity of mobile devices and their inherent security vulnerabilities, the authors' approach, which amalgamates static and site popularity features with machine learning algorithms, is both timely and pertinent. The research findings underscore the prowess of the Random Forest classifier, achieving a commendable detection accuracy of 93.85%. This scholarly work offers a significant contribution to the realm of cyber security by proposing an efficient machine learning-centric approach to detect malicious mobile webpages. The results not only highlight the precision of phishing detection but also pave the way for future research endeavors aiming to bolster mobile security through machine learning techniques.

In the article [48] "CCrFS: Combine Correlation Features Selection for Detecting Phishing Websites Using Machine Learning", the authors introduce CCrFS, a pioneering methodology tailored for the detection of phishing websites. Recognizing the escalating threat landscape characterized by phishing campaigns, the authors' approach emphasizes feature selection based on correlations, offering a fresh perspective on enhancing machine learning models for phishing detection. This research endeavor stands as a beacon in the field, potentially serving as a reference for subsequent studies aiming to bolster cyber security measures against phishing threats. The unique focus of the CCrFS approach on feature selection rooted in correlations promises to offer valuable insights, poised to significantly influence the development of machine learning models tailored for phishing detection.

In the scholarly article [49] "Cyber Threat Intelligence-Based Malicious URL Detection Model Using Ensemble Learning" by Alsaedi and colleagues, the authors underscore the potential of amalgamating cyber threat intelligence with ensemble learning to enhance malicious URL detection. The authors' methodology, which employs multiple learning algorithms, promises to offer significant insights into bolstering web application security. This research endeavor offers a significant contribution to the realm of cyber security by introducing a practical and effective methodology to shield web applications and their users from the threats posed by malicious websites. The authors posit that future research endeavors could pivot towards leveraging larger datasets and exploring diverse feature sets to further refine the precision of malicious URL detection.

In the article [50] "Detecting Malicious URLs Using Machine Learning Techniques: Review and Research Directions", the authors offer a comprehensive review of machine learning techniques tailored for the detection of malicious URLs. Recognizing the escalating threat landscape, the paper provides invaluable research directions, emphasizing the need for advanced feature extraction techniques and robust learning algorithms.

In the research paper [51] "Machine-Learning-Based Android Malware Family Classification Using Built-In and Custom Permissions" by Minki Kim and colleagues, the authors explore into the potential of machine learning in classifying Android malware families. The authors' approach, which employs both built-in and custom permissions, promises to offer significant insights into bolstering Android malware detection.

In the article [52] "Significance of Machine Learning for Detection of Malicious Websites on an Unbalanced Dataset" by Ul Hassan and colleagues, the authors address the challenge of detecting malicious websites, which often misuse user data for nefarious purposes. The authors propose a machine learning-centric approach to differentiate between malicious and benign websites on an unbalanced dataset, a common challenge in cyber security. The insights gleaned from this study can significantly aid in the development of robust systems adept at detecting malicious websites, thereby fortifying cyber security measures.

In 2022, Lázaro Bustio-Martínez [21] and colleague's highlights into the development of a feature set aimed at enhancing the detection of phishing attempts in IoT devices. Their research introduced a feature selection algorithm tailored to pinpoint the most potent set of URL lexical features. Utilizing the proposed dataset representation, the Random Forest (RF) algorithm demonstrated an impressive accuracy of 99.57%.

Yukun Li and his team in [53] 2019, unveiled a sophisticated stacking model that leverages both URL and HTML-based attributes to identify phishing webpages. This model amalgamates the strengths of GBDT, XGBoost, and LightGBM in a multi-layered architecture, thereby amplifying detection capabilities. They employed two distinct real-world datasets for their evaluation: 50K-PD, encompassing 49,947 web pages, and 50K-IPD, housing 53,103 web pages, each replete with URLs and HTML scripts. Their findings revealed that the proposed methodology achieved an accuracy of 98.60% on the 50K-IPD dataset.

In a similar vein, Ammara Zamir [54] and associates in 2019, pioneered an innovative technique for detecting phishing websites. Their approach harnessed the power of feed-forward Neural Networks (NN) and various ensemble learning models. Despite relying on a relatively compact dataset of 11,055 entries, their method achieved a commendable accuracy rate of 97.30%.

A.A. Orunsolu and team in [55] 2019, proposed a refined machine learning model that extracts a 15-dimensional feature set from website URLs, webpage attributes, and behavioral patterns. This model, designed to bolster anti-phishing systems, employs SVM and NB classifiers in a component-based approach, achieving a stellar accuracy rate of 99.96%. However, it's worth noting that their research was constrained by a smaller dataset of 5,041 instances, a plethora of features, and a limited set of evaluation algorithms.

Lastly, Abdulhamit Subasia [56] and colleagues in 2020, crafted an intelligent framework that juxtaposed the accuracy of two ensemble models tailored for phishing detection on websites. Their research explain the URL lexical features, JavaScript attributes of web pages, and external factors like domain age and DNS records. Their findings spotlighted AdaBoost's superior performance over multi-boosting, registering an accuracy of 97.61% on a 20,000-entry dataset. However, it was evident that the dataset used was somewhat limited in scope, and the proposed methodology necessitated intricate computations for feature engineering.

In addition to that, surveys offer a more exhaustive overview of phishing URL detection techniques. The chosen classifiers were developed utilizing Python's renowned machine learning library, sci-kit-learn, to assess the proposed dataset. Furthermore, the machine learning algorithms were configured using their default parameters.

Research work	No of Features	Dataset Size	Algorithm	Accuracy %
M.Ha et. al. [20] (2023)	20	213,345	DT, RF, KNN, AdaBoost	94.83
L. Bustio et. al. [21] (2022)	9	50,000	RF, SVM, DT, KNN	99.57
A. Subasi, et. al. [56] (2019)	32	20,000	SVM, KNN, ANN, RF, REPT, RT	97.61
A. Zamir et. al. [54] (2020)	32	11,055	NB, KNN, SVM, RF, Bagging, NN	97.31
A.A. Orunsolu et. al. [55] (2019)	15	5,041	SVM, NB	99.60

Table 1: Recent Work on Phishing Detection with its Proposed Feature Numbers, Data Size, Models, and Achieved Accuracies

In the presented Table 1, a detailed comparison of recent phishing detection methodologies is outlined. This analysis considers several crucial factors, including the selection of features, the scale of the data sets utilized, and the specific machine learning algorithms employed, as well as their corresponding accuracy levels. A significant observation from this comparison is that many of these techniques predominantly aim for high accuracy, which sometimes leads to the selection of less optimal URL attributes. A notable limitation in these approaches is their reduced effectiveness when dealing with large-scale phishing URL data sets. This issue underscores the urgent need for a more universally applicable and scalable machine learning framework in phishing detection. Ideally, this framework should operate effectively with a concise yet potent set of features, ensuring reliable detection rates even when applied to extensive data sets. Such a framework is critical for enhancing cyber security measures in an increasingly digital world, where phishing attacks are becoming more sophisticated and widespread.

The principal contributions of this research include:

- **a.** A refined selection of URL-based features.
- **b.** A comprehensive dataset, encompassing 274,131 entries, to assess the efficacy of proposed models.
- **c.** A nimble, scalable model designed to achieve superior detection rates when paired with traditional machine learning classifiers.

2.7 Research Direction

The intricacies of phishing URL detection stem from the distinct attributes of URLs. Typically, URLs span about 65 characters in length. Interestingly, phishing URLs tend to be more extended, averaging 80 characters, in contrast to legitimate URLs, which are around 51 characters. Given that URLs are composed of arbitrary characters, they do not possess the conventional grammatical constructs and inherent meanings we usually associate with standard sentences. Further complicating matters is the striking resemblance between legitimate and phishing URLs, with the distinctions often being nuanced.

A survey of the existing literature unveils a myriad of strategies to categorize URLs as either legitimate or phishing. These methodologies harness a range of features for phishing detection, encompassing lexical examination, natural language processing (NLP), heuristic methods, and direct website scrutiny. Notwithstanding the diverse features employed, there's a consensus in the academic community about the potency of lexical scrutiny in the realm of URL-centric phishing detection. A plethora of classifiers have been evaluated to ascertain their efficacy and gauge their performance in varied scenarios. Traditional methods like K-Nearest Neighbor, Decision Tree, and Random Forest are compared with newer techniques like Deep Learning and Genetic Algorithms.

A critical examination of the training datasets highlighted in scholarly works indicates a pronounced emphasis on lexical attributes, like the count of characters and symbols. Some features highlights into Neural Networks (NN), Deep Learning, NLP methodologies, both Linear and non-linear spatial transformations, Ensemble strategies, and TF-IDF. Yet, only a handful of these representations truly leverage the inherent traits of each category, like Mutual Information or term entropy. Furthermore, the features and algorithms earmarked for phishing detection frequently hinge on intricate methodologies, extensive feature selection, and relatively confined datasets.

Given the gaps discerned from the literature, this research endeavor seeks to pioneer a streamlined URL representation tailored for real-time phishing URL detection, with a particular focus on website-based phishing. The envisaged methodology is predicated on the proven efficacy of URL-centric techniques that harness URL-based lexical attributes, a testament to the commendable classification precision documented in extant studies.

2.8 Summary

The literature review chapter provides an in-depth examination of the structure and significance of website URLs, explaining the function of each segment in a URL request. It offers a concise summary of classification techniques and machine learning algorithms, particularly focusing on Random Forest, K-Nearest Neighbors, Support Vector Machines, Decision Trees, Gradient Boosting, XGBoost, and LightGBM, all commonly used in phishing detection research. The chapter also explores recent scholarly work, the methodologies proposed, and the challenges addressed in the field of phishing detection. It critically evaluates existing methodologies, highlighting their limitations and potential improvements. This analysis sets the stage for the research outlined in the subsequent chapters. The next chapter will thoroughly explore the Research Methodology.

CHAPTER 3

RESEARCH METHODOLOGY

In this chapter, the research methodology used in this study is discussed in detail. This includes an overview of the classification workflow and the introduction of the proposed phishing URL indicators. The chapter goes into detail about the creation of the feature set and the thorough process of gathering the dataset.

3.1 Proposed Methodological Approach

A comprehensive examination of existing literature indicates that many studies have focused on analyzing webpage contents using Natural Language Processing (NLP) to extract features from website characteristics. This typically involves examining the website's code and visual components, which can be large in volume, inconsistent, and not always updated in real-time. Notably, even if potentially harmful websites are short-lived (active for less than 24 hours), the characteristics embedded in their URLs remain relevant. Lexical features derived from URLs are crucial as they tend to be more consistent and conform to specific standards compared to other types of features. Therefore, this study focuses on analyzing URL-based lexical attributes.

3.1.1 Dataset Composition and Categorization

a. Dataset Origin and Composition

The research is supported by a substantial dataset of 274,131 URLs, sourced from a wide range of platforms. These URLs provide a representative sample of the broader digital environment and are carefully categorized into five distinct groups, each representing a specific type of web activity:

- **Benign -** URLs that are harmless and do not exhibit any malicious intent.

- **Defacement -** URLs associated with websites that have been altered or defaced by unauthorized entities.
- **Phishing -** URLs designed to deceive users into sharing sensitive information under the guise of legitimate requests.
- **Malware -** URLs that host or redirect to malicious software intended to compromise user systems.
- **Spam -** URLs that promote unsolicited content, often for advertising purposes.

b. Data Sources and Integration

To ensure the dataset's comprehensiveness and relevance, malicious URLs were extracted from open-source platforms such as OpenPhish, Phishtank, Zone-H, and Kaggle. Each URL, encapsulated as a string of characters, is precisely paired with its corresponding label, and these datasets are then combined to form a comprehensive training set.

3.1.2 Machine Learning Classification

a. Algorithm Selection

For optimal classification results, the study employs five advanced machine learning algorithms: Logistic Regression, Random Forest, Gradient Boosting, XGBoost, and LightGBM. The selection of these algorithms was based on their proven effectiveness in similar classification tasks and their adaptability to the nuances of the dataset.

b. Feature Engineering and Optimization

Feature engineering is a key aspect of the methodology. Through iterative analysis and testing, a set of 24 critical URL features has been identified and optimized. These features include lexical attributes and structural patterns, refined to improve the predictive accuracy of the classification models.

3.1.3 Research Workflow

The research follows three main stages:

a. **Dataset Aggregation:** A comprehensive accumulation process, ensuring a diverse and representative set of URLs, each annotated with one of five distinct labels.

b. **Feature Engineering -** An iterative and methodical process, encompassing the identification, extraction, and optimization of URL-specific features, ensuring they capture the underlying patterns and nuances.

c. **Classifier Implementation -** The deployment phase, wherein tailored machine learning classifiers are trained and tested, leveraging the curated dataset and optimized features.

The study employs evaluative metrics such as accuracy, precision, recall, and the f1 score to assess the methodology's effectiveness and guide further refinement.

3.2 Design

3.2.1 Data Refinement

The initial raw data underwent a rigorous cleaning and preprocessing phase, removing corrupted, duplicated, broken, or incorrectly formatted URLs. URLs with identical domain names were limited to a maximum of 14 instances within the dataset to avoid over-representation.

3.2.2 Extraction of URL-Based Features

a. Feature Identification

The next phase centered on the extraction of salient features from the URLs. Features were identified based on five pivotal attributes of URLs: length, count, rate of change, insights derived from natural language processing, and overall feature significance. A deeper dive into these feature parameters will be provided in the subsequent sections.

b. Optimizing the Feature Set

After extraction, the features were optimized using advanced selection algorithms such as Logistic Regression, Random Forest, Gradient Boosting, XGBoost, and LightGBM, facilitated by the Python-based machine learning library, sci-kit learn. The goal was to retain only the most predictive features.

3.2.3 Data Evaluation Using Machine Learning

a. Model Training and Testing

With the optimized feature set, the dataset was subjected to four supervised machine learning algorithms for training and testing, developing scalable and robust classification models for real-time URL phishing detection.

b. Classification

The trained models (Logistic Regression, Random Forest, Gradient Boosting, XGBoost, and LightGBM) were then used to classify each URL as Benign, Defacement, Phishing, Malware, or Spam, based on its inherent features.

3.3 Research Methodology

3.3.1 Data Collection

In the present research, we utilized a comprehensive dataset encompassing a myriad of URLs. These URLs are categorized into five distinct classifications: benign, defacement, phishing, spam, and malware. The compilation of this dataset was achieved by aggregating data from several open-source repositories, including but not limited to OpenPhish, Phishtank, Zone-H, and the Kaggle platform.

Figure-2 depicts a bar chart showing the frequency of different types of cyber security incidents classified into five categories: benign, defacement, phishing, spam, and malware. The tallest bar represents 'benign' incidents, indicating a majority of the data points did not involve malicious activity. 'Defacement' incidents, which involve unauthorized alterations to website appearances, constitute the second most frequent category. 'Phishing', the act of attempting to acquire sensitive information fraudulently, and 'spam', unsolicited bulk messages, are less common but present. 'Malware', software intended to damage or disrupt systems, has the fewest occurrences. This distribution is essential for understanding the landscape of cyber security threats and guiding protective measures.

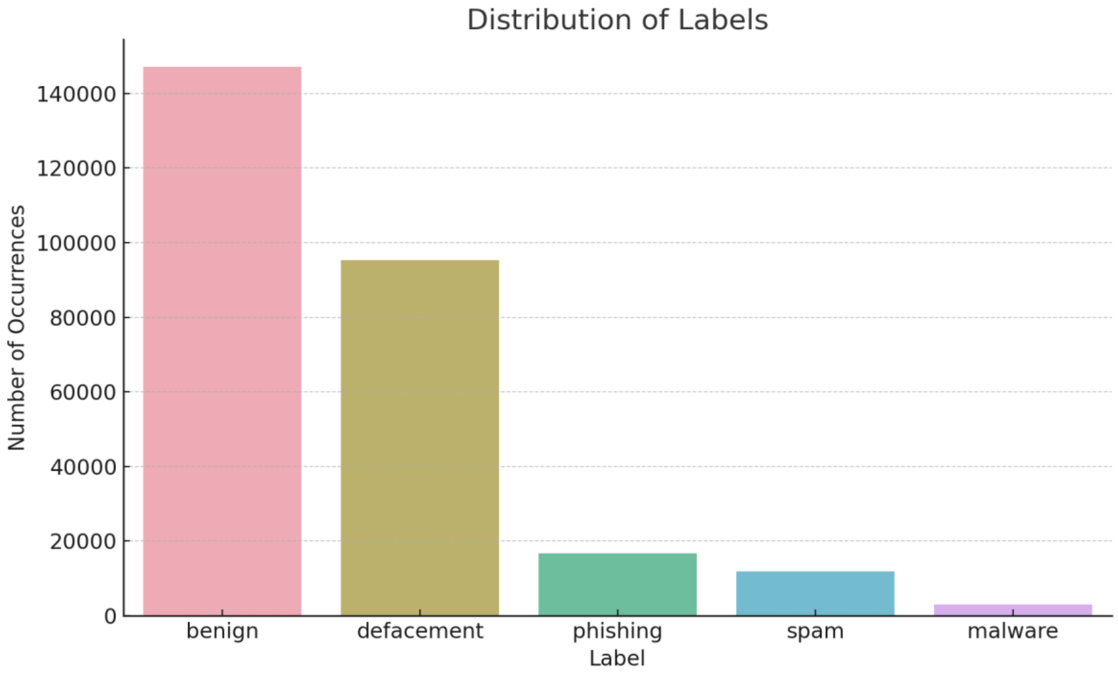


Figure 2: Categorization of Labels in Dataset

In the context of this research, the dataset under consideration was meticulously divided into two distinct segments: a training dataset and a testing dataset. This bifurcation was executed to ensure the robustness and validity of the machine learning models being employed.

a. **Training dataset:** This segment encompasses a significant 80% of the entire dataset. Its primary purpose is to serve as the foundational data upon which the machine learning algorithms are trained. By exposing the models to a vast majority of the data, the aim to equip them with the necessary patterns and nuances that are inherent to the dataset.
b. **Testing dataset:** The remaining 20% of the dataset is allocated to this segment. Its primary role is to act as a benchmark for assessing the efficacy and accuracy of the machine learning models post-training. By evaluating the models on this dataset, this can gauge their performance on data that remained unseen during the training phase. This approach offers a more genuine reflection of the model's potential applicability and reliability in real-world scenarios.

A pivotal aspect of this research is the representation of each URL within the dataset. Every URL is delineated by a comprehensive set of features. These features, which are essentially attributes or characteristics, provide a multi-dimensional perspective of the URL. The intention behind curating such a feature set is to encapsulate a wealth of pertinent information. This, in turn, aids the machine learning algorithms in discerning the subtle differences between phishing URLs and their benign counterparts.

To ensure clarity and a deeper understanding, the subsequent section delves into the intricate process of feature extraction. This process is paramount as it elucidates how each URL is transformed into a quantifiable and analyzable entity, thereby facilitating the machine learning models in their classification tasks.

3.3.2 Feature Extraction

Feature extraction transforms raw URL data into a structured format suitable for machine learning algorithms. This process involved converting each URL into a multi-dimensional feature vector through various heuristic techniques.

The features include basic properties like URL length, character count, presence of specific keywords, etc. The goal of feature extraction was to capture attributes that distinguish phishing URLs from benign ones, enhancing the predictive accuracy of the machine learning models.

Let's analyze into a comprehensive elucidation of each feature:

1. **'length':** Measures the total character count of the URL, encompassing every single character present.
2. **'slashes':** Quantifies the instances of slashes ('/') within the URL, offering insights into the URL's directory hierarchy.
3. **'dots':** Enumerates the dots ('.') present in the URL, shedding light on the domain's depth and its associated subdomains.
4. **'http' and 'https':** These features tally the instances of the strings "http" and "https" within the URL, signifying the protocols the URL employs.
5. **'suspicious_keywords'**: Enumerates specific keywords, such as 'login', 'signin', 'bank', and others, within the URL. A higher count might hint at the URL's dubious nature.
6. **'subdomain'**: This binary feature discerns the presence of a subdomain within the URL. A value of 1 indicates that the URL's hostname contains multiple dots ('.').

7. **'ip':** Another binary feature, it ascertains if the URL's hostname is an IP address, assigning a value of 1 if the hostname solely comprises digits and dots.
8. **'domain_extension'**: Denotes the character count of the parsed URL's path component.
9. **'num_digits':** Enumerates the digits present within the URL.
10. **'num_special_chars':** Tally of specific special characters, such as ';', '&', '%', '?', '=', and '-', found in the URL.
11. **'num_subdirectories':** Quantifies the slashes ('/') in the URL, indicative of the number of subdirectories.
12. **'tld':** A binary feature that checks if the URL's top-level domain (TLD) belongs to a predefined set, including 'xyz', 'online', 'club', and 'top'.
13. **'www':** Determines the presence of 'www' within the URL's hostname.
14. **'num_query_params':** Enumerates the query parameters present in the URL.
15. **'netloc_length':** Measures the character count of the netloc (network location) component in the parsed URL.
16. **'num_netloc_segments':** Enumerates the segments present within the netloc.
17. **'avg_path_segment_length':** Denotes the mean character count of segments within the path.
18. **'avg_query_segment_length':** Represents the mean character count of segments within the query.
19. **'num_colon', 'num_percent', 'num_plus', 'num_comma':** These features enumerate the instances of specific characters within the URL.
20. **'path_length':** Measures the character count of the parsed URL's path component.
21. **'query_length':** Denotes the character count of the parsed URL's query component.
22. **'num_path_segments', 'num_query_segments':** These features mirror the values represented by ['num_subdirectories'] and ['num_query_params'] respectively.
23. **'num_equals', 'num_at':** Enumerate the occurrences of '=' and '@' characters within the URL.
24. **'has_port', 'has_query', 'has_fragment', 'has_password', 'has_username', 'has_params':** These binary features signify the presence of specific components within the parsed URL, such as port, query, fragment, password, username, and params.

Indeed, a comprehensive set of 24 distinct features was meticulously extracted from every individual URL. This extensive feature set was curated with the primary objective of encapsulating a holistic representation of each URL. By doing so, this aimed to furnish the machine learning algorithms with a rich and detailed dataset. This, in turn, is anticipated to bolster the models' predictive prowess, ensuring they can discern and classify URLs with heightened accuracy and precision.

3.3.3 Data Preprocessing

In machine learning, feature extraction is just the beginning of a complex process. After extracting features, it's crucial to thoroughly prepare the data for the optimal performance of machine learning algorithms. The first step in this detailed process involves label encoding. This technique is essential as it converts categorical labels into numerical values, a format necessary for machine learning algorithms to process and interpret the data effectively.

Following this conversion, the dataset is carefully divided into two parts: a training set and a test set. The training set forms the basis for model training, while the test set is used to evaluate the effectiveness of the models. Notably, the test set makes up only 20% of the total dataset, leaving a substantial amount of data for training.

A significant hurdle in this phase is the issue of class imbalance, which can adversely affect the accuracy of machine learning models. To address this, the Synthetic Minority Over-sampling Technique (SMOTE) is employed. SMOTE is known for its ability to balance class representation by creating synthetic instances of the less represented class, thus ensuring a more balanced distribution of classes in the training dataset.

As the process advances, it becomes apparent that the extracted features vary in scale and magnitude. To achieve uniformity and prevent any undue influence on models (particularly those sensitive to feature size), the features are standardized using the StandardScaler. This technique adjusts the features to have a mean of zero and a standard deviation of one, ensuring consistent input data.

The adoption of these preprocessing methods is a result of careful consideration, with the primary goal being to refine the data structure to suit the intricate needs of machine learning algorithms.

3.3.4 Model Selection

In this research effort, [22] [57] [58] a selection of five distinct machine learning models has been made with careful consideration, each bringing its own strengths to the task of phishing detection. The choice of these particular models is based on their well-established reputation and demonstrated success across various classification challenges. These models have been recognized for their ability to effectively handle tasks like phishing detection, making them suitable candidates for this study [59]–[61].

a. **Logistic Regression**

   This model, commonly employed for binary classification challenges, is anchored in the logistic function. It endeavors to estimate the probability that a particular instance belongs to a designated class. Despite its simplicity, logistic regression has consistently showcased commendable performance, especially with linearly separable classes. A notable advantage is its diminished propensity for overfitting.

b. **Random Forest**

   This model adopts an ensemble learning strategy, which involves the generation of multiple decision trees during its training phase. The final output is derived either by choosing the class that emerges most frequently across the trees or by averaging the predictions from the individual trees. This model's inherent flexibility allows it to adeptly capture complex interrelationships between variables.

c. **Gradient Boosting**

   This iterative ensemble technique seeks to incrementally improve the accuracy of the response variable's estimation. It operates by training successive models on the residuals of preceding models and then amalgamating the predictions. The essence of gradient boosting revolves around refining predictions through a series of iterations, each building upon the errors of the last.

d. **XGBoost**

   Recognized for its dominance in machine learning competitions, this model employs gradient boosting machines, renowned for their scalability and precision. The methodology presented here leverages a parallel tree boosting technique, adept at addressing a myriad of data science challenges with remarkable efficiency.

e. **LightGBM**

This model integrates tree-based learning algorithms within the gradient boosting framework. Distinctively designed, it boasts attributes of distribution and efficiency, translating to advantages in both training duration and model accuracy.

The rationale for the selection of these specific models is anchored in their widespread recognition and their exemplary performance across a spectrum of classification tasks.

Model	Speed	Efficiency	Learning Ability	Handles Imbalance	Interpretability
Logistic Regression	High	High	Medium	No	High
Random Forest	Medium	Medium	High	Yes	Medium
Gradient Boosting	Low	Medium	High	Yes	Low
XGBoost	Medium	High	High	Yes	Low
LightGBM	High	High	High	Yes	Low

Table 2: Comparison of Machine Learning Algorithms against rating parameters

Explanation of Rating Parameters:

Here is the explanation of each rating parameter:

a. **Speed**

   This metric evaluates the temporal efficiency of the model, encompassing both its training and prediction phases. A model's speed is paramount, particularly when confronted with voluminous datasets, as it directly influences the time-to-insight and the overall feasibility of deploying the model in real-time or near-real-time scenarios.

b. **Efficiency**

   This criterion assesses the model's adeptness in harnessing computational resources. A model that is efficient optimizes the use of memory, processing power, and other system resources, ensuring that the maximum output is derived from the minimum input. Such efficiency becomes crucial in scenarios where computational resources are constrained or when optimizing for cost-effectiveness.

c. **Learning Ability**

This metric explores the model's capacity to discern and represent complex relationships within the data. It gauges the model's proficiency in capturing both linear and intricate non-linear associations, ensuring that the underlying patterns and nuances of the dataset are accurately mirrored in the model's predictions.

d. **Handles Imbalance**

Datasets in the real world often exhibit class imbalances, where certain classes are underrepresented. This metric evaluates whether the model possesses inherent mechanisms or strategies to address such imbalances, ensuring that the predictions are not unduly biased towards the majority class.

e. **Interpretability**

This criterion gauges the transparency and comprehensibility of the model's decision-making process. An interpretable model allows stakeholders to understand the rationale behind its predictions, fostering trust and facilitating its adoption, especially in sectors where explicability is mandated, like healthcare or finance.

3.3.5 Hyperparameter Tuning

Hyperparameters are specific settings of a machine learning model that are not directly learned from the data during training. These parameters are set before training begins and have a significant impact on the model's performance. Therefore, optimizing hyperparameters is a critical step in developing a machine learning model.

In this study, the hyperparameter tuning phase for the models was conducted using RandomizedSearchCV. This method differs from traditional tuning approaches like GridSearchCV by conducting a randomized search within pre-defined hyperparameter ranges. While GridSearchCV exhaustively searches all possible combinations in a parameter grid, RandomizedSearchCV offers a balance between computational efficiency and model quality, ensuring effective use of resources without sacrificing performance.

It is important to note that each model in this study had its unique set of hyperparameters, chosen to align with its specific characteristics and requirements. The optimal hyperparameter configuration was determined by evaluating various combinations through cross-validation on the training data. This thorough process ensured that the chosen hyperparameters were not only suitable for the model's architecture but also fine-tuned for enhanced predictive accuracy.

Figure 3 gives an overview of whole process step by step from initial phase to the last phase, highlighting the phase of data collection to data cleaning/optimization to the training of machine learning algorithms and testing phase against dataset.

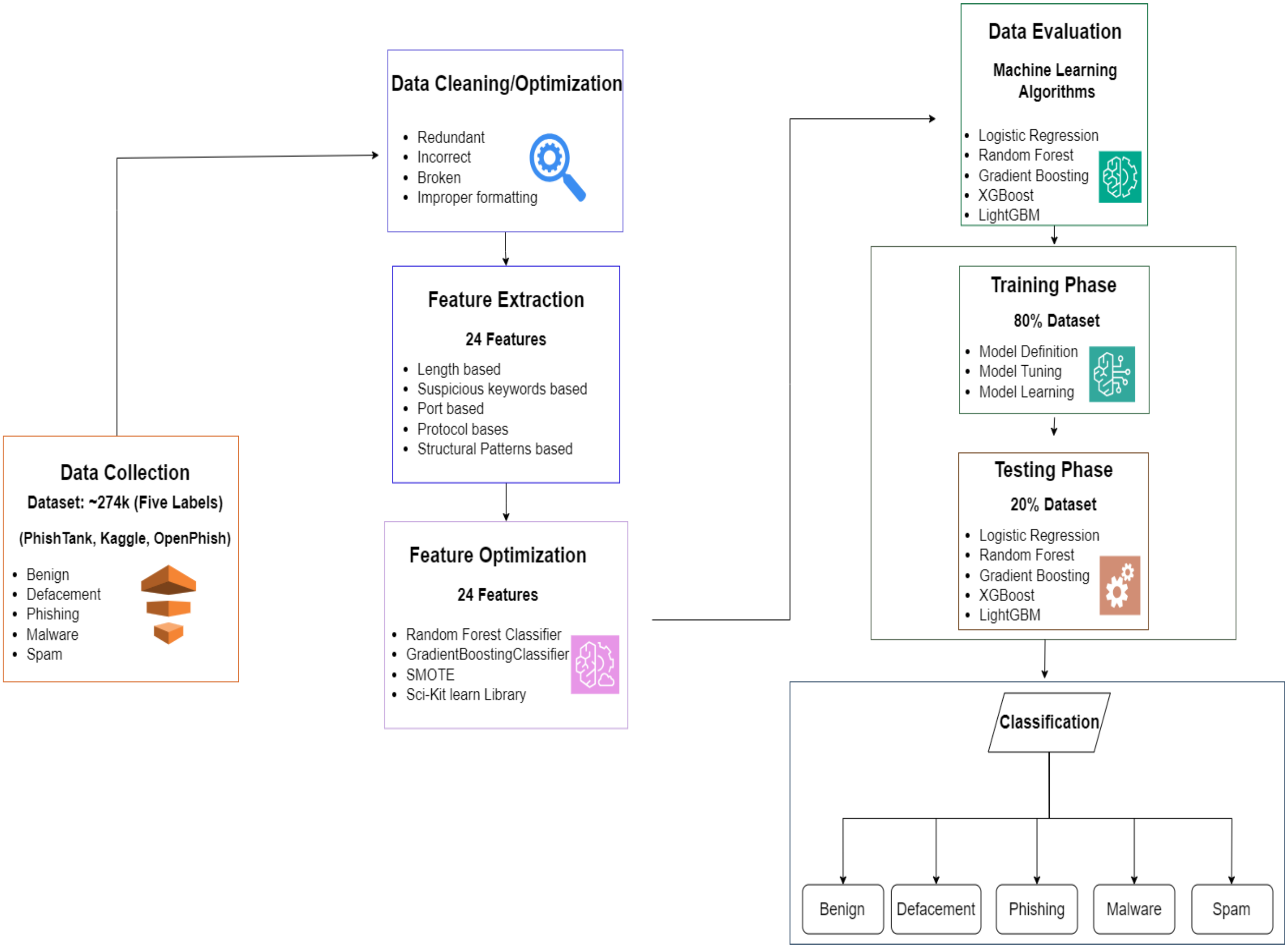


Figure 3: Research Methodology Flowchart

3.4 Summary

This chapter has thoroughly examined the methodology underpinning the research, carefully detailing each step of the experimental process. It has also highlighted the foundational design objectives, serving as guiding principles for this study, to offer context and clarity.

The architecture of the research is anchored by several key components, each playing a crucial role in the overall process. The first phase is Data Collection, where the focus was on systematically gathering dataset samples. This step was vital to ensure that the collected data was comprehensive and representative, laying a solid foundation for the analyses that would follow.

Following the collection of data, the research moved into the Feature Selection phase. This stage was particularly critical as it involved identifying and isolating the most relevant features from the data. The selection of these features is a pivotal aspect because it determines the variables that are instrumental in driving the predictive models.

Once the key features were identified, the next step was Feature Optimization. This phase was centered on refining the selected features to enhance their effectiveness. The objective was to improve the efficacy of the results, ensuring that the features not only encapsulate the core aspects of the data but also augment the performance of the model.

The final step in the research's architecture was the Implementation of Machine Learning Classifiers. At this stage, with the data preprocessed and ready, machine learning classifiers were deployed. These algorithms, specifically tailored to the unique characteristics of the dataset, were responsible for identifying patterns and making predictions. This phase marked the culmination of the process, where the data, having been collected, selected, and optimized, was now being analyzed and interpreted through advanced machine learning techniques.

In next chapter, the discussion will be on the detailed exposition of the experiments conducted. Additionally, the metrics employed to gauge and interpret the results will be comprehensively discussed, providing insights into the evaluative framework that underlies the research.

CHAPTER 4

EXPERIMENTS AND EVALUATION

Chapter 4 delves into the details of the experimental framework, covering the setup, conducted experiments, obtained results, and the phases of processing schemes. Central to this chapter is the assessment of the proposed methodology's effectiveness. It involves setting up a sophisticated development environment, utilizing various machine learning algorithms, and analyzing evaluation metrics. In response to the growing sophistication of phishing attacks, the research has led to the development of an advanced machine-learning-based solution. This chapter aims to provide a comprehensive understanding of the methodology, highlighting its effectiveness through detailed results from the machine learning algorithms. Additionally, the system's performance, evaluated using various metrics, is discussed to emphasize the robustness of the approach.

4.1 Overview

The digital world is experiencing an increase in phishing attacks, with perpetrators employing advanced social engineering techniques. This research presents an effective machine-learning solution to detect and prevent such threats. The following sections will present the results of integrating machine learning algorithms and discuss the evaluation metrics used to measure the system's performance.

4.2 Experimental Setup

4.2.1 Setting up the Environment

To enhance the outcomes from machine learning methods, an extensive dataset of both benign and phishing URLs was compiled. The experimental setup was conducive for running Python-based scripts, essential for training and evaluating machine learning models. For data extraction and empirical studies, an AWS EC-2 instance on a Windows 10 platform was used. The technical specifications of the computing device used in the study are outlined in a subsequent table.

Specification	Description
Manufacturer	AWS EC-2
Architecture	X64
Operating System	Windows 10
Processor	AMD EPYC 7R 13 Processor (192 CPUs)
RAM	1.5 TB

Table 3: Experimental Machine Specifications

4.2.2 Constructing the Dataset

The research utilized a dataset of 274,131 URLs, categorized into five categories: benign, defacement, phishing, spam, and malware. This dataset is a compilation from multiple open-source repositories, notably OpenPhish, Phishtank, and Kaggle. It integrates data from these sources to form a comprehensive set.

4.2.3 Python Source Code

Python's versatility supports various domains, including machine learning. The core code for the study was developed using Python and its machine learning libraries. Python was used for scripting, data visualization, and deploying machine learning algorithms, with libraries like Pandas, Numpy, TensorFlow, Scikit-Learn, and Keras. Python 3.10, obtained from its official portal, was installed following standard guidelines and integrated into the MacBook's VS Code IDE. The "pandas" library was used to import the CSV dataset, while "scikit-learn" was crucial for implementing and evaluating machine learning algorithms. An enumeration of the libraries used and their roles in the machine learning model's architecture is provided.

Some other software were required to be pre-installed before executing the Python code. The following list identifies all such softwares:

- WinRAR

- The latest version of Python Interpreter
- Visual Studio Code

4.3 Phases of Processing Scheme

The experimental design to evaluate the proposed approach was divided into two primary stages: Training and Testing. Each stage encompassed several detailed sub-processes. The dataset used was split in an 80:20 ratio, with 80% allocated for training and the remaining 20% for testing.

4.3.1 Training Phase

This phase began with extracting features from URLs, focusing on aspects such as length, rate, count, language, and significant features. The second stage involved optimizing these extracted features using feature selection algorithms from sci-kit-learn, specifically Logistic Regression, Random Forest, Gradient Boosting, XGBoost, and LightGBM. In the third stage, these features were further scrutinized and evaluated, ultimately forming the final dataset for the proposed model. The final stage involved applying various machine learning algorithms to this refined dataset to develop an anti-phishing URL detection model.

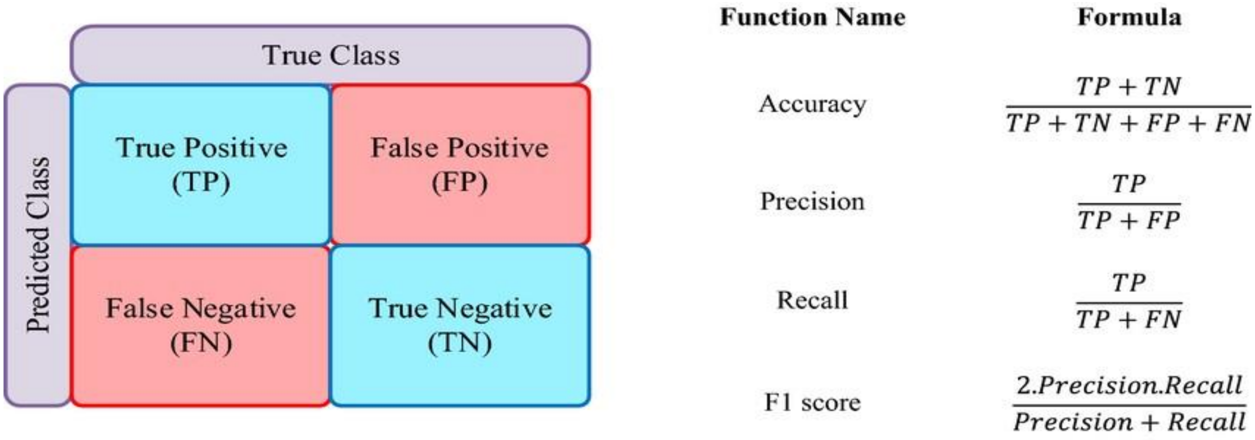
4.3.2 Testing Phase

During the testing phase, the trained classification model was used to categorize new URLs. The initial stage involved extracting features from these new URLs, similar to the training phase. These feature-extracted URLs were then introduced to the machine learning models to be categorized as benign, defacement, phishing, spam, or malware.

4.4 Evaluation Metrics

4.4.1 Confusion Metrics

In classification tasks within machine learning, a confusion matrix is a crucial tool for evaluating an algorithm's performance. It provides an in-depth look at the model's effectiveness by comparing the actual outcomes with the model's predictions. This matrix helps in understanding how well the model can correctly identify and classify the different categories of URLs.



Function Name	Formula
Accuracy	$\frac{TP + TN}{TP + TN + FP + FN}$
Precision	$\frac{TP}{TP + FP}$
Recall	$\frac{TP}{TP + FN}$
F1 score	$\frac{2.Precision.Recall}{Precision + Recall}$

Figure 4: Confusion Matrix and Performance Calculations Formula [62]

The matrix itself is a table layout that presents the actual vs. predicted classifications for a classification problem. The fundamental components of a confusion matrix are:

- **True Positive (TP) -** These are the cases in which the model predicted positive, and the truth was also positive.
- **True Negative (TN) -** Here, the model predicted negative, and the truth was indeed negative.
- **False Positive (FP) -** The model predicted positive, but the truth was negative. Often referred to as "Type I error."
- **False Negative (FN) -** The model predicted negative, but the truth was positive. This is sometimes called a "Type II error."

In the realm of classification algorithms, various metrics are employed to evaluate their performance. These metrics, derived from the number of instances, include Accuracy, Precision, Recall, and the F1 score. Their definitions are explained below:

- **Accuracy -** Accuracy serves as an indicator of the overall efficacy of a classification algorithm. It is determined by the proportion of instances that have been correctly classified, both as true and false, to the total instances encompassing True Positive, True Negative, False Positive, and False Negative.

- **Recall -** Often referred to as sensitivity or the true positive rate, Recall evaluates the proficiency of a classifier in identifying actual positive instances. It is deduced by the ratio of True Positive instances to the aggregate of True Positive and False Negative instances.
- **Precision -** Precision, in contrast to recall, emphasizes the classifier's aptitude in discerning positive instances from all predicted positives. It is the quotient of true positive instances to the combined count of true positives and false positives.
- **F1 Score -** The F1 score integrates the concepts of precision and recall, providing a comprehensive assessment of a classifier's performance. This score is calculated as the harmonic mean of precision and recall.

In summary, accuracy provides an overview of the classifier's overall correctness, while precision and recall focus on its ability to accurately identify positive instances and its sensitivity towards detecting these instances, respectively. The F1 score combines precision and recall into a single metric for a more holistic evaluation. Analyzing the confusion matrix and its associated metrics offers valuable insights into the model's performance, highlighting the strengths and weaknesses of the classification algorithms. This analysis is instrumental in identifying areas for improvement and assessing the model's appropriateness for particular tasks or domains. Through such an evaluation, it's possible to fine-tune the model for optimal performance in specific scenarios.

Table 4 summarizes the evaluation of five machine learning algorithms: Logistic Regression, Random Forest, LightGBM, Gradient Boosting, and XGBoost, using metrics such as Accuracy, Precision, Recall, and F1 Score. Ensemble methods—Random Forest, XGBoost, LightGBM, and Gradient Boosting—display superior performance across these metrics, whereas Logistic Regression shows comparatively lower values.

ML Algorithms	Accuracy	Precision	Recall	F1 Score
Logistic Regression	0.83	0.92	0.83	0.86
Random Forest	0.98	0.98	0.98	0.98
LightGBM	0.97	0.98	0.97	0.98
Gradient Boosting	0.97	0.98	0.97	0.97
XGBoost	0.98	0.98	0.98	0.98

Table 4: Evaluation Matrices result of ML Algorithms

Figure 4 presents a heatmap visualizing the performance metrics of various machine learning models. The models evaluated include Logistic Regression, Random Forest, LightGBM, Gradient Boosting, and XGBoost. The performance metrics are Accuracy, Precision, Recall, and F1 Score, with each cell in the heatmap displaying the percentage score of a model against a metric, ranging from 83.00 to 98.00. Darker shades represent higher scores as indicated by the color scale on the right. Ensemble methods such as Random Forest, LightGBM, Gradient Boosting, and XGBoost exhibit high performance across all metrics, consistently reaching scores of 97.00 and above. Logistic Regression has lower scores by comparison, with its highest metric being Precision at 92.00. This heatmap effectively communicates the comparative strengths of the models in a visually intuitive manner.

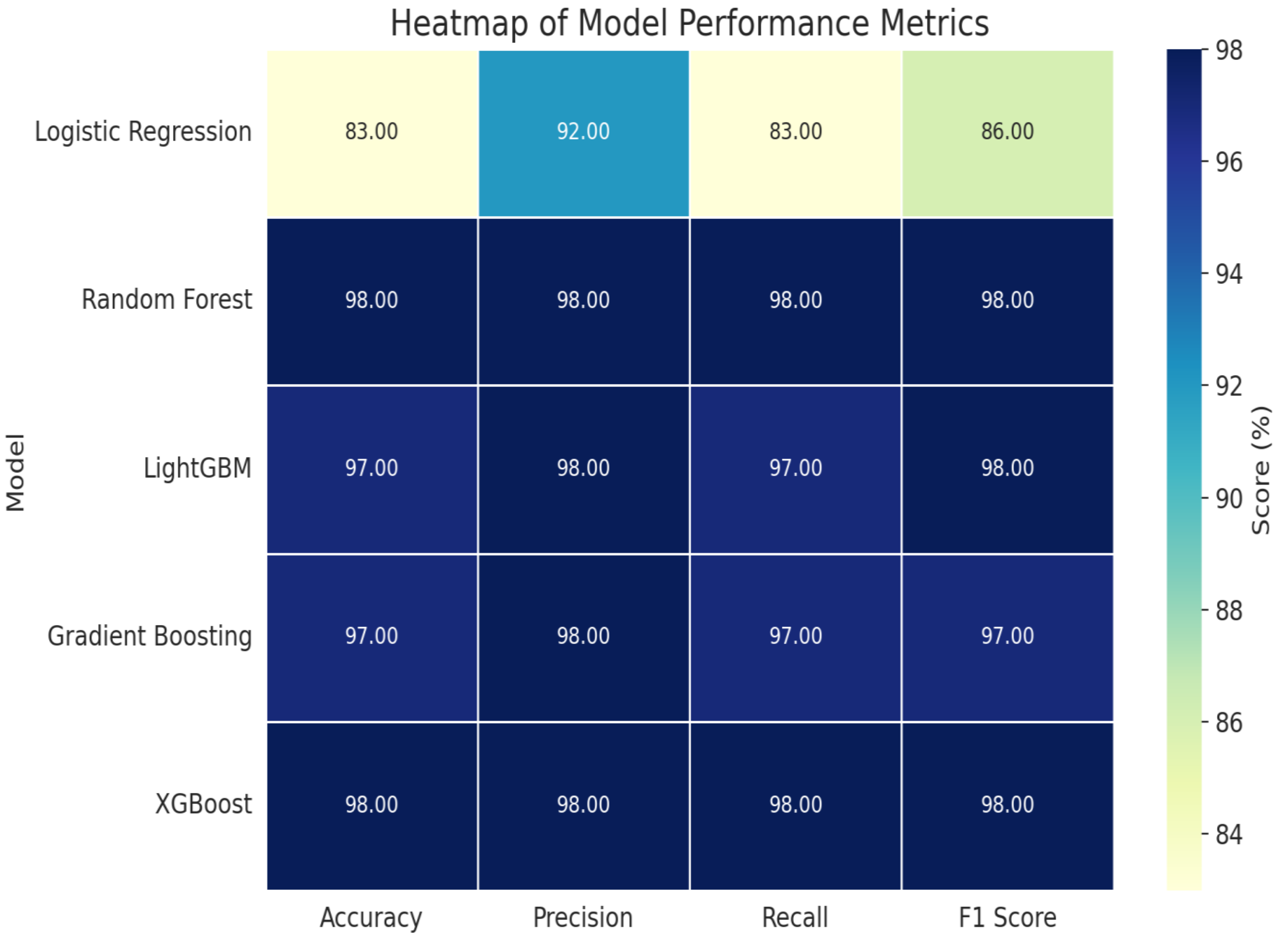


Figure 5: Heatmap of Model Performance Metrics

Figure 5 displays a bar chart comparing performance metrics for several machine learning models: Logistic Regression, Random Forest, LightGBM, Gradient Boosting, and XGBoost. The metrics assessed are Accuracy, Precision, Recall, and F1 Score, each represented by a distinct color and measured on a percentage scale from 0 to 100.

The chart shows that Logistic Regression has the lowest scores among the evaluated models, with its Accuracy and Recall at 83.00% and F1 Score at 86.00%. The other four models—Random Forest, LightGBM, Gradient Boosting, and XGBoost—achieve remarkably high scores across all metrics, with most scores at or above 97.00%. These results indicate the high predictive performance of ensemble methods compared to the single-predictor Logistic Regression model. The bar chart effectively communicates the differences in model performance, providing clear insights into which models are best suited for tasks requiring high Accuracy, Precision, Recall, and F1 Scores.

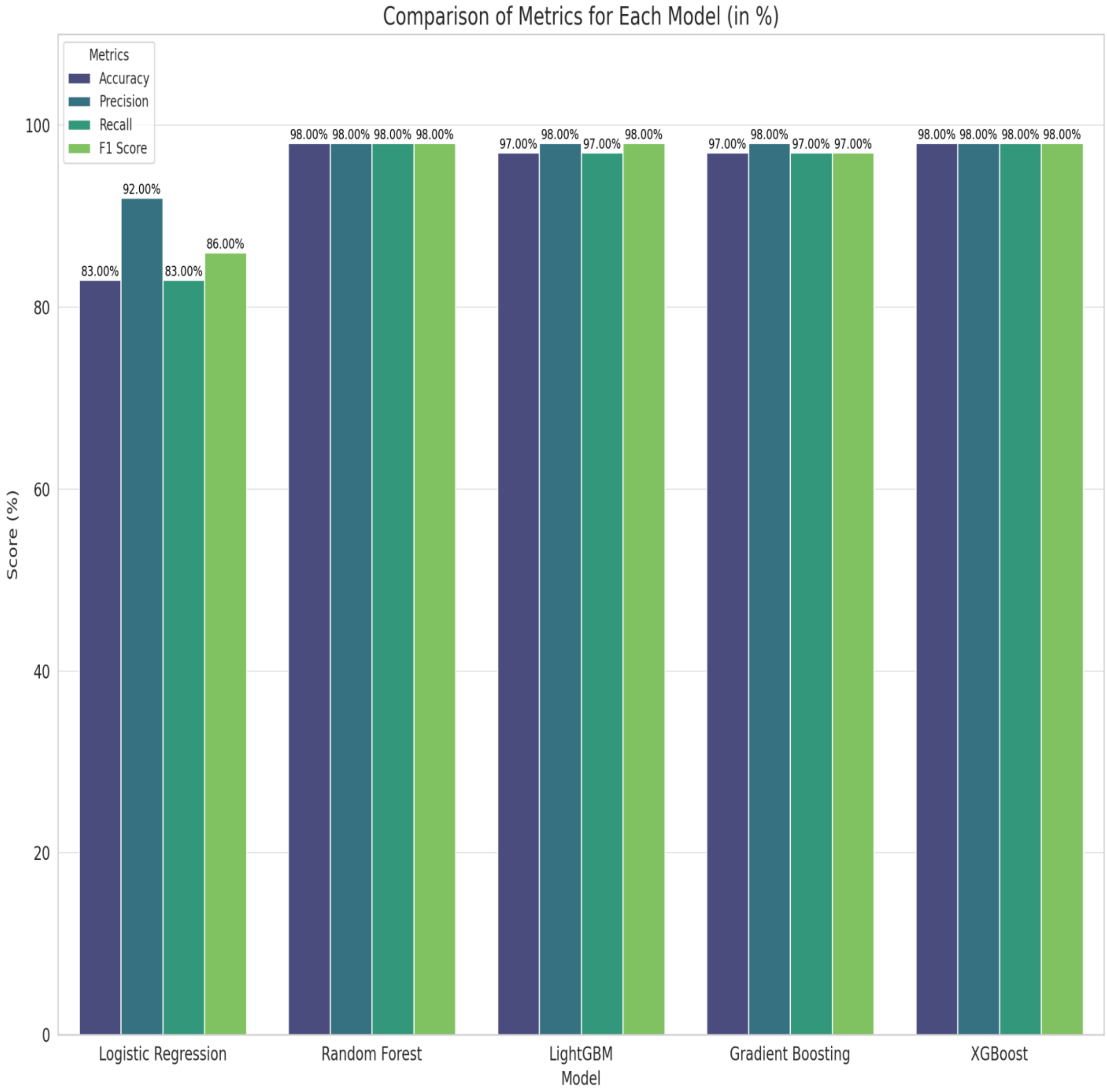


Figure 6: Comparison of Metrics for each Model

Following confusion matrices presented offer a quantitative evaluation of machine learning algorithms within a multi-class classification framework, identifying classes such as Benign, Defacement, Phishing, Spam, and Malware. Each matrix warrants a thorough examination to assess the effectiveness of the corresponding algorithm. The matrix's diagonal figures, reflecting correct predictions for each class, are indicative of the model's classification accuracy.

a. Logistic Regression Confusion Matrix Analysis

- The increased number of misclassifications, shown by the high false positive and false negative counts across classes, indicates a limitation in the model's discriminative capability.
- The notably lower classification success for Phishing and Spam may signal challenges in distinguishing these categories effectively.

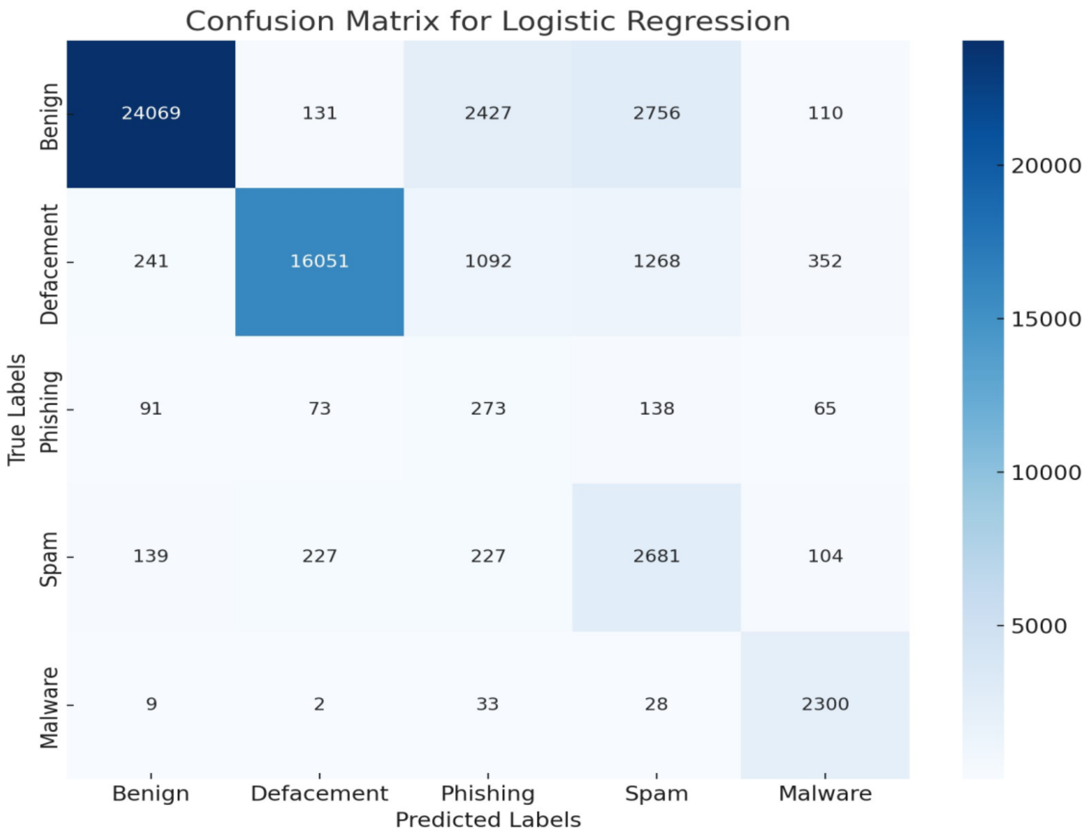


Figure 7: Confusion Matrix for Logistic Regression

b. Random Forest Confusion Matrix Analysis

- The high accuracy observed for Benign and Defacement classes suggests strong performance, yet an increased number of false negatives for Phishing and Spam is noted.

- The minimal misclassification between Benign and Malware classes suggests that the model is particularly effective in differentiating these categories.

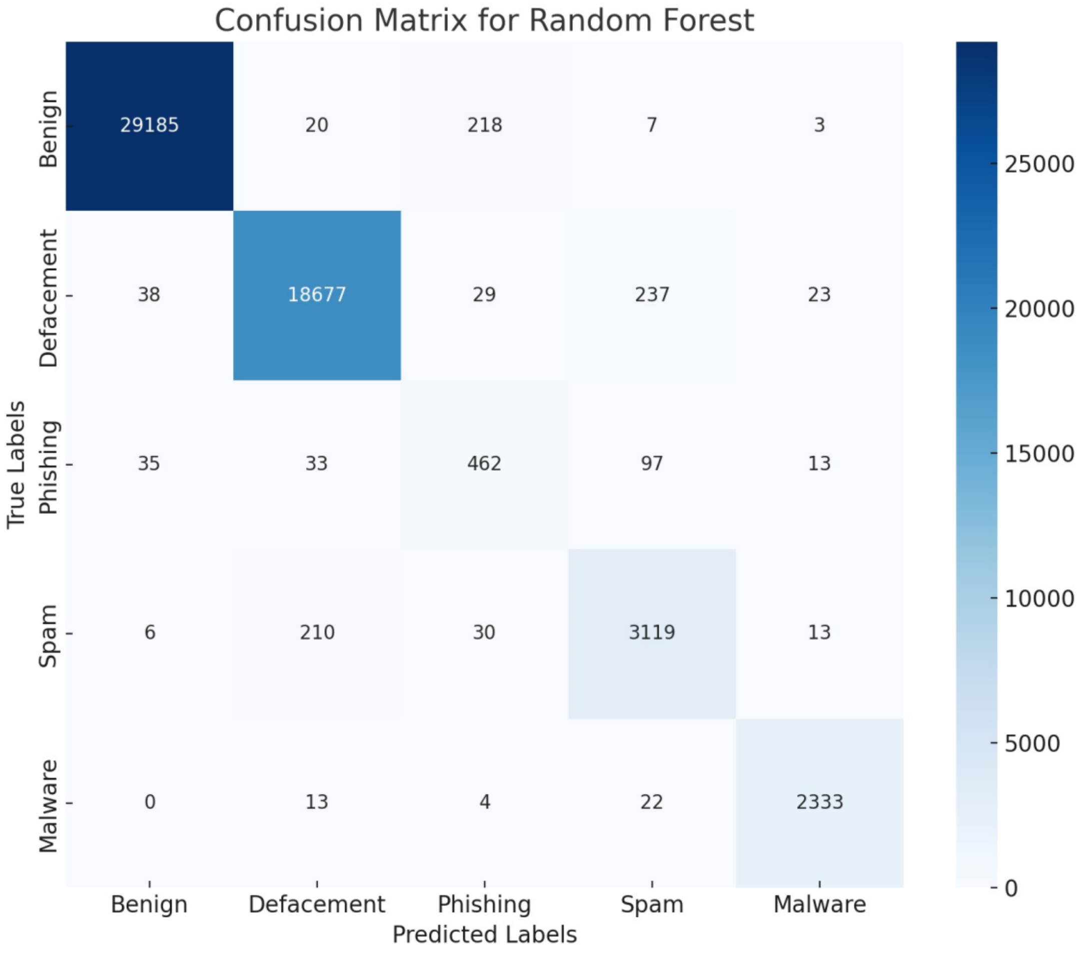


Figure 8: Confusion Matrix for Random Forest

c. LightGBM Confusion Matrix Analysis

- An improvement in the classification of Defacement and Spam categories is observed, suggesting a more refined performance by this model compared to Gradient Boosting.
- A reduction in false positive rates, such as the decrease in Benign instances classified as Phishing, demonstrates the model's improved precision.

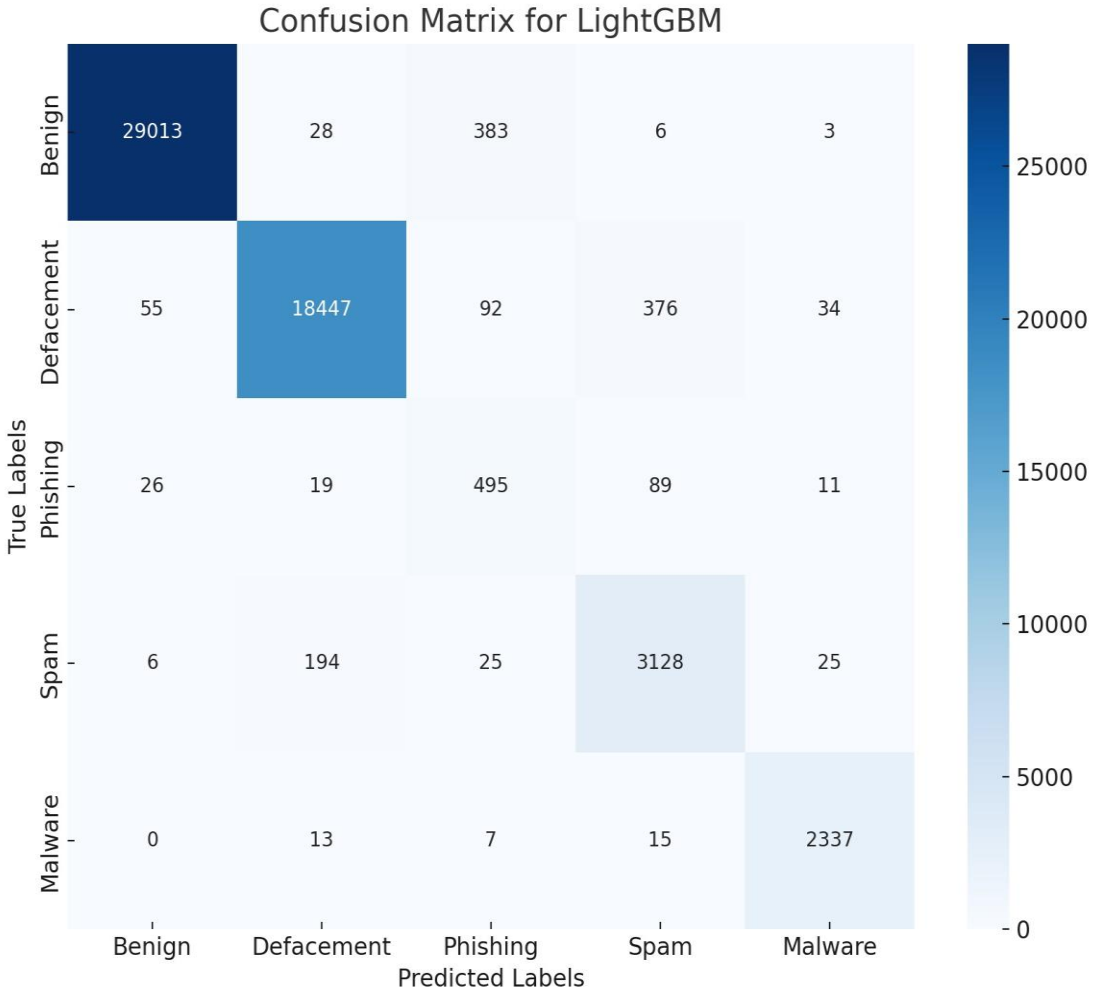


Figure 9: Confusion Matrix for LightGBM

d. Gradient Boosting Confusion Matrix Analysis

- The matrix's diagonal figures, reflecting correct predictions for each class, are indicative of the model's classification accuracy.
- The off-diagonal numbers, representing misclassifications, such as the 40 Benign instances classified as Defacement, highlight areas for potential improvement in the model's accuracy.

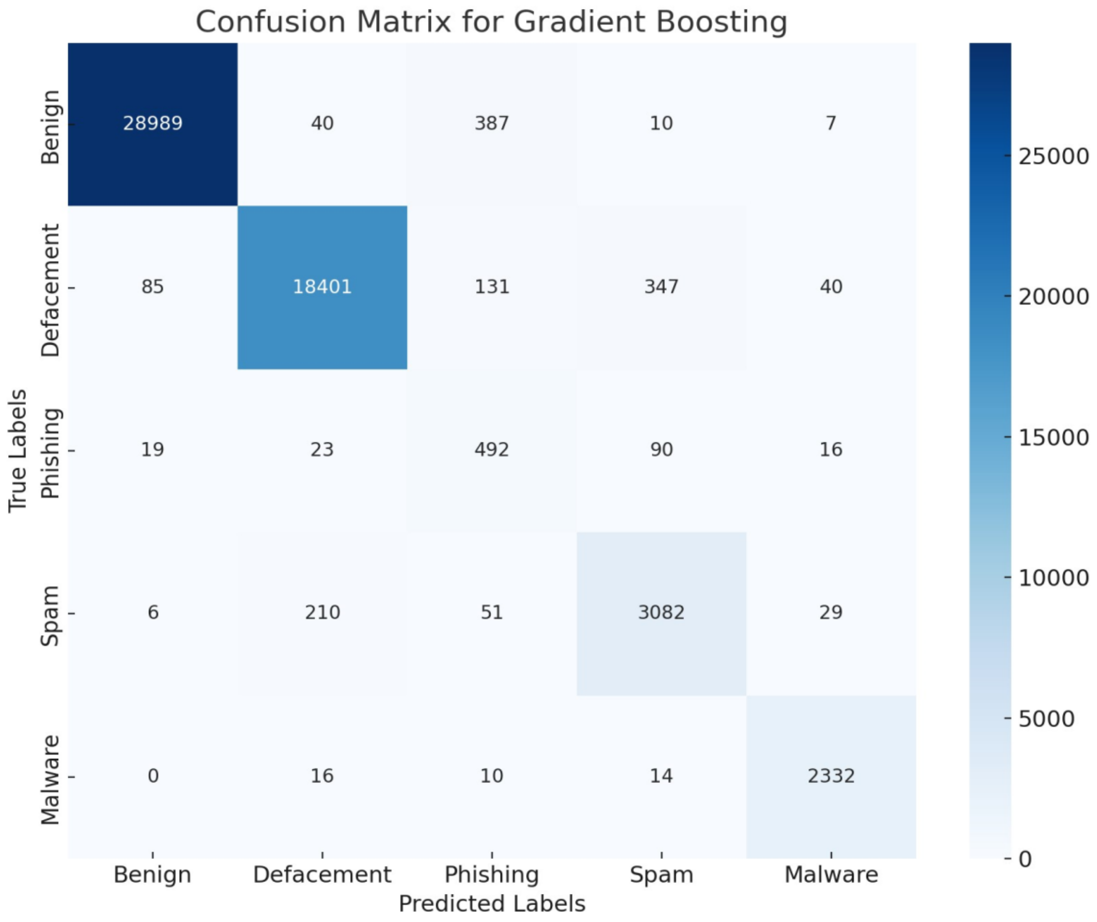


Figure 10: Confusion Matrix for Gradient Boosting

e. XGBoost Confusion Matrix Analysis

- Demonstrating a strong overall performance, this model achieves a notable balance in correctly identifying true positives and limiting misclassifications.
- Its relatively consistent performance across less frequent classes underscores the algorithm's robustness in managing class imbalance.

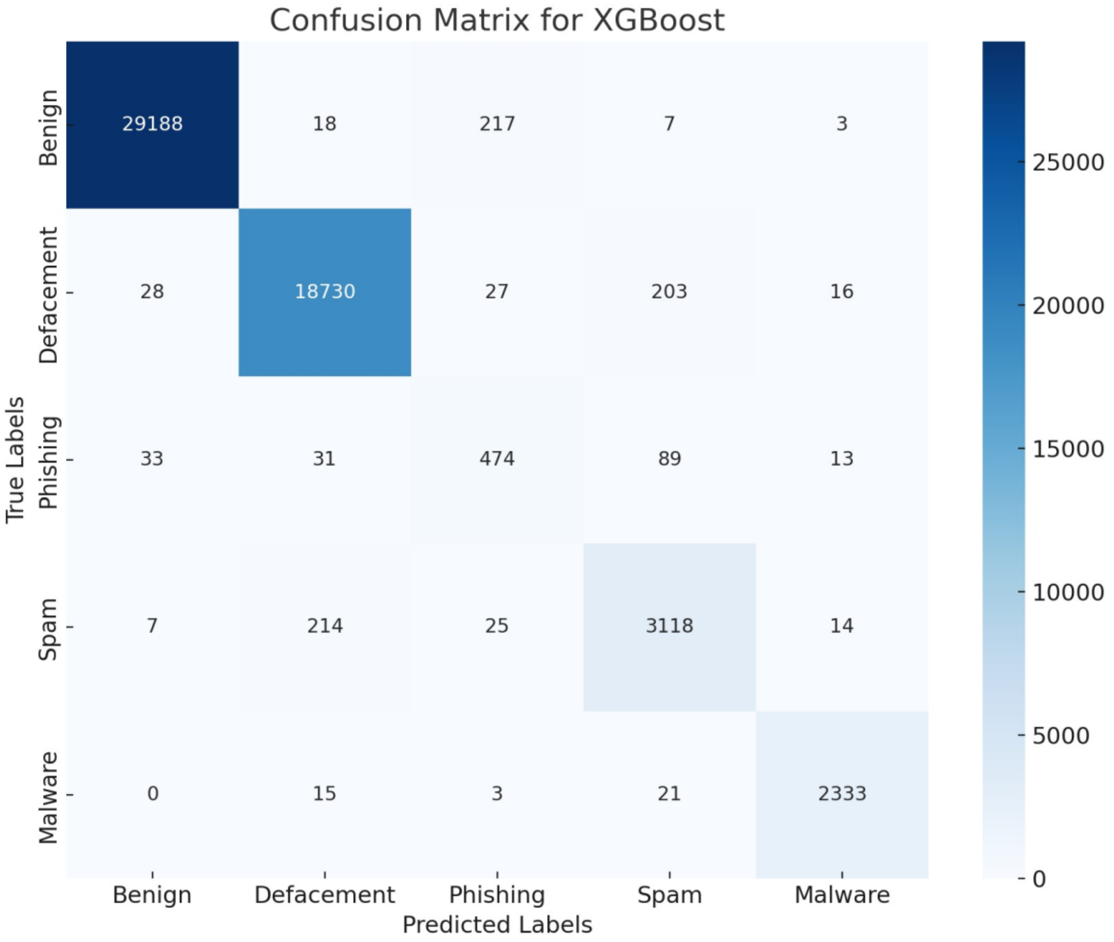


Figure 11: Confusion Matrix for XGBoost

4.5 Summary

In this chapter, the focus is on the testing conducted to evaluate the effectiveness of detecting phishing attacks using computational methods. The results of these tests were analyzed using a tool known as the confusion matrix. This analysis helped in assessing the performance of the newly developed approach. According to the findings, techniques such as Random Forest, Gradient Boosting, XGBoost, and LightGBM demonstrated notable effectiveness. The upcoming chapter will delve into comparing these results with other established methods, providing an understanding of how the new approach compares in the broader context of phishing detection.

CHAPTER 5

DISCUSSION AND ANALYSIS

The previous chapter detailed the experimental procedures and the metrics used to evaluate the proposed approach's effectiveness. In this chapter, the focus is on a detailed analysis and discussion of the experimental results. These findings will be compared to a benchmark methodology, assessing the superiority of the proposed solution over the standard reference model. Additionally, the chapter will highlight the significance and practical application of the proposed approach in URL phishing detection.

5.1 Overview

This chapter examines the effectiveness of the proposed research methodology by comparing the outcomes obtained from four distinct classification algorithms with a standard benchmark. The research employed a selective strategy, using only the most impactful features to identify patterns and gain insights from the data.

To evaluate the proficiency of the proposed methodology against the work of M.Ha et. al. [20], the benchmark technique was analyzed. This involved using four classification algorithms and incorporating nine foundational features from the benchmark study on an expanded dataset. Interestingly, testing this dataset of 274,131 entries revealed a noticeable drop in accuracy. As a result, the models were re-adjusted to align with the proposed methodology. This revised strategy focused on 24 key features, compared to the nine used in the benchmark study. Empirical data from this research suggest that the new methodology outperforms the benchmark, achieving similar accuracy levels even with the larger dataset. The following section provides an in-depth discussion of these findings and a careful comparison of both methodologies.

5.2 General Comparison with Recent Techniques

Research work	No of Features	Dataset Size	Algorithm	Accuracy %
Proposed Approach	24	274,131	LN, RF, XGBoost, Gradient Boosting, LightGBM	98
M.Ha et. al. [20] (2023)	20	213,345	DT, RF, KNN, AdaBoost	94.83
L. Bustio et. al. [21] (2022)	9	50,000	RF, SVM, DT, KNN	99.57
A. Subasi, et. al. [56] (2019)	32	20,000	SVM, KNN, ANN, RF, REPT, RT	97.61
A. Zamir et. al. [54] (2020)	32	11,055	NB, KNN, SVM, RF, Bagging, NN	97.31
A.A. Orunsolu et. al. [55] (2019)	15	5,041	SVM, NB	99.60

Table 5: Comparison of Recent Phishing Detection Techniques with its Proposed Feature Numbers, Data Size, and Achieved Accuracies

Table 5 provides a comparative analysis of current phishing detection methods, focusing on the number of features used, the size of the dataset, and the achieved precision. It reveals that many recent techniques achieve high accuracy but often with a limited number of features and a smaller dataset, or a combination of the two. In contrast, the proposed methodology in this research manages to achieve significant accuracy while utilizing an efficient number of features and handling a larger dataset.

To illustrate these differences, a graphical representation was created, represented in Figure 12. This graph visualizes the relationship between the number of features and the size of the datasets used in various phishing detection systems. The graph includes labels at the bottom indicating research from the past four years. The left side of the graph shows different dataset sizes, while the right side indicates the number of features used. A prominent horizontal blue line across the graph represents the effectiveness of each methodology in relation to its feature count and dataset size, offering a clear visual comparison of how the proposed methodology stands against others in the field.

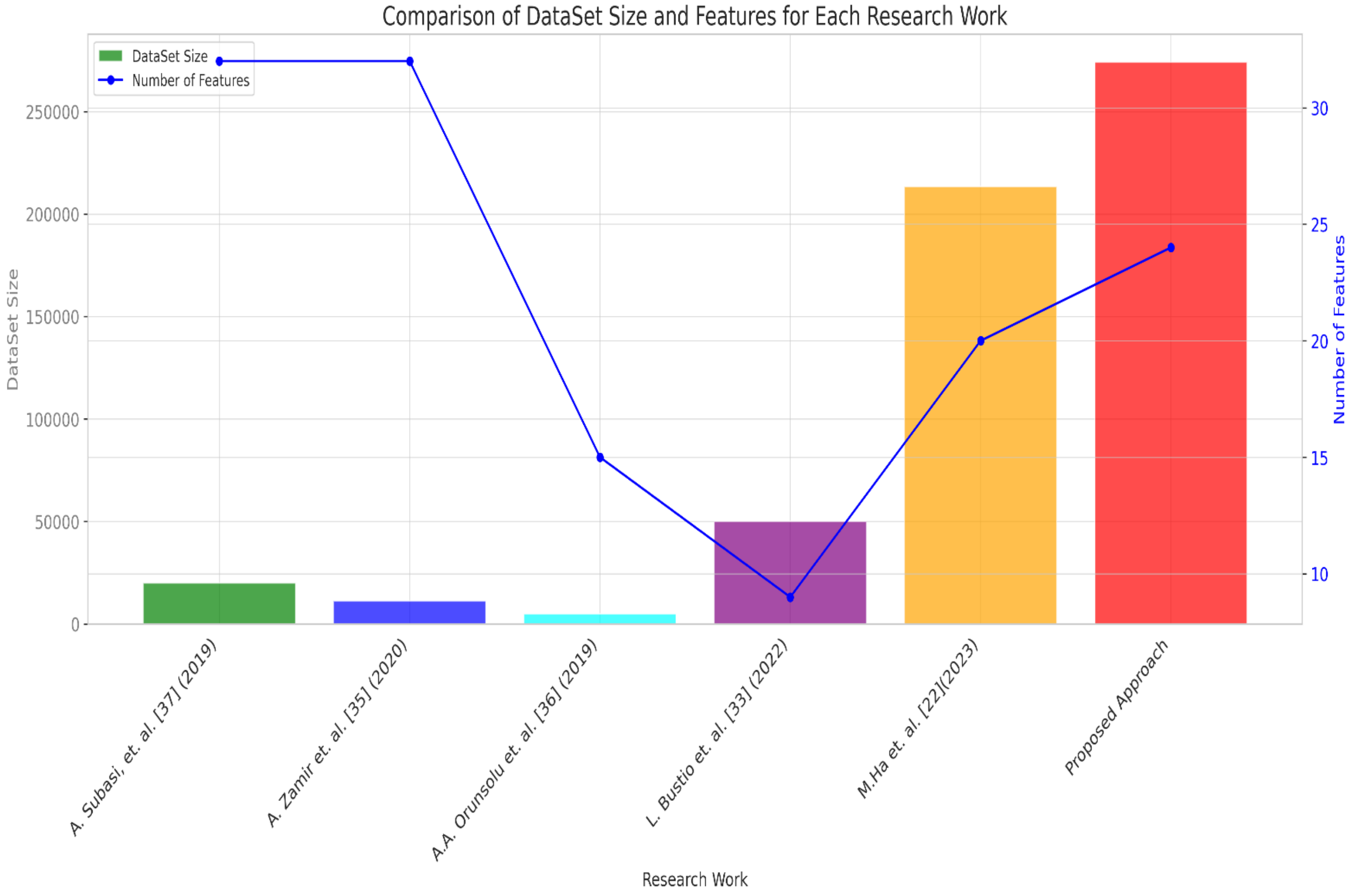


Figure 12: Comparison of Dataset Size and Featured for Each Research Work

Here's a concise interpretation of the graph:

a. The blue bars in the graph symbolize the dataset's volume, while the horizontal line delineates the feature count. References to the respective research papers, where these methodologies were employed, are anchored at the graph's base.

b. The graph clearly illustrates that with a selection of 24 features and proposed dataset comprising 274,131 entries, the performance indicator is positioned slightly beneath the average mark, especially when juxtaposed against the dataset dimensions and feature count employed in recent studies.

c. Furthermore, the graph underscores the utilization of the most expansive dataset coupled with an optimized feature set.

5.3 Discussion, Contribution, and Applications

The work presented here aligns with the methodology of M.Ha et. al. [20] focusing on a "Classification of Malicious Websites Using Machine Learning Based on URL Characteristics". This is achieved using 20 lexical features and four distinct Machine Learning algorithms. The reference method reported an accuracy of 94.83% on a dataset comprising 213,345 entries. This underscores the limitation of the benchmark approach in scaling effectively to deliver enhanced accuracy on expansive datasets. In response, proposed research introduced an optimized set of 24 features, elevating the detection accuracy to 98% on a dataset containing 274,131 samples. Notably, proposed dataset is nearly double the size of the reference dataset.

The overarching objective of this endeavor is to craft a lightweight, versatile, and scalable Machine Learning model akin to the reference but demanding fewer features, even when applied to larger datasets. The outcomes achieved validate the approach, with algorithms like Random Forest, Gradient Boosting, XGBoost, and LightGBM emerging as top performers, each registering an accuracy score of 98%. The experiments and evaluations were conducted on the AWS EC-2 Windows 10 platform.

To grasp the potential advantages and applications of the proposed solution, consider the following scenarios: This approach can expedite the detection of phishing onslaughts, adeptly categorize phishing sites, and thereby curtail or avert financial and reputational setbacks. The points shed light on the efficacy of the research, grounded in the results obtained:

a. Proposed solution harnesses URL-based lexical and other attributes to distinguish between phishing and legitimate websites.

b. By seamlessly melding with prevalent internet systems, this approach offers a robust shield against phishing incursions.

c. Major corporations and institutions can readily adopt this solution for real-time phishing detection.

An anti-phishing model centered on website URLs is instrumental in shielding users from the perils of phishing attacks. By adeptly identifying and neutralizing attempts to access malevolent URLs, it plays an indispensable role in diminishing associated hazards, enhancing online safety, and nurturing a secure digital realm. Through its meticulous analysis and discernment, this model stands as a bulwark, ensuring a more secure online experience for all.

5.4 Summary

This chapter has elucidated the results, engaged in a discourse, and undertaken an analytical review of the experiment executed. The research compares the outcomes derived from four distinct classification algorithms against a standard benchmark approach. This comparison aims to assess the efficacy and scalability of the proposed research methodology, especially in the context of phishing detection within a voluminous dataset. A graphical representation has been crafted to facilitate a visual understanding of these disparities. Additionally, the chapter sheds light on the notable contributions and potential applications stemming from this research. The subsequent chapter will pivot towards presenting the conclusion, delineating the limitations, and charting the course for future endeavors.

CHAPTER 6

CONCLUSION & FUTURE WORK

This chapter wraps up the research, highlighting its constraints and potential avenues for future exploration. A concise recap of the study is presented, emphasizing the main takeaways and pinpointing areas ripe for enhancement in subsequent research endeavors.

6.1 Conclusion

As internet usage and online services have grown, so too has the threat of cyber-attacks. With over 1.12 billion active websites, the risk of these sites compromising user security has notably increased. Machine learning frameworks offer a promising solution, capable of creating accurate models to identify phishing websites. The effectiveness of these models heavily depends on the size of the dataset used and the careful selection of relevant features.

This study discusses a flexible and scalable phishing detection system based on URL features. It utilizes an optimal set of features and employs machine learning classifiers, training and testing on an extensive dataset. Website URLs were gathered from various open-source repositories, including OpenPhish, Phishtank, and Kaggle, for phishing URLs. Relevant features were then extracted from these URLs. The aim of this thesis was to improve and extend the approach established by the benchmark, increasing the dataset size while maintaining accuracy levels comparable to the benchmark.

The results demonstrate that the proposed method, with a streamlined set of features, performs better than the benchmark approach on the larger dataset. Features related to URL length, character and symbol counts, rate-based attributes of URLs, and other key features were included, resulting in a refined set of 24 features. Through testing and the application of feature importance ranking algorithms from sci-kit-learn, this was narrowed down to 24 essential features suitable for a large-scale dataset.

The effectiveness of the proposed strategy was confirmed through tests using four different machine learning classifiers and various dataset sizes, as detailed in the "Experiments and Evaluation" chapter. Notably, the Random Forest classifier was identified as the most accurate, achieving an impressive accuracy rate of 98%.

6.2 Future Work and Limitations

This research lays the groundwork for phishing website detection, yet there's a vast expanse for refinement and broadening. As we explore novel methodologies, the domain of phishing detection can witness significant advancements, bolstering the potency of anti-phishing initiatives. Here are some forward-looking suggestions for refining anti-phishing models:

1. **Holistic Approach**

   Elevating the efficacy of anti-phishing models necessitates a holistic strategy. Incorporating user behavior analytics can shed light on patterns like users clicking dubious links or divulging sensitive information on unverified sites. Such insights can facilitate timely warnings and guide users away from potential phishing threats.

2. **Diverse Training Data**

   The richness of training data is pivotal for peak performance. It's imperative to encompass a spectrum of phishing modalities and attack avenues, from phishing emails and misleading sites to social engineering ploys and nascent tactics. Such a diverse dataset equips the model to adeptly spot and generalize novel and intricate phishing endeavors.

3. **Interactive Phishing Simulations**

   Crafting interactive and lifelike phishing simulation platforms is advisable. These platforms immerse users in hands-on scenarios, honing their skills to discern and counteract evolving phishing strategies, thereby amplifying their cyber vigilance and instilling robust cyber security practices.

4. **Ongoing Evaluation and Upgrades**

   The dynamic nature of the phishing realm demands consistent model evaluations and updates. Periodic performance checks, coupled with the infusion of fresh techniques and approaches, are vital to ensure the models stay abreast of emerging patterns and defensive measures.

5. **User Education and Awareness**

   A proactive stance against phishing threats hinges on enlightening users. Imparting knowledge about prevalent phishing strategies, empowering users to spot suspect emails or sites, and underscoring online security best practices are paramount for a proactive defense mechanism.

By embracing these comprehensive strategies, we can fortify anti-phishing models, rendering them adept at tackling the fluid and intricate landscape of phishing onslaughts.

REFERENCES

[1] J. Milletary, “Technical Trends in Phishing Attacks”, Accessed: Dec. 12, 2023. [Online]. Available: https://www.cisa.gov/sites/default/files/publications/phishing_trends0511.pdf

[2] “2021 Internet Crime Report - Homeland Security Digital Library.” Accessed: Dec. 12, 2023. [Online]. Available: https://www.hsdl.org/c/2021-internet-crime-report/

[3] J. Zhou, A. Holzinger, F. Chen, E. Jorge, A. Suárez, and V. Monzon Baeza, “Evaluating the Role of Machine Learning in Defense Applications and Industry,” *Machine Learning and Knowledge Extraction 2023, Vol. 5, Pages 1557-1569*, vol. 5, no. 4, pp. 1557–1569, Oct. 2023, doi: 10.3390/MAKE5040078.

[4] Z. Alkhalil, C. Hewage, L. Nawaf, and I. Khan, “Phishing Attacks: A Recent Comprehensive Study and a New Anatomy,” *Front Comput Sci*, vol. 3, p. 563060, Mar. 2021, doi: 10.3389/FCOMP.2021.563060/BIBTEX.

[5] A. Aleroud and L. Zhou, “Phishing environments, techniques, and countermeasures: A survey,” *Comput Secur*, vol. 68, pp. 160–196, Jul. 2017, doi: 10.1016/J.COSE.2017.04.006.

[6] M. Lee and E. Park, “Real-time Korean voice phishing detection based on machine learning approaches,” *J Ambient Intell Humaniz Comput*, vol. 14, no. 7, pp. 8173–8184, Jul. 2023, doi: 10.1007/S12652-021-03587-X/TABLES/7.

[7] B. Fitzpatrick, X. " Sherwin, " Liang, J. Straub, and C. Author, “Fake News and Phishing Detection Using a Machine Learning Trained Expert System,” Aug. 2021, Accessed: Dec. 11, 2023. [Online]. Available: https://arxiv.org/abs/2108.08264v1

[8] P. Saraswat and M. Singh Solanki, “Phishing Detection in E-mails using Machine Learning,” *Proceedings of International Conference on Technological Advancements in Computational Sciences, ICTACS 2022*, pp. 420–424, 2022, doi: 10.1109/ICTACS56270.2022.9987839.

[9] A. A. Adzhar, Z. Mabni, and Z. Ibrahim, “A Comparative Study on Email Phishing Detection Using Machine Learning Techniques,” *2022 IEEE International Conference on Computing, ICOCO 2022*, pp. 96–101, 2022, doi: 10.1109/ICOCO56118.2022.10031671.

[10] P. Habib, U. Sharma, and K. S. Sethi, “Phishing Detection with Machine Learning,” vol. 10, 2022, doi: 10.22214/ijraset.2022.48276.

[11] R. S. Rao, T. Vaishnavi, and A. R. Pais, “CatchPhish: detection of phishing websites by inspecting URLs,” *Journal of Ambient Intelligence and Humanized Computing 2019 11:2*, vol. 11, no. 2, pp. 813–825, May 2019, doi: 10.1007/S12652-019-01311-4.

[12] W. Wei, Q. Ke, J. Nowak, M. Korytkowski, R. Scherer, and M. Woźniak, “Accurate and fast URL phishing detector: A convolutional neural network approach,” *Computer Networks*, vol. 178, p. 107275, Sep. 2020, doi: 10.1016/J.COMNET.2020.107275.

[13] A. Trozze *et al.*, “Cryptocurrencies and future financial crime,” *Crime Sci*, vol. 11, no. 1, pp. 1–35, Dec. 2022, doi: 10.1186/S40163-021-00163-8/TABLES/8.

[14] H. F. Atlam and O. Oluwatimilehin, “Business Email Compromise Phishing Detection Based on Machine Learning: A Systematic Literature Review,” *Electronics 2023, Vol. 12, Page 42*, vol. 12, no. 1, p. 42, Dec. 2022, doi: 10.3390/ELECTRONICS12010042.

[15] R. Abdulraheem, A. Odeh, M. Al Fayoumi, and I. Keshta, “Efficient Email phishing detection using Machine learning,” *2022 IEEE 12th Annual Computing and Communication Workshop and Conference, CCWC 2022*, pp. 354–358, 2022, doi: 10.1109/CCWC54503.2022.9720818.

[16] J. Rashid, T. Mahmood, M. W. Nisar, and T. Nazir, “Phishing Detection Using Machine Learning Technique,” *Proceedings - 2020 1st International Conference of Smart Systems and Emerging Technologies, SMART-TECH 2020*, pp. 43–46, Nov. 2020, doi: 10.1109/SMART-TECH49988.2020.00026.

[17] “Machine Learning Applications in the Cybersecurity Space.” Accessed: Dec. 12, 2023. [Online]. Available: https://securityintelligence.com/posts/machine-learning-applications-in-the-cybersecurity-space/

[18] Sasirekha C, Nandhini R, Karthiga Mai N L, Bhuvaneshwari R S, and Chandra V S, “Email Phishing Detection Using Machine Learning,” *International Journal of Engineering Research & Technology*, vol. 11, no. 3, Jun. 2023, doi: 10.17577/IJERTCONV11IS03045.

[19] G. Harinahalli Lokesh and G. BoreGowda, “Phishing website detection based on effective machine learning approach,” *Journal of Cyber Security Technology*, vol. 5, no. 1, pp. 1–14, Jan. 2021, doi: 10.1080/23742917.2020.1813396.

[20] Muon Ha, Yulia Shichkina, Nhan Nguyen, and Thanh-Son Phan, “Classification of Malicious Websites Using Machine Learning Based on URL Characteristics,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 14112 LNCS, pp. 317–327, 2023, doi: 10.1007/978-3-031-37129-5_26/COVER.

[21] L. Bustio-Martínez, M. A. Álvarez-Carmona, V. Herrera-Semenets, C. Feregrino-Uribe, and R. Cumplido, “A lightweight data representation for phishing URLs detection in IoT environments,” *Inf Sci (N Y)*, vol. 603, pp. 42–59, Jul. 2022, doi: 10.1016/J.INS.2022.04.059.

[22] N. N. Gana and S. M. Abdulhamid, “Machine Learning Classification Algorithms for Phishing Detection: A Comparative Appraisal and Analysis,” *2019 2nd International Conference of the IEEE Nigeria Computer Chapter, NigeriaComputConf 2019*, Oct. 2019, doi: 10.1109/NIGERIACOMPUTCONF45974.2019.8949632.

[23] A. Sundararajan, G. Gressel, and K. Achuthan, “Feature selection for phishing detection with machine learning,” *Int J Eng Adv Technol*, vol. 8, no. 6 Special Issue 3, pp. 1039–1045, Sep. 2019, doi: 10.35940/IJEAT.F1331.0986S319.

[24] M. Almseidin, A. M. Abu Zuraiq, M. Al-kasassbeh, and N. Alnidami, “Phishing Detection Based on Machine Learning and Feature Selection Methods,” *International Journal of Interactive Mobile Technologies (iJIM)*, vol. 13, no. 12, pp. 171–183, Dec. 2019, doi: 10.3991/IJIM.V13I12.11411.

[25] O. K. Sahingoz, E. Buber, O. Demir, and B. Diri, “Machine learning based phishing detection from URLs,” *Expert Syst Appl*, vol. 117, pp. 345–357, Mar. 2019, doi: 10.1016/J.ESWA.2018.09.029.

[26] “The Anatomy of a URL – Stella Sofia’s Blog.” Accessed: Dec. 12, 2023. [Online]. Available: https://stellasis.home.blog/2021/09/17/the-anatomy-of-a-url/

[27] M. Abutaha, M. Ababneh, K. Mahmoud, and S. A. H. Baddar, “URL Phishing Detection using Machine Learning Techniques based on URLs Lexical Analysis,” *2021 12th International Conference on Information and Communication Systems, ICICS 2021*, pp. 147–152, May 2021, doi: 10.1109/ICICS52457.2021.9464539.

[28] N. Kumaran, P. Sri Sai, and L. Manikanta, “Web Phishing Detection using Machine Learning,” *International Journal of Innovative Technology and Exploring Engineering*, vol. 11, no. 4, pp. 56–59, Mar. 2022, doi: 10.35940/IJITEE.C9804.0311422.

[29] D. M. Divakaran and A. Oest, “Phishing Detection Leveraging Machine Learning and Deep Learning: A Review,” *IEEE Secur Priv*, vol. 20, no. 5, pp. 86–95, 2022, doi: 10.1109/MSEC.2022.3175225.

[30] J. Kolla, S. Praneeth, M. S. Baig, and G. reddy Karri, “A comparison study of machine learning techniques for phishing detection,” *Journal of Business and Information Systems (e-ISSN: 2685-2543)*, vol. 4, no. 1, pp. 21–33, Jun. 2022, doi: 10.36067/JBIS.V4I1.120.

[31] N. Puri, P. Saggar, A. Kaur, and P. Garg, “Application of ensemble Machine Learning models for phishing detection on web networks,” *Proceedings - 2022 5th International Conference on Computational Intelligence and Communication Technologies, CCICT 2022*, pp. 296–303, 2022, doi: 10.1109/CCICT56684.2022.00062.

[32] J. Tanimu and S. Shiaeles, “Phishing Detection Using Machine Learning Algorithm,” *Proceedings of the 2022 IEEE International Conference on Cyber Security and Resilience, CSR 2022*, pp. 317–322, 2022, doi: 10.1109/CSR54599.2022.9850316.

[33] A. Hannousse and S. Yahiouche, “Towards benchmark datasets for machine learning based website phishing detection: An experimental study,” *Eng Appl Artif Intell*, vol. 104, p. 104347, Sep. 2021, doi: 10.1016/J.ENGAPPAI.2021.104347.

[34] U. Ozker and O. K. Sahingoz, “Content Based Phishing Detection with Machine Learning,” *2020 International Conference on Electrical Engineering, ICEE 2020*, Sep. 2020, doi: 10.1109/ICEE49691.2020.9249892.

[35] M. Das, S. Saraswathi, R. Panda, A. K. Mishra, and A. K. Tripathy, “Exquisite Analysis of Popular Machine Learning–Based Phishing Detection Techniques for Cyber Systems,” *Journal of Applied Security Research*, vol. 16, no. 4, pp. 538–562, 2021, doi: 10.1080/19361610.2020.1816440.

[36] M. M. Uddin, K. Arfatul Islam, M. Mamun, V. K. Tiwari, and J. Park, “A Comparative Analysis of Machine Learning-Based Website Phishing Detection Using URL Information,” *2022 5th International Conference on Pattern Recognition and Artificial Intelligence, PRAI 2022*, pp. 220–224, 2022, doi: 10.1109/PRAI55851.2022.9904055.

[37] A. Chandra, Gregorius, M. S. J. Immanuel, A. A. S. Gunawan, and Anderies, “Accuracy Comparison of Different Machine Learning Models in Phishing Detection,” *ICOIACT 2022 - 5th International Conference on Information and Communications Technology: A New Way to Make AI Useful for Everyone in the New Normal Era, Proceeding*, pp. 24–29, 2022, doi: 10.1109/ICOIACT55506.2022.9972107.

[38] N. Jagdale and P. Chavan, “Hybrid Ensemble Machine Learning Approach for URL Phishing Detection,” *2022 2nd Asian Conference on Innovation in Technology, ASIANCON 2022*, 2022, doi: 10.1109/ASIANCON55314.2022.9908667.

[39] S. Das Guptta, K. T. Shahriar, H. Alqahtani, D. Alsalman, and I. H. Sarker, “Modeling Hybrid Feature-Based Phishing Websites Detection Using Machine Learning Techniques,” *Annals of Data Science*, 2022, doi: 10.1007/s40745-022-00379-8.

[40] M. Korkmaz, O. K. Sahingoz, and B. Diri, “Detection of Phishing Websites by Using Machine Learning-Based URL Analysis,” in *2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, 2020, pp. 1–7. doi: 10.1109/ICCCNT49239.2020.9225561.

[41] M. M. Yadollahi, F. Shoeleh, E. Serkani, A. Madani, and H. Gharaee, “An Adaptive Machine Learning Based Approach for Phishing Detection Using Hybrid Features,” *2019 5th International Conference on Web Research, ICWR 2019*, pp. 281–286, Apr. 2019, doi: 10.1109/ICWR.2019.8765265.

[42] O. K. Sahingoz, E. Buber, O. Demir, and B. Diri, “Machine learning based phishing detection from URLs,” *Expert Syst Appl*, vol. 117, pp. 345–357, Mar. 2019, doi: 10.1016/J.ESWA.2018.09.029.

[43] D. Mehanović and J. Kevrić, “Phishing website detection using machine learning classifiers optimized by feature selection,” *Traitement du Signal*, vol. 37, no. 4, pp. 563–569, Oct. 2020, doi: 10.18280/TS.370403.

[44] E. Nowroozi, Abhishek, M. Mohammadi, and M. Conti, “An Adversarial Attack Analysis on Malicious Advertisement URL Detection Framework,” *IEEE Transactions on Network and Service Management*, vol. 20, no. 2, pp. 1332–1344, Jun. 2023, doi: 10.1109/TNSM.2022.3225217.

[45] M. Aljabri *et al.*, “An Assessment of Lexical, Network, and Content-Based Features for Detecting Malicious URLs Using Machine Learning and Deep Learning Models,” *Comput Intell Neurosci*, vol. 2022, pp. 1–14, Aug. 2022, doi: 10.1155/2022/3241216.

[46] S. R. Abdul Samad *et al.*, “Analysis of the Performance Impact of Fine-Tuned Machine Learning Model for Phishing URL Detection,” *Electronics 2023, Vol. 12, Page 1642*, vol. 12, no. 7, p. 1642, Mar. 2023, doi: 10.3390/ELECTRONICS12071642.

[47] A. K. Jain, N. Debnath, and A. K. Jain, “APuML: An Efficient Approach to Detect Mobile Phishing Webpages using Machine Learning,” *Wirel Pers Commun*, vol. 125, no. 4, pp. 3227–3248, Aug. 2022, doi: 10.1007/S11277-022-09707-W/TABLES/7.

[48] J. Moedjahedy, A. Setyanto, F. K. Alarfaj, and M. Alreshoodi, “CCrFS: Combine Correlation Features Selection for Detecting Phishing Websites Using Machine Learning,” *Future Internet*, vol. 14, no. 8, Aug. 2022, doi: 10.3390/FI14080229.

[49] F. A. Ghaleb, M. Alsaedi, F. Saeed, J. Ahmad, and M. Alasli, “Cyber Threat Intelligence-Based Malicious URL Detection Model Using Ensemble Learning,” *Sensors*, vol. 22, no. 9, May 2022, doi: 10.3390/S22093373.

[50] M. Aljabri *et al.*, “Detecting Malicious URLs Using Machine Learning Techniques: Review and Research Directions,” *IEEE Access*, vol. 10, pp. 121395–121417, 2022, doi: 10.1109/ACCESS.2022.3222307.

[51] M. Kim, D. Kim, C. Hwang, S. Cho, S. Han, and M. Park, “Machine-learning-based android malware family classification using built-in and custom permissions,” *Applied Sciences (Switzerland)*, vol. 11, no. 21, Nov. 2021, doi: 10.3390/APP112110244.

[52] I. U. Hassan, R. H. Ali, Z. U. Abideen, T. A. Khan, and R. Kouatly, “Significance of Machine Learning for Detection of Malicious Websites on an Unbalanced Dataset,” *Digital 2022, Vol. 2, Pages 501-519*, vol. 2, no. 4, pp. 501–519, Oct. 2022, doi: 10.3390/DIGITAL2040027.

[53] Y. Li, Z. Yang, X. Chen, H. Yuan, and W. Liu, “A stacking model using URL and HTML features for phishing webpage detection,” *Future Generation Computer Systems*, vol. 94, pp. 27–39, May 2019, doi: 10.1016/J.FUTURE.2018.11.004.

[54] A. Zamir *et al.*, “Phishing web site detection using diverse machine learning algorithms,” *Electronic Library*, vol. 38, no. 1, pp. 65–80, Mar. 2020, doi: 10.1108/EL-05-2019-0118.

[55] A. A. Orunsolu, A. S. Sodiya, and A. T. Akinwale, “A predictive model for phishing detection,” *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 2, pp. 232–247, Feb. 2022, doi: 10.1016/J.JKSUCI.2019.12.005.

[56] A. Subasi and E. Kremic, “Comparison of Adaboost with MultiBoosting for Phishing Website Detection,” *Procedia Comput Sci*, vol. 168, pp. 272–278, Jan. 2020, doi: 10.1016/J.PROCS.2020.02.251.

[57] A. A. Akinyelu, “Machine Learning and Nature Inspired Based Phishing Detection: A Literature Survey,” *https://doi.org/10.1142/S0218213019300023*, vol. 28, no. 5, Aug. 2019, doi: 10.1142/S0218213019300023.

[58] A. Cuzzocrea, F. Martinelli, and F. Mercaldo, “A machine-learning framework for supporting intelligent web-phishing detection and analysis,” *ACM International Conference Proceeding Series*, Jun. 2019, doi: 10.1145/3331076.3331087.

[59] A. N. Tultul, R. Afroz, and M. A. Hossain, “Comparison of the efficiency of machine learning algorithms for phishing detection from uniform resource locator,” *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 28, no. 3, pp. 1640–1648, Dec. 2022, doi: 10.11591/IJEECS.V28.I3.PP1640-1648.

[60] S. Y. Siddiqui, A. S. Akram, H. M. Usama, A. M. Momani, N. A. Al-Dmour, and W. T. Al-Sit, “Improved Hybrid Model for Phishing Detection by Using Machine Learning,” *International Conference on Cyber Resilience, ICCR 2022*, 2022, doi: 10.1109/ICCR56254.2022.9995980.

[61] M. F. A. Razak, M. I. Jaya, F. Ernawan, A. Firdaus, and F. A. Nugroho, “Comparative Analysis of Machine Learning Classifiers for Phishing Detection,” *Proceedings - International Conference on Informatics and Computational Sciences*, vol. 2022-September, pp. 84–88, 2022, doi: 10.1109/ICICOS56336.2022.9930531.

[62] Md. J. Raihan, Md. A.-M. Khan, S.-H. Kee, and A. Nahid, “Detection of the chronic kidney disease using XGBoost classifier and explaining the influence of the attributes on the model using SHAP,” *Sci Rep*, vol. 13, Apr. 2023, doi: 10.1038/s41598-023-33525-0.