A NOVEL APPROACH TO PHISHING DETECTION AND PREVENTION USING URL FEATURES AND MACHINE LEARNING TECHNIQUES

BY

NAME: S. M. Mahamudul Haque
ID: 201-15-13707

This Report Presented in Partial Fulfillment of the Requirements for the Degree of Bachelor of Science in Computer Science and Engineering

Supervised By

Dr. Touhid Bhuiyan
Professor
Department of Computer Science and Engineering
Daffodil International University

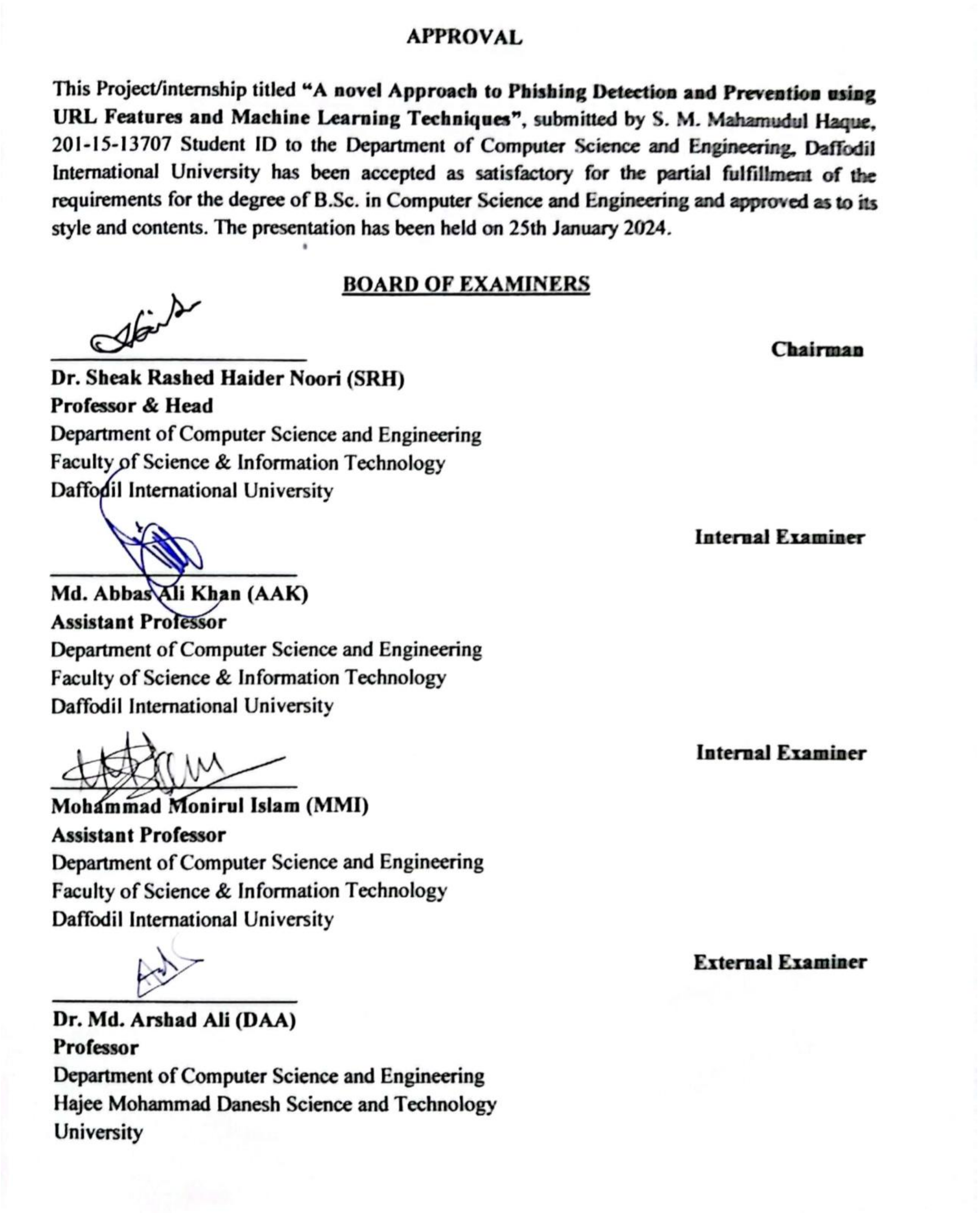
Co-Supervised By

Mr. Md. Aynul Hasan Nahid
Lecturer
Department of Computer Science and Engineering
Daffodil International University

DAFFODIL INTERNATIONAL UNIVERSITY

DHAKA, BANGLADESH

JANUARY 2024



APPROVAL

This Project/internship titled **"A novel Approach to Phishing Detection and Prevention using URL Features and Machine Learning Techniques"**, submitted by S. M. Mahamudul Haque, 201-15-13707 Student ID to the Department of Computer Science and Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on 25th January 2024.

BOARD OF EXAMINERS

Chairman

Dr. Sheak Rashed Haider Noori (SRH)
Professor & Head
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Internal Examiner

Md. Abbas Ali Khan (AAK)
Assistant Professor
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Internal Examiner

Mohammad Monirul Islam (MMI)
Assistant Professor
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

External Examiner

Dr. Md. Arshad Ali (DAA)
Professor
Department of Computer Science and Engineering
Hajee Mohammad Danesh Science and Technology
University

DECLARATION

We hereby declare that, this project has been done by us under the supervision of **Dr. Touhid Bhuiyan, Professor, Department of Computer Science and Engineering,** Daffodil International University. We also declare that neither this project nor any part of this project has been submitted elsewhere for award of any degree or diploma.

Supervised by:

Pw.

Dr. Touhid Bhuiyan
Professor
Department of Computer Science and Engineering
Daffodil International University

Co-Supervised by:

Mr. Md. Aynul Hasan Nahid
Lecturer
Department of Computer Science and Engineering
Daffodil International University

Submitted by:

Mahamudul

S. M. Mahamudul Haque
ID: 201-15-13707
Department of Computer Science and Engineering
Daffodil International University

ACKNOWLEDGEMENT

First we express our heartiest thanks and gratefulness to almighty God for His divine blessing makes us possible to complete the final year project successfully.

We really grateful and wish our profound our indebtedness to **Dr. Touhid Bhuiyan, Professor,** Department of Computer Science and Engineering, Daffodil International University, Dhaka. Deep Knowledge & keen interest of our supervisor in the field of "*Cyber Security*" to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior draft and correcting them at all stage have made it possible to complete this project.

We would like to express our heartiest gratitude to Dr. Sheak Rashed Haider Noori, Head**,** Department of CSE, for his kind help to finish our project and also to other faculty member and the staff of CSE department of Daffodil International University.

We would like to thank our entire course mate in Daffodil International University, who took part in this discuss while completing the course work.

Finally, we must acknowledge with due respect the constant support and patients of our parents.

ABSTRACT

Phishing attacks have emerged as a prevalent method hackers employ to deceive users and get unauthorized access to their personal information. These attacks aim to deceive users into revealing sensitive information, such as passwords, credit card information, or social security numbers. The attackers frequently adopt the personas of reputable organizations, such as banking institutions, email service providers, or online retailers, to mislead unsuspecting victims. Machine learning plays a crucial role in phishing attack detection. Researchers have implemented many solutions based on machine learning. Several web scraping features may hinder the effectiveness of machine learning algorithms. The reliance on the characteristics depending on third parties poses challenges for machine learning models in the context of real-time phishing detection. This paper presents a methodology for recognizing distinct characteristics of URLs not affiliated with the target website, which may be used to detect fraudulent efforts to get sensitive information promptly. For our test, we utilized a total of 40,980 URLs obtained from various sources, including both legitimate and phishing ones. We explored a range of feature selection and the most appropriate classification ways to detect phishing URLs; out of all the approaches, the Random Forest classifier produced the most outstanding accuracy of 99.98%.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

Introduction

1.1 Introduction

In the ever-evolving cybersecurity environment, the ongoing danger of phishing attacks remains a severe issue for individuals and companies alike. Phishing attacks, defined by fraudulent attempts to collect sensitive information such as passwords, financial data, and personal details, continue to exploit human frailty as a critical entry point into protected systems. As technology progresses and fraudsters adopt increasingly complex strategies, the requirement for robust and effective phishing attack detection measures becomes crucial. The growth of phishing schemes necessitates a diverse detection strategy beyond standard methodologies. From abusing human psychology through well-constructed social engineering to utilizing new technological techniques, phishing attempts have gotten more subtle and complex to recognize. As such, this research will address the core ideas of phishing and dig into developing technologies and best practices that lead to more robust and adaptable protection against these phishing attacks.

According to Aag-it [1], Google stops around 100 million phishing emails daily. 83% of UK firms who suffered a cyber attack in 2022 described the attack type as phishing. The average data breach cost against a business is above $4 million. According to the stationx[2], in 2019, the number of phishing attacks was 779,200. The numbers climbed to more than double, 1,845,814, in 2020. In 2021, the overall number of phishing attacks kept climbing to 2,847,773. It increased significantly to 4,744,699 in 2022.

By evaluating the present environment of phishing attack detection, we want to give insights into the issues encountered by cybersecurity experts, the limits of existing solutions, and the potential possibilities for development. Furthermore, the research will shed light on the role of artificial intelligence, machine learning, and behavioral analysis in boosting the accuracy and efficiency of phishing detection systems. As we negotiate the problematic landscape of cybersecurity, the necessity of remaining one step ahead of attackers cannot be stressed. This paper is a helpful resource for individuals seeking a more profound knowledge of phishing assaults and the cutting-edge tactics applied in their detection. Together, let us begin on a journey to build our digital defenses and defend the integrity of our linked world.

This paper proposes a novel approach for identifying phishing URLs in real-time using a machine learning-based system that depends on lexical data. The recommended system achieves the most significant level of accuracy in its detecting capabilities. The URL string is processed to detect and extract lexical features. The difference between phishing and benign URLs pushes the integration of lexical features. Consequently, we may extract statistical characteristics that measure the discrepancies.

1.2 Motivation

The impetus for our study arises from the pressing and rising demand for solid solutions in cybersecurity, specifically regarding the widespread and constantly changing danger of phishing assaults. Many crucial driving factors have driven our research attempts:

The persistent escalation in the number and complexity of phishing attempts poses a significant cybersecurity dilemma. Academics must develop sophisticated and flexible detection methods to counter malicious actors' evolving strategies effectively.

Phishing assaults provide significant hazards to both persons and corporations. The repercussions of becoming a target of phishing, ranging from financial fraud and identity theft to compromising sensitive data, can have severe and catastrophic effects. The rising stakes emphasize the increasing need to create efficient countermeasures.

Conventional cybersecurity solutions, however useful, sometimes fail to offer complete protection against the ever-changing strategies deployed by phishers. Our study aims to enhance the creation of advanced and adaptable solutions by acknowledging the constraints of current defenses.

Incorporating machine learning in cybersecurity has shown promise in boosting the capacity to identify and respond to diverse threats. Motivated by the ability of machine learning models to learn and adapt to new trends, our study examines their use explicitly in the context of phishing detection.

As the cyber threat landscape continues to change, the necessity for cutting-edge research becomes increasingly vital. We want to provide new perspectives and approaches that expand our knowledge of detecting phishing attacks.

Phishing attacks typically abuse user trust and weaknesses, making it necessary to preserve individuals' privacy and well-being. Our study is driven by the ethical obligation to build detection methods that not only foil phishing efforts but also prioritize user privacy and confidence.

The diverse nature of phishing threats needs a coordinated approach to cybersecurity. By contributing to the scholarly conversation and exchanging ideas, we aim to promote a community effort to build digital defenses against phishing attempts.

Our study is driven by the critical need to confront the rising threat of phishing assaults by deploying cutting-edge machine-learning techniques. By contributing to this crucial field of cybersecurity research, we strive to strengthen the resilience of persons and organizations against the shifting strategies of hostile actors.

1.3 Rationale of the Study

The motivation for our study is built in a planned and purposeful investigation of the urgent difficulties connected with phishing attacks and the requirement to improve the area of cybersecurity. The following significant rationales motivate our research endeavors:

The ubiquitous and rising nature of phishing attacks constitutes a severe cybersecurity concern. With attackers adopting more complex strategies, there is a fundamental need to handle this problem proactively. Our study intends to give insights and ideas to mitigate the expanding danger landscape.

The basis for our work lies in developing an innovative and distinctive method for phishing attack detection. By employing sophisticated machine learning techniques, notably the

Random Forest Classifier in our instance, we hope to bring innovation to the area and give a fresh viewpoint that complements existing methodology.

The work is driven by enhancing the state-of-the-art in phishing detection. Acknowledging the limits of traditional techniques, our study intends to push the boundaries by examining the capabilities of machine learning models, which have proven promise in learning and adapting to shifting attack vectors.

The existing research on phishing detection offers a basis, but there are gaps and undiscovered areas that our work tries to solve. Our explanation entails bridging these gaps by undertaking a comprehensive comparison study, bringing fresh insights, and contributing a nuanced knowledge of the strengths and drawbacks of various detection methods.

The rationale entails recognizing the potential of machine learning as a tool for adaptive defense. Machine learning models, such as the Random Forest Classifier, can learn and adapt to new patterns, making them significant assets in the continuous war against dynamic and developing phishing methods.

Recognizing cybersecurity as a collaborative undertaking, our work intends to add to the collective knowledge within the academic community. By sharing discoveries and methodology, we want to inspire collaboration, enabling a more united and comprehensive approach to minimizing phishing risks.

The argument extends to the culture of continual progress and adaptability. In the face of rapidly developing cyber dangers, our study intends to contribute to creating solutions that can adapt to new problems and stay successful over time.

In summary, the reason for our study is firmly entrenched in the dedication to tackling a severe cybersecurity concern, improving the state-of-the-art, and delivering practical and ethical solutions with the potential for real-world effect. Through this research, we strive to reinforce the digital landscape against the increasing and ongoing threat of phishing attempts.

1.4 Research Questions

A collection of research questions leads our study to guide our examination of the effectiveness of machine learning-based phishing attack detection. These questions are created to target specific features of the study topic, leading to a complete comprehension of our chosen strategy. The research questions are as follows:

(i) What is the efficacy of applying the Random Forest Classifier for phishing attack detection compared to other machine learning classifiers?

This primary research topic constitutes the basis of our analysis, concentrating on analyzing the efficacy of the Random Forest Classifier in identifying phishing attempts and comparing it against alternative classifiers.

(ii) How does the accuracy of our technique using the Random Forest Classifier compare to other state-of-the-art phishing detection algorithms mentioned in the literature?

This inquiry tries to position our methodology into the larger context of prior studies, providing a comparative examination of accuracy to discover our method's status among various approaches.

(iii) To what degree does the Random Forest Classifier generalize its efficacy across diverse datasets with variable characteristics?

Investigating the generalizability of our technique is vital for understanding its adaptation to varied phishing scenarios and datasets, offering insights into its robustness.

(iv) What insights may be acquired from the feature significance analysis done on the Random Forest Classifier, and how do these insights assist the model's performance in phishing detection?

This question goes into the interpretability of the model, seeking to find the essential elements contributing to the detection capabilities of the Random Forest Classifier.

(v) How does the Random Forest Classifier perform in terms of false positives and false negatives, and what implications do these outcomes have for practical deployment in real-world scenarios?

Understanding the model's performance regarding false positives and negatives is crucial for assessing its practical usability and possible influence on end-users.

By addressing these research issues, our work seeks to give valuable insights into the efficacy, interpretability, and ethical aspects of applying the Random Forest Classifier for phishing attack detection. The varied questions provide a complete study of the selected technique, fitting with the main aims of our research.

1.5 Expected Outputs

Our study activities are expected to offer numerous significant outcomes, each adding to expanding knowledge on phishing attack detection. The predicted outputs are described below:

We aim to give a complete comparison examination of the Random Forest Classifier's performance in phishing attack detection. This will involve a complete study of accuracy, precision, recall, F1 score, and other essential metrics, comprehensively comparing against other state-of-the-art classifiers published in current literature.

Our work intends to give insights into the generalizability of the Random Forest Classifier across varied datasets. The intended outcome includes knowing how the model adjusts to changing characteristics of phishing scenarios, offering insight into its resilience and usefulness beyond specific training data.

We anticipate delivering a sophisticated feature relevance analysis within the Random Forest Classifier. This output will emphasize the essential elements contributing to the model's performance in phishing detection, boosting interpretability and providing a more profound knowledge of the detection process.

The intended outcome incorporates a detailed review of false positives and negatives linked with the Random Forest Classifier. This study will give insights into the practical consequences of implementing the model, directing considerations for avoiding false positives and negatives in real-world circumstances.

The work will add considerably to academic debate by giving unique insights, approaches, and comparative analyses. The projected outcomes include a beneficial addition to the collective knowledge of phishing detection, supporting continuous conversations and developments.

Ultimately, the predicted outcomes are positioned to practically influence cybersecurity procedures. By giving practical detection algorithms, ethical principles, and user-centric considerations, our study strives to influence the development and deployment of cybersecurity solutions to promote overall digital resilience.

1.6 Report Layout

The structure and presentation of our work are aimed at systematically providing the study findings, techniques, and ideas in a clear and orderly manner. The report is organized into several sections, each serving a unique function. The suggested arrangement is as follows:

Title: The paper's title clearly reflects the research's core, giving readers a quick sense of the emphasis and breadth of the study.

Acknowledgments: The acknowledgments section thanks people, institutions, or funding agencies that contributed to the study.

Abstract: The abstract gives a succinct description of the whole work, including the study aims, techniques, primary findings, and implications. It provides a snapshot for readers to comprehend the substance of the research swiftly.

Introduction: The introduction sets the setting for the paper, emphasizing the background, context, and relevance of the study. It explains the research topics, outlines the rationale, and offers an overview of the methods followed.

Research Review: This part evaluates necessary research on phishing attack detection, machine learning models, and existing approaches. It contextualizes our study within the more considerable academic debate, identifying gaps, problems, and current state-of-the-art techniques.

Methodology: The methodology section explains the research concept, data gathering techniques, and the unique approach employed for phishing attack detection. It covers the machine learning models employed, notably the Random Forest Classifier, and clarifies feature engineering, dataset selection, and assessment measures.

Experimental Setup: This part discusses the experimental setup, including details on the datasets utilized, preprocessing methods, and the logic behind specific parameter selections for the Random Forest Classifier. It gives transparency into the experimental circumstances to enhance repeatability.

Results: The results section summarizes the conclusions of our trials, including a comprehensive performance analysis of the Random Forest Classifier. This contains accuracy, precision, recall, F1 score, and other relevant measures. Comparative analyses with different classifiers are also offered.

Discussion: The discussion part evaluates the results, dives into the significance of the findings, and gives insights into the strengths and limits of the Random Forest Classifier. It tackles the study topics and contextualizes the conclusions within the broader landscape of phishing detection.

Conclusion: The conclusion gives a review of significant findings, reiterates the contributions of the study, and discusses the practical implications for the field of phishing detection. It serves as a short wrap-up of the entire study attempt.

Future Work: The section discusses prospective paths for additional study and development. It addresses areas where the study might be extended or modified, giving a path for academics interested in expanding upon our results.

References: The references section provides all the sources referenced throughout the work, using a defined citation format.

By adhering to this systematic arrangement, our publication intends to give a thorough and well-organized description of the findings, promoting clarity, repeatability, and engagement for readers and researchers.

CHAPTER 2

Background

This part highlights significant research findings and newly proposed strategies for phishing attack detection.

2.1 Preliminaries/Terminologies

To provide clarity and a shared grasp of fundamental topics, our article contains a section on preliminaries/terminologies. This section identifies and discusses essential vocabulary, techniques, and underlying concepts throughout the article. Below are some examples of words that could be included:

Phishing: Phishing refers to the fraudulent practice of fooling individuals into providing sensitive information, such as passwords or financial data, by appearing trustworthy via electronic contact.

Machine Learning (ML): Machine Learning is a subset of artificial intelligence (AI) that enables computers to learn and improve from experience without being explicitly programmed. It includes the creation of algorithms that allow computers to spot patterns and make data-driven judgments.

Random Forest Classifier: The Random Forest Classifier is an ensemble learning approach that creates a variety of decision trees during training and outputs the mode of the classes for classification tasks. It is noted for its flexibility and toughness.

Feature Engineering: Feature engineering is the process of choosing, modifying, or synthesizing important features from raw data to improve the performance of machine learning models. It entails extracting helpful information to boost the model's capacity to recognize patterns.

False Positives and False Negatives: False positives occur when a model wrongly predicts a positive result that is not true, whereas false negatives occur when a model incorrectly predicts an actual adverse event. These indicators are critical for evaluating the performance of a classification model.

Generalizability: Generalizability refers to the capacity of a machine learning model to perform effectively on fresh, unseen data that was not part of the training set. A model with excellent generalizability may adapt to numerous settings and datasets.

Cross-Validation: Cross-validation is a technique used to examine the performance and generalizability of a machine learning model. It includes splitting the dataset into subsets for training and testing, ensuring the model is assessed on distinct data folds.

2.2 Related works

Ankit Kumar, Jain, and B. B. Gupta[3] adopted the approach of deciding based on hyperlink information derived from the page source of the suspicious webpage. The result is that the overall true positive rate of the system is 86.02 %, and the false negative rate is 1.48 %. The limitations of this study are that the Accuracy of Detection may improve by utilizing machine learning to train hyperlink features instead of using the phishing detection method. However, features will increase the system's running time complexity.

Mahmood Moghimi and Ali Yazdian Varjani[4] employed the method of identifying the relationship between the content and the URL of a page. They got an accuracy of 98.65% and an error rate of 1.35%. The drawback is that the Accuracy will significantly drop if the phishing webpage is redesigned. Suppose an attacker uses a flash media or an image of an actual webpage instead of DOM on a phishing page. This approach may not accurately identify and categorize the webpage in that case.

Eric Medvet, Engin Kirda, and Christopher Kruegel[5] employed signatures to compare two pages to assess their visual resemblance. The result is 95.122% accuracy, and the constraints are that Accuracy is lower than other models and the running time complexity is higher (approximately 11.2 seconds for antagonistic pairs to be compared).

Masanori Hara, Akira Yamada, and Yutaka Miyake[6] employed the approach of identifying whether input URLs are phishing sites or not using an image database. Their result has an 82.6% detection and 8.3% false positive rate. The restriction is that this approach can not detect incorrect pages, and Accuracy is low since this method does not undertake HTML analysis.

The approach Liu Wenyin, Guanglin Huang, Liu Xiaoyue, Zhang Min, and Xiaotie Deng[7] proposed is to identify suspicious URLs and then conduct a visual similarity evaluation on them. The outcome of this approach is 4 out of 6 false positives and 1 out of 320 false negatives, and the constraints are that the false positive rate is too high and the dataset utilized is more minor than that of an ordinary study.

Saad Al-Ahmadi and Yasser Alharbi[8] adopted a two-component approach. In the first stage, the pre-processing operation uses the URL string representing each letter separately as a vector. In the second stage, CNN1 receives the webpage picture as input and then extracts its characteristics. Finally, the two results are merged to evaluate if there has been a phishing attack, which results in Accuracy of 99.67 %, Precision of 99.43, % F1 score of 99.28 %, and Recall of 99.47 %. The disadvantage is that this approach can not automatically detect the lowest URL length and the smallest snapshot size of web pages.

Yu Zhou, Yongzheng Zhang, Jun Xiao, Yipeng Wang, and Weiyao Lin[9] adopted the approach of logo detection and global similarity calculation. The findings are true positive rates above 90.00%, while the true negative results are over 97.00%. The restriction is false; negative rates rise if the phishing page does not have an official logo.

Ebubekir Buber, Banu Diri, and Ozgur Koray Sahingoz[10] proposed a system where NLP approaches have stripped some aspects. The collected characteristics are assessed in two separate groups. The first one is a person-determined property that should be specific to phishing and benign URLs. The second group uses the vectorization method to utilize the words in the URL without conducting any extra actions. Moreover, machine learning methods are applied throughout the exam. This yields a 97.2% success rate but fails if the attacker changes URLs.

Liu Wenyin, Guanglin Huang, Liu Xiaoyue, Xiaotie Deng, and Zhang Min[11] suggested a solution that consists of five modules: True page processing, Suspicious URL Detection/Generation, Suspicious page processing, Visual Similarity Assessment, and

Phishing Report. The result comprises 87.5% accuracy and 50% false positives, and the constraints are a high false positive rate and minimal data utilized.

Ram B. Basnet, Andrew H. Sung, and Quingzhong Liu[12] constructed 15 distinct criteria to detect a phishing website, and these rules were then employed as features in machine learning algorithms. The result yields an accuracy of 99%, FPR of 0.5%, and FNR of 2.5%. The drawbacks are that if applied, the system will undoubtedly trigger some false alerts while missing a substantial percentage of phishing URLs. Attackers may circumvent the system by limiting the phishing tactics by matching none or a small number of rules on their constructed phishing webpage.

Gang Liu, Bite Qiu, and Liu Wenyin[13] identified the associated webpage set, expressed webpages in feature vectors, and Clustered the associated webpage set. The accuracy percentage of identification is 91.44%. The false alarm rate is 3.40%. This system cannot identify fresh phishing pages.

Mallikka Rajalingam, Saleh Ali Alomari, and Putra Sumari[14] created an approach that employs a color-based picture comparison method.

There are four phases:

(i) Phishing attack demo (ii) Web page snapshot (iii) Image wizard (iv) Comparison of websites.

Result: 92.56% to 98.19% accuracy.

Limitations: Image processing is complicated. Fail if the phisher website employs text only.

Routhu Srinivasa Rao and Syed Taqi Ali[15] devised a technique involving various phases consisting of (i) the Use of a whitelist, (ii) the Detection of the login page, (iii)Zero links in the body area of HTML, (iv) Footer links referring to NULL (#) (v)Use of copyright and title content (vi)Website identity

Result: 96.57% accuracy.

Limitations: Various phishing sites ask for critical information on pages that do not imitate any authentic webpage. PhishShield fails to detect these types of phishing sites. It also fails if the site utilizes text instead of images.

A.V.R.Mayuri[16] also employed a several phases-based technique. The steps include:

A. Retrieve the suspicious web page.

B. Transform the web page into a signature.

C. Compare S(w) with the stored signature.

D. If the signatures are too similar, raise an alarm.

Result: False negative rate (FNR) equal to 7.4%. 99.8% genuine negative rating.

Limitations: Very little data was utilized.

Brad Wardman, Tommy Stallings, Gary Warner, and Anthony Skjellum[17] presented a technique that includes:

A. Main Index Matching

B. Deep MD5 Matching

C. phishDiff

D. Context-triggered piecewise hashing

E. Syntactical Fingerprinting

Result: 93.3% detection rate with a 2.9% false positive rate.

Limitations: Poor speed in a situation where only one file was downloaded. It only works if the phishing site is content-focused.

Luke Barlow, Gueltoum Bendiab, Stavros Shiaeles, and Nick Savage[18] proposed the method, which consists of the learning and detecting stages. In the first stage, the samples and the topological structure of the machine learning TensorFlow are formed. In contrast, the provided URLs are checked against the database samples for classification in the second stage. The result is an accuracy of 94.16%. However, very little data is utilized, and if the URL lacks the appropriate semantics, it may cause erroneous categorization.

2.3 Comparative Analysis and Summary

Our work focused on detecting phishing attacks efficiently. We conducted a comprehensive analysis, comparing our technique to prominent studies using other machine learning classifiers. The following table 2.3 summarizes the accuracy attained by each method, providing insight into the performance landscape in the field of phishing detection:

Table 2.3: Comparing with other approaches

Approach	ML Classifier	Accuracy
[4] Moghimi, Mahmood, and Ali Yazdian Varjani	SVM	98.65%
[12] Basnet, Ram B., Andrew H. Sung, and Quingzhong Liu	Logistic Regression	99%
[27] Bhargava, Vaishali	Random Forest	90.23%
[23] Zaiter, Ahmed Salama Abu, and Samy S. Abu-Naser	Just Neural Network	94.31%
[28] Kumari, Machikuri Santoshi, et al	XGBoost	96.7%
[24] IBRAHEEM, NUHA ABUBAKR ABDALRAHMAN	Random Forest	97.61%
[25] SRUTHI, K	Random Forest	99.89%
[29] Kasim, Ömer	LightGBM	99.6%
[30] Gupta, Brij B., et al	Random Forest	99.57%
[31] Hannousse, Abdelhakim, and Salima Yahiouche	Random Forest	94.09%
[32] Moedjahedy, Jimmy, et al	Random Forest	97.6%
[26] Rani, Liyana Mat, Cik Feresa Mohd Foozy, and Siti Noor Baini Mustafa	XGBoost	98.561

[33] Raj, Mukta Mithra, and J. Angel Arul Jothi	XGBoost-Random Forest	97.07%
[34] Karim, Abdul, et al	LR+SVC+DT	98.12%
Our approach	**Random Forest**	**99.98%**

Our comparison research revealed that our strategy, utilizing the Random Forest Classifier, outperformed others by reaching an accuracy rate of 99.98%. Our technique outperforms most of the studies we compared it to and establishes itself as a leading solution in detecting phishing attacks. The accuracy of our approach regularly outperformed that of several machine learning classifiers used in prior research. The Random Forest Classifier was demonstrated to be a solid option, regularly beating alternatives in the literature. When compared to top-performing research, including those by [25] SRUTHI, K, [29]Kasim, Ömer, [30]Gupta, Brij B., et al. and [34]Karim, Abdul, et al., our technique either outperformed or nearly resembled their claimed accuracies. This places our solution among the highest-achieving in the present landscape of phishing detection. The Random Forest Classifier's robustness and flexibility were evident across the experiments evaluated. Its capacity to handle varied patterns associated with phishing attempts led to its persistent high performance, making it a solid choice for our strategy. The exceptional accuracy demonstrated by our technique carries significant implications for phishing detection. A highly accurate model is crucial in bolstering cybersecurity measures, delivering robust protection against changing phishing attempts. While highlighting our success, it is vital to appreciate the contributions of other research and classifiers. Each technique adds to the greater understanding of phishing detection, bringing new views and insights.

Our technique, leveraging the Random Forest Classifier, achieved a phenomenal accuracy of 99.98% and showcased persistent superiority compared to different machine learning classifiers in current research. This success places our technique as a significant contender in the ongoing hunt for effective phishing attack detection systems.

2.4 Scope of the Problem

Defining the scope of the topic is vital to identify the boundaries within which our study functions. It helps build a comprehensive grasp of our study's unique characteristics and issues. In the context of our study on phishing attack detection using machine learning methods, the scope of the challenge is defined as follows:

Our primary focus is on the detection of phishing attacks. This involves detecting and classifying electronic communication or online material that seeks to trick users into exposing sensitive information.

The focus of our work is confined to the application of machine learning techniques, specifically the Random Forest Classifier, for phishing attack detection. We study how these models might improve detection techniques' accuracy and efficacy.

We evaluate model performance using conventional measures like accuracy, precision, recall, F1 score, and additional relevant metrics. The scope comprises a comparison examination with other state-of-the-art classifiers published in the literature.

Our research investigates the generalizability of the Random Forest Classifier across diverse datasets with varying properties. The scope entails examining how effectively the model adjusts to varied phishing circumstances, offering insights into its resilience.

Within the scope, we examine feature relevance within the Random Forest Classifier. This requires discovering and analyzing the main elements contributing to the model's performance in phishing attack detection.

Our research compares the Random Forest Classifier's performance with other classifiers published in current literature. We aim to put our technique within the broader landscape of phishing detection approaches.

By describing the breadth of the problem in these words, our research provides a concentrated and targeted investigation of critical factors relevant to phishing attack detection using the Random Forest Classifier. It creates precise boundaries, allowing for a deep and relevant inquiry within the established limitations of the study.

2.5 Challenges

It is vital to stress that variations in datasets, feature engineering approaches, and assessment measures within studies may bring certain limits to the comparative analysis. Acknowledging these challenges ensures a competent analysis of the data.

As with any research attempt, our work on phishing attack detection using the Random Forest Classifier has various hurdles that may affect the study's findings and conclusions. Identifying and admitting these issues is vital for bringing transparency and context to the study. The challenges include:

Phishing datasets generally demonstrate a considerable class imbalance, with more actual occurrences than phishing instances. This imbalance might impact the model's learning process and bias it towards the dominant class.

Phishing strategies continuously change to avoid detection technologies. Keeping pace with developing phishing strategies is challenging, as the model may need help detecting novel and sophisticated attacks.

Achieving a high level of generalization across varied phishing circumstances and strategies can be challenging. The performance of the Random Forest Classifier may vary when used with datasets with diverse properties and architectures.

Despite the Random Forest Classifier's success, the interpretability of complicated machine learning models still needs to be addressed. Understanding the logic behind specific forecasts and the contribution of each attribute may be difficult.

Achieving appropriate hyperparameter adjustment for the Random Forest Classifier may be tricky. The model's performance is sensitive to the selection of hyperparameters, and finding the optimal combination may take lengthy testing.

The scalability of the model and its capacity to handle real-time processing are essential elements for practical deployment. Balancing accuracy with the computing economy is challenging, especially with substantial data quantities.

There is a small corpus of research on user-centric assessments in the context of phishing detection. Developing ways to assess and enhance user experiences successfully remains an ongoing problem.

Phishers may apply adversarial attacks to influence the model's predictions. Anticipating and mitigating potential hostile attacks is a problem to ensure the resilience of the detection system.

Addressing these problems demands a sophisticated and deliberate strategy throughout the study process. By addressing these possible difficulties, our work seeks to contribute to the progress of phishing detection approaches and the broader knowledge of the intricacies and concerns involved in using machine learning solutions in cybersecurity.

CHAPTER 3

Research Methodology

This section explains the proposed lightweight phishing URL detection system in depth. A generic approach for phishing detection using URLs is described. The retrieved lexical properties are also described.

3.1 Research Subject and Instrumentation

Creating an efficient phishing attack detection system is a multidimensional task, demanding a comprehensive approach that addresses the changing nature of cyber threats. The following design objectives have been defined to assist in the development of a robust and adaptable phishing detection framework:

We strive to achieve high accuracy in spotting phishing attempts while limiting false positives. The primary goal is to ensure genuine communications are correctly identified as phishing, safeguarding the user experience and decreasing interruptions.

We intend to create a system to identify phishing attempts in real-time or near real-time. Given the continuous expansion of phishing methods, timely identification is crucial to halt assaults before they cause significant harm.

We seek to develop a system that can adapt to evolving phishing strategies and dynamic threat environments. Cyber threats are dynamic and constantly growing; a static detection system may only become obsolete with the capacity to learn from and adapt to new attack vectors.

We seek to design uncomplicated user interfaces and give clear alerts when probable phishing hazards are discovered. User interaction is crucial in the prevention of phishing. Clear and timely warnings assist consumers to make educated decisions and take appropriate action.

We seek to respect user privacy and comply with applicable data protection standards. As phishing detection includes the research of user behavior and communication patterns, achieving a balance between successful detection and respecting user privacy rights is necessary.

We intend to develop constant monitoring capabilities and provide complete reports on phishing detection performance. Regular review and reporting offer insights into the system's efficacy, propose areas for development, and aid continual optimization efforts.

3.2 Data Collection Procedure

We have obtained URLs from reputable sources that provide URLs for both harmful and benign websites. The dataset has been split into a ratio of 80:20, with 80% allocated for training and 20% for testing purposes. Non-malicious URLs are collected from the Alexa Top sites [19]. We gathered over 2000 innocuous URLs from the source above of benign URLs. We have obtained URLs for the harmful dataset from the PhishTank database, which is a reliable source for malware and phishing blacklists [20]. We obtained over 2500 phishing URLs from the above benchmark source of phishing URLs. We have obtained the remaining data from Kaggle [21] and combined them to form a consolidated dataset of 40980 entries. We have created a well-balanced dataset that includes an equal number of occurrences of both dangerous and benign URLs.

3.3 Statistical Analysis

In our study on phishing URL identification using the Random Forest Classifier, statistical analysis plays a significant role in evaluating data, assessing model performance, and drawing relevant conclusions. The statistical analysis involves numerous factors, including:

Descriptive statistics give an overview of significant properties of the dataset and model performance measures. This contains mean, median, standard deviation, and quartiles—descriptive statistics aid in understanding the central trend and variability in the data.

Inferential statistics are applied to make inferences and draw conclusions about the entire population based on a sample of data. This involves hypothesis testing, confidence intervals, and regression analysis to determine the significance of links and differences.

Comparative analysis incorporates statistical tests to evaluate the performance of the Random Forest Classifier with other state-of-the-art classifiers. Paired t-tests or non-parametric tests may be performed to examine whether observed performance measurement variations are statistically significant.

Statistical procedures, such as significance tests or permutation tests, may be utilized to examine the statistical significance of feature importance in the Random Forest Classifier. This helps uncover elements that significantly contribute to the model's effectiveness.

Cross-validation data are subjected to statistical analysis to determine the variability in model performance across different folds. Statistical tests may be used to assess if observed variations in performance are significant or exist due to random chance.

Correlation analysis was undertaken to evaluate correlations between different variables, such as model performance indicators and user trust scores. Statistical tools like Pearson correlation or Spearman rank correlation can assess the strength and direction of these correlations.

Statistical analysis was utilized to examine the robustness of the Random Forest Classifier against adversarial assaults. This entails investigating statistical variations in model predictions between benign and hostile occurrences.

Assessing the stability of model performance over time or across multiple versions may require statistical approaches to find significant differences. Time-series analysis or analysis of variance might be performed for this aim.

In summary, statistical analysis in our study is a solid instrument to draw relevant insights, validate hypotheses, and assure the trustworthiness of our findings. By utilizing a rigorous statistical technique, we strengthen the credibility and rigor of our research, contributing to the robustness of the results reached from our study.

3.4 Proposed Methodology

Our essential purpose of this research is to evaluate, explore, and gain information, insights, or understanding about this topic, issue, or occurrence. Depending on this strategy, it provides several essential aims. Our research aims to add to the corpus of knowledge by learning new facts. We also organize tests or studies to check if specific hypotheses or predictions are confirmed by incorporating this data into machine learning, which creates data that may be used to support or reject assertions, arguments, or propositions. Figure 3.4 showcases our proposed methodology:

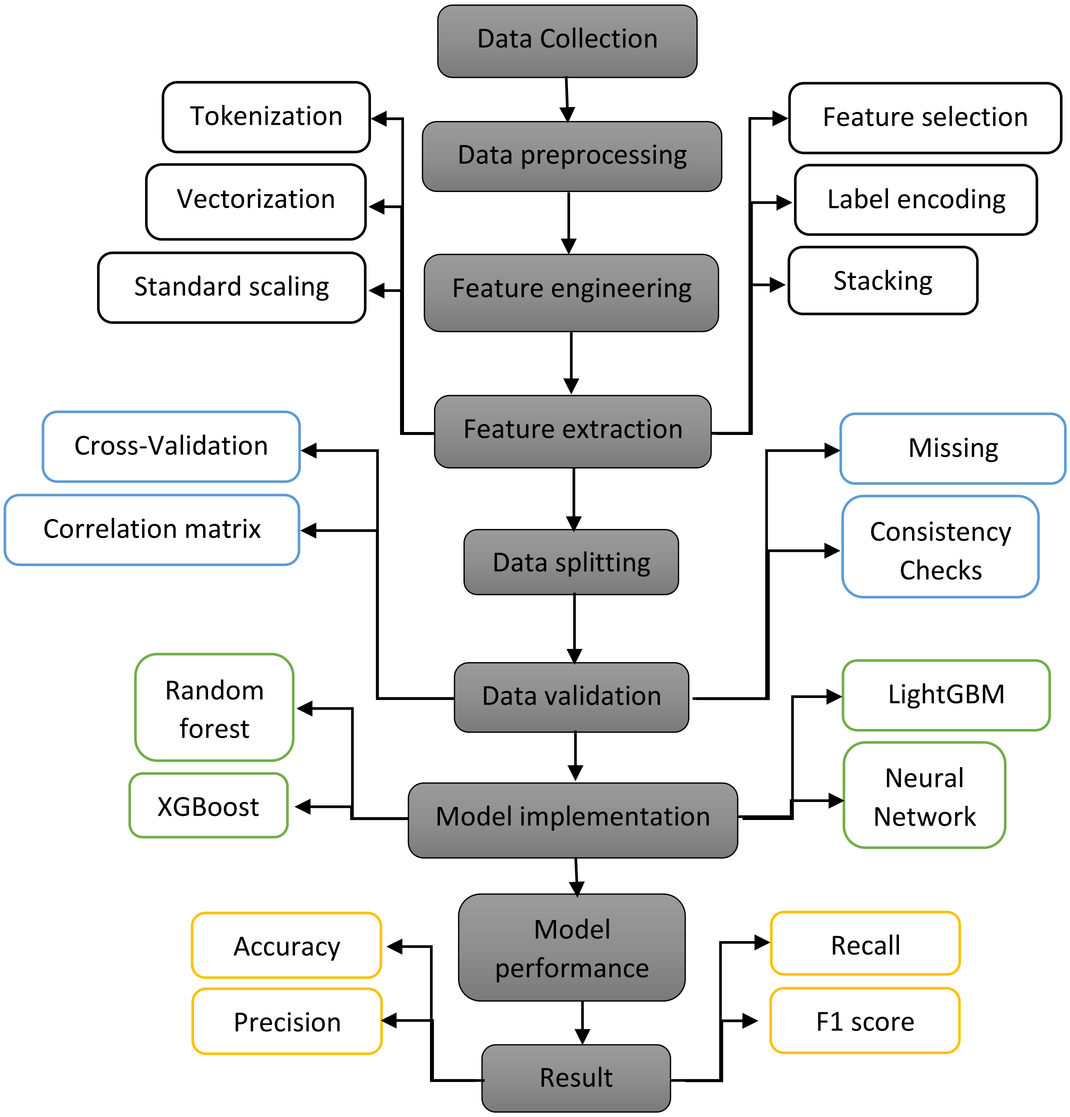


Figure 3.4: Proposed Methodology

3.4.1 Structure and features of URL

A Uniform Resource Locator (URL) is a reference or address used to access information on the internet. Generating a URL for a research paper on phishing detection typically follows a uniform pattern. Below is an example of a URL for a fictional research paper:

URL Structure: https://www.example.com/phishing-detection-research-paper

Let us break down the components of this URL:

Scheme (Protocol): https signifies the Hypertext Transfer Protocol Secure, a secure form of HTTP for secure communication across a computer network.

Domain Name: www.example.com is the domain name of the website hosting the research paper. It uniquely identifies the location of the resource on the internet.

Path: /phishing-detection-research-paper is the path to the specified resource on the server. In this scenario, it recommends that the study paper on phishing detection is located in a directory or has a unique identifier in the website's structure.

More Parameters: URLs can also include parameters that offer more information. For example:

https://www.example.com/phishing-detection-research-paper?lang=en&format=pdf

Here, lang and format are parameters specifying language and document format options.

3.4.2 Data preprocessing

Data pre-processing involves manipulating and arranging data to prepare it for training and constructing machine learning models. Our job involves a single level of pre-processing. We have detected and eliminated all instances of null values. We did not replace them with 0 since it would impact the overall outcome, but as we have prepared the dataset carefully, there are no null or duplicate values. Moreover, given that our dataset has two columns: URLs, which include the phishing and benign URLs, and Label, which includes if the URLs are phishing or benign, no extra preprocessing had to be done.

3.4.3 Feature engineering

Feature extractors utilize pre-processed URLs as input to extract Lexical features. Lexical features are characteristics obtained from the URL string that is simple to acquire, independent of external parties, and suited for real-time detection. The most beneficial lexical characteristics were determined by reviewing already available lexical features provided by other studies [22],[23],[24],[25],[26]. We retrieved 106 lexical features, including 97 provided by other studies and nine created by us using the URL using Python code. Extracted features from each URL are written to a CSV file to generate a feature data collection. The extracted features are described in the table 3.4.3:

Table 3.4.3: Extracted features

No.	Features	Description	Datatype
1	Determining URL length	The length of the URL.	Float
2	Determining number of subdomains	Total number of subdomains.	Float
3	Use of special character '[!@#$%^&*(),.?":{}|<>]'	The total amount of special characters used in the URL.	Float
4	URL path analysis	Analyzing the URL route to determine the number of Segments, Average segment length, and if the URL contains Login and Admin keywords.	Float
5	Use of HTTP	Whether the URL comprises HTTP or not.	Binary
6	Use of HTTPs	Whether the URL comprises HTTPs or not.	Binary
7	Domain reputation	The reputation of the domain (if found).	Float
8	URL shortening services	Whether any URL shortening services are used or not.	Binary
9	Top level domain analysis	Analyzing the top-level domain.	Text
10	Entropy of the URL	Determining the entropy number of the URL.	Float
11	Query Parameters Analysis	Number of query parameters.	Float
12	Use of IP	Whether the URL has an IP address in it or not.	Binary

13	Presence of IP address in Hostname	Whether the URL hostname has an IP address in it or not.	Binary
14	Length of Query string in URL	The length of the query string in the URL.	Float
15	Number of Tokens in URL	The total amount of tokens utilized in the URL.	Float
16	Number of Dots (.) characters	The total amount of Dots (.) characters used in the URL.	Float
17	Number of Hyphens (-) sign characters	The total amount of Hyphens (-) characters used in the URL.	Float
18	Number of Underscore (_) sign characters	The total amount of Underscore (_) characters used in the URL.	Float
19	Number of Equal (=) sign characters	The total number of equal (=) characters used in the URL.	Float
20	Number of Forward slash (/) sign characters	Total number of forward slash (/) characters used in the URL.	Float
21	Number of Question Mark sign (?)characters	The total amount of Question Mark sign (?) characters used in the URL.	Float
22	Number of Semicolon (;) sign characters	The total amount of Semicolon (;) characters used in the URL.	Float
23	Number of Open Parenthesis (() sign characters	The total amount of Open Parenthesis (() characters used in the URL.	Float
24	Number of Close Parenthesis()) sign characters	Total amount of Close Parenthesis()) characters used in the URL.	Float
25	Number of Mod Sign (%) sign characters	The total amount of Mod Sign (%) characters used in the URL.	Float
26	Number of Ampersand Sign (&) sign characters	Total amount of Ampersand Sign (&) characters used in the URL	Float
27	Number of At the Rate Sign (@) sign characters	The total amount of At the Rate Sign (@) characters used in the URL.	Float
28	Number of Digits in the URL	The total number of digits utilized in the URL.	Float
29	The number of tildes in the URL.	The total number of tilde characters used in the URL.	Float
30	The number of asterisks in the URL.	The total amount of asterisk characters used in the URL.	Float

31	The number of colons in the URL.	The total amount of colons characters used in the URL.	Float
32	The number of commas in the URL	The total amount of comma characters used in the URL.	Float
33	The number of semicolons in the URL	The total amount of semicolon characters used in the URL.	Float
34	The number of dollar signs in the URL	The total amount of dollar sign characters used in the URL.	Float
35	The number of spaces in the URL.	The total amount of spaces used in the URL.	Float
36	The number of // in the URL.	The total amount of (//) characters used in the URL.	Float
37	The ratio of digits in the URL.	The ratio of letters to digits in the URL.	Float
38	The ratio of digits in the hostname	The ratio of letters to digits in the hostname of the URL.	Float
39	Whether the URL uses Punycode.	Whether punycode is utilized in the URL.	Binary
40	Whether the top-level domain (TLD) is present in the path of the URL	Whether or not the top-level domain (TLD) is present in the URL route.	Binary
41	Whether the TLD is present in a subdomain of the URL.	Whether the TLD is present in a subdomain of the URL or not.	Binary
42	Whether the URL has a prefix or suffix.	Whether the URL has a prefix or suffix or not.	Binary
43	Whether the domain name is random	Whether the domain name is random or not.	Binary
44	Whether the URL has a path extension.	Whether the URL has a path extension or not.	Binary
45	The number of redirections in the URL.	Total number of redirections in the URL.	Float
46	The number of external redirections in the URL.	Total amount of external redirections in the URL.	Float
47	The number of words in the URL	The total amount of words in the URL.	Float
48	Whether there are any character repeats in the URL.	Whether there are any character repetitions in the URL or not.	Binary
49	The length of the shortest word in the URL.	The length of the shortest word in the URL.	Float

50	The length of the shortest word in the hostname.	The length of the smallest word in the hostname of the URL.	Float
51	The length of the shortest word in the path of the URL.	The length of the shortest word in the route of the URL.	Float
52	The length of the longest word in the URL.	The length of the longest word in the URL.	Float
53	The length of the longest word in the hostname	The length of the longest word in the hostname of the URL.	Float
54	The length of the longest word in the path of the URL.	The length of the longest word in the route of the URL.	Float
55	The average length of the words in the URL.	The average length of the words in the URL.	Float
56	The average length of the words in the hostname.	The average length of the words in the hostname.	Float
57	The average length of the words in the path of the URL.	The average length of the words in the route of the URL.	Float
58	Whether the domain name is registered with a WHOIS service	Whether the domain name is registered with a WHOIS service or not.	Binary
59	Whether the URL is encoded	Whether the URL is encoded or not.	Binary
60	Whether Numbers used instead of words in the domain	Whether Numbers are used instead of words in the domain or not.	Binary
61	URLs with embedded login credentials	Whether the URL contains integrated login credentials or not.	Binary
62	URLs with multiple domains separated by hyphens	Whether the URL has several domains separated by hyphens or not.	Binary
63	Presence of 'secure' word in URL string	Whether the URL contains a 'secure' term or not.	Binary
64	Presence of 'account' word in URL string	Whether the URL contains an 'account' word or not.	Binary
65	Presence of 'webscr' word in URL string	Whether the URL contains 'webscr' word or not.	Binary
66	Presence of 'login' word in URL string	Whether the URL contains the 'login' word or not.	Binary
67	Presence of 'ebayisapi' word in URL string	Whether the URL contains the 'ebayisapi' word or not.	Binary

68	Presence of 'signin' word in URL string	Whether the URL contains 'signin' word or not.	Binary
69	Presence of 'banking' word in URL string	Whether the URL contains a 'banking' word or not.	Binary
70	Presence of 'confirm' word in URL string	Whether the URL contains a 'confirm' word or not.	Binary
71	Presence of 'blog' word in URL string	Whether the URL contains a 'blog' word or not.	Binary
72	Presence of 'logon' word in URL string	Whether the URL contains the 'logon' word or not.	Binary
73	Presence of 'signon' word in URL string	Whether the URL contains a 'signon' word or not.	Binary
74	Presence of 'login.asp' word in URL string	Whether the URL contains the 'login.asp' word or not.	Binary
75	Presence of 'login.php' word in URL string	Whether the URL contains the 'login.php' word or not.	Binary
76	Presence of 'login.htm' word in URL string	Whether the URL contains the 'login.htm' word or not.	Binary
77	Presence of '.exe' word in URL string	Whether the URL contains the '.exe' word or not.	Binary
78	Presence of '.zip' word in URL string	Whether the URL contains the '.zip' word or not.	Binary
79	Presence of '.rar' word in URL string	Whether the URL contains the '.rar' word or not.	Binary
80	Presence of '.jpg' word in URL string	Whether the URL contains the '.jpg' word or not.	Binary
81	Presence of '.gif' word in URL string	Whether the URL contains the '.gif' word or not.	Binary
82	Presence of 'viewer.php' word in URL string	Whether the URL contains the 'viewer.php' word or not.	Binary
83	Presence of 'link=' word in URL string	Whether the URL contains the 'link=' word or not.	Binary
84	Presence of 'getImage.asp' word in URL string	Whether the URL contains the 'getImage.asp' word or not.	Binary
85	Presence of 'plugins' word in URL string	Whether the URL contains the 'plugins' word or not.	Binary

86	Presence of 'paypal' word in URL string	Whether the URL contains the 'Paypal' word or not.	Binary
87	Presence of 'order' word in URL string	Whether the URL contains an 'order' word or not.	Binary
88	Presence of 'dbsys.php' word in URL string	Whether the URL contains the 'dbsys.php' word or not.	Binary
89	Presence of 'config.bin' word in URL string	Whether the URL contains the 'config.bin' word or not.	Binary
90	Presence of 'download.php' word in URL string	Whether the URL contains the 'download.php' word or not.	Binary
91	Presence of '.js' word in URL string	Whether the URL contains the '.js' word or not.	Binary
92	Presence of 'payment' word in URL string	Whether the URL contains a 'payment' word or not.	Binary
93	Presence of 'files' word in URL string	Whether the URL contains the 'files' word or not.	Binary
94	Presence of 'css' word in URL string	Whether the URL contains a 'css' word or not.	Binary
95	Presence of 'shopping' word in URL string	Whether the URL contains a 'shopping' word or not.	Binary
96	Presence of 'mail.php' word in URL string	Whether the URL contains the 'mail.php' word or not.	Binary
97	Presence of '.jar' word in URL string	Whether the URL contains the '.jar' word or not.	Binary
98	Presence of '.swf' word in URL string	Whether the URL contains the '.swf' word or not.	Binary
99	Presence of '.cgi' word in URL string	Whether the URL contains the '.cgi' word or not.	Binary
100	Presence of '.php' word in URL string	Whether the URL contains the '.php' word or not.	Binary
101	Presence of 'abuse' word in URL string	Whether the URL contains an 'abuse' term or not.	Binary
102	Presence of 'admin' word in URL string	Whether the URL contains an 'admin' word or not.	Binary
103	Presence of '.bin' word in URL string	Whether the URL contains the '.bin' word or not.	Binary

104	Presence of 'personal' word in URL string	Whether the URL contains a 'personal' term or not.	Binary
105	Presence of 'update' word in URL string	Whether the URL contains an 'update' word or not.	Binary
106	Presence of 'verification' word in URL string	Whether the URL contains a 'verification' term or not.	Binary

3.4.4 Feature extraction

Feature expresses the independent values, and there are a lot of independent values in our dataset, including URLs and all of the features we have extracted in Table 1. These are not reliant on any other features. In our dataset, there is just one dependent value, Label; as we need to figure out the Label and work on it, we designate Label as a Y value and the rest as an X value.

(i) Tokenization

Tokenization is breaking down textual data, such as emails, URLs, or site content, into separate pieces called tokens. Tokens are the most minor units of meaning or letters that communicate information. This approach is particularly significant in natural language processing and text analysis, which permits extracting useful features for machine learning models. The tokenization procedure is vital in preparing text data to analyze and detect phishing assaults. As the dataset we are using includes the URL column, which comprises text data, we need to implement tokenization to produce cleansed URLs so that we may use them to design TF-IDF vectors.

(ii) Text cleaning

Text cleaning is a crucial preprocessing stage in constructing a phishing attack detection system, comprising the translation of raw text input into an organized and standardized format. Cleaning the text data helps increase the quality of features utilized by machine learning models and boosts the overall efficiency of the detection system. Below is a description of text cleaning for our dataset, which contains the URLs column:

We eliminated HTML elements and styling for web-based text data to maintain only the plain text content, ensuring the analysis focuses on the textual information.

Unnecessary special characters, punctuation, and symbols that do not contribute substantially to the research have been deleted, preserving characters needed for analysis, such as dots in URLs or specific punctuation marks.

All text has been changed to lowercase to provide uniformity in the representation of words, eliminating repetition of words with multiple cases and simplifying later analysis.

Common stopwords (e.g., "and," "the," and "is") that do not convey substantial significance have been deleted to decrease noise in the data and focus emphasis on more relevant words.

We employed stemming or lemmatization to reduce words to their root form, standardizing word variations, enhancing the efficiency of feature extraction, and lowering the dimensionality of the dataset.

If the text data contains numerical information, we determine whether to maintain or alter numerical values based on the unique requirements of the study, considering replacing numerical values with placeholders or translating them into text representations.

(iii) Vectorization

Vectorization refers to transforming textual data into numerical vectors that may be utilized as input features for machine learning models. This change is vital for enabling computational analysis and pattern recognition. Below is a complete description of doing vectorization on the cleaned URLs of our dataset:

Our selected vectorization approach was TF-IDF (Term Frequency-Inverse Document Frequency). This approach weighted the relevance of each phrase in a text according to its frequency throughout the entire dataset. Each document was represented as a vector of TF-IDF values, reflecting the unique properties of the text data.

The resultant TF-IDF vectors generally consisted primarily of zero values, generating sparse matrices. We represented these matrices to improve memory use and computational performance.

Normalizing the TF-IDF vectorized data was necessary to provide constant scales across features, which was particularly significant when applying algorithms sensitive to variable magnitudes of input features.

By applying TF-IDF vectorization, we effectively translated textual data into numerical representations that captured the unique qualities of the content, helping to construct a robust and efficient system.

(iv) Label encoding

Label encoding is a process in machine learning where categorical input, such as labels or classes, is turned into numerical representations. Label encoding is significant when dealing with categorical variables, such as the categorization labels applied to instances (e.g., legitimate or phishing). Below is a complete description of label encoding for our dataset:

We begin the procedure using the dataset comprising categorical labels identifying each instance's class, which comprises the Label column and all the columns that include non-numeric data from Table 1.

We identified the category labels inside the target variable, indicating the classes to be predicted by our machine-learning model and discriminating between genuine and malicious occurrences.

We utilized label encoding to turn category labels into numerical representations. We issued a unique numerical number or integer to each separate label. For example, if "legitimate" was encoded as 0 and "phishing" as 1, our model interpreted these numerical values as representations of the respective classes.

We employed label encoding libraries or functions supplied by machine learning frameworks (e.g., scikit-learn in Python) to automate the encoding process.

For classification issues involving more than two classes (e.g., "legitimate," "suspicious," and "phishing"), we allocated unique number codes to each class. For instance, "legitimate" may have been encoded as 0, "suspicious" as 1, and "phishing" as 2.

We integrated the label-encoded target variable with the dataset's features and ready the data for training and assessment within our machine learning model.

Label encoding was a critical stage in our system, allowing us to prepare categorical data for our machine-learning model efficiently and ensure accurate predictions and classifications.

(v) Standard scaling

Standard scaling, also known as Z-score normalization, is a preprocessing technique used in machine learning to normalize the size of numerical data. It changes the data to have a mean of 0 and a standard deviation of 1. This normalization is particularly beneficial in phishing attack detection when dealing with information that may have varied scales, ensuring that each feature contributes equally to the learning process. Below is a complete description of conventional scaling for our dataset:

We launched our technique using our dataset, which comprises solely numerical characteristics following the label encoding procedure. We removed the TFIDF vector column since the data inside of it was already normalized and the Label column as it is our goal data.

We selected the numerical characteristics within the dataset that required standard scaling, understanding that these features may have varied units or ranges, making them acceptable candidates for normalization.

We used the standard scaling method for each numerical characteristic individually. For each feature X in the dataset, we calculated standard scaling using the formula:

$$Z = \frac{(\mathrm{X} - \mu)}{\sigma}$$

Where is Z the standardized value, X is the original value, μ is the mean of the feature, and σ is the standard deviation of the feature.

We estimated the mean (μ) and standard deviation (σ) of each numerical characteristic in the dataset. These values were utilized in the usual scaling calculation.

We applied the conventional scaling formula to change the original values into standardized values for each occurrence and numerical characteristic in the dataset. This resulted in a new dataset where each feature had a mean of 0 and a standard deviation of 1.

We effortlessly merged the standardized numerical features into the dataset, replacing the original values. The dataset was now suitable for training our machine learning models.

Standard scaling was vital in our phishing attack detection technique, guaranteeing that numerical data with varied scales would not bias our machine learning model. By standardizing the characteristics, our algorithm efficiently learned patterns and correlations within the data, improving accuracy in identifying phishing URLs.

(vi) Stacking

In our phishing attack detection research, we utilized a strategic strategy to boost the symbolic strength of our dataset by merging textual information produced from TF-IDF (Term Frequency-Inverse Document Frequency) vectors with numerical characteristics. This merger was done by stacking the TF-IDF vector column with existing numerical columns, a process conducted using the vstack method. Below is a description of this Process for our dataset:

Stacking TF-IDF vectors with numerical columns was to produce a more complete and informative feature set for our machine learning models. By integrating textual and numerical information, we intended to capture a greater variety of features in our dataset that may help accurately identify phishing URLs.

The stacking method required mixing the TF-IDF vector column with the current numerical columns using the stacking technique. This Process vertically stacked the TF-IDF vectors on top of the feature scaled numerical columns, forming a unified dataset with textual and numerical representations.

The output of the stacking procedure was an integrated dataset where each instance preserved its numerical attributes alongside the newly included TF-IDF vectors. This merged dataset served as the input for later rounds of our investigation and machine learning model training.

The combined dataset, enhanced with both TF-IDF vectors and feature scaled numerical features, was employed as input for our machine learning models. This allowed the algorithms to exploit textual and quantitative information when learning patterns and correlations within the data.

By stacking TF-IDF vectors with feature scaled numerical columns using vstack, we effectively blended the strengths of textual and numerical representations, generating a more robust and informative dataset for our phishing attack detection system. This fusion of characteristics was crucial in increasing our machine learning models' overall performance and robustness against phishing URLs.

(vii) Feature selection

In our phishing attack detection study, feature selection played a vital role in increasing the efficiency and performance of our machine learning model. We applied the Random Forest algorithm as a feature selection tool to discover and prioritize the most significant attributes for differentiating between legal and fraudulent occurrences. Below is a complete description of how we accomplished feature selection using Random Forest in our dataset:

The primary purpose of our feature selection approach was to boost the performance of our phishing attack detection model by selecting a group of features that significantly contribute to correct categorization. Feature selection is intended to simplify the dataset, concentrating on the most important features while removing less relevant ones.

We selected the Random Forest algorithm as our feature selection approach due to its intrinsic capabilities to evaluate feature relevance. Random Forest generates an ensemble of decision trees, and the relevance scores assigned to variables by the ensemble provide significant insights into their effect on the model's predictive abilities.

After installing random forest, we gathered scores for all the features in our dataset. These scores measure the contribution of each feature to the prediction performance of the model, presenting a prioritized list of features depending on their influence.

We methodically found and exploited the most significant characteristics in our phishing attack detection model by employing Random Forest for feature selection. This method resulted in a more focused, interpretable, and efficient system, eventually boosting the model's capacity to categorize instances of phishing URLs reliably. The result of the feature important scores are stated in figure 3.4.4:

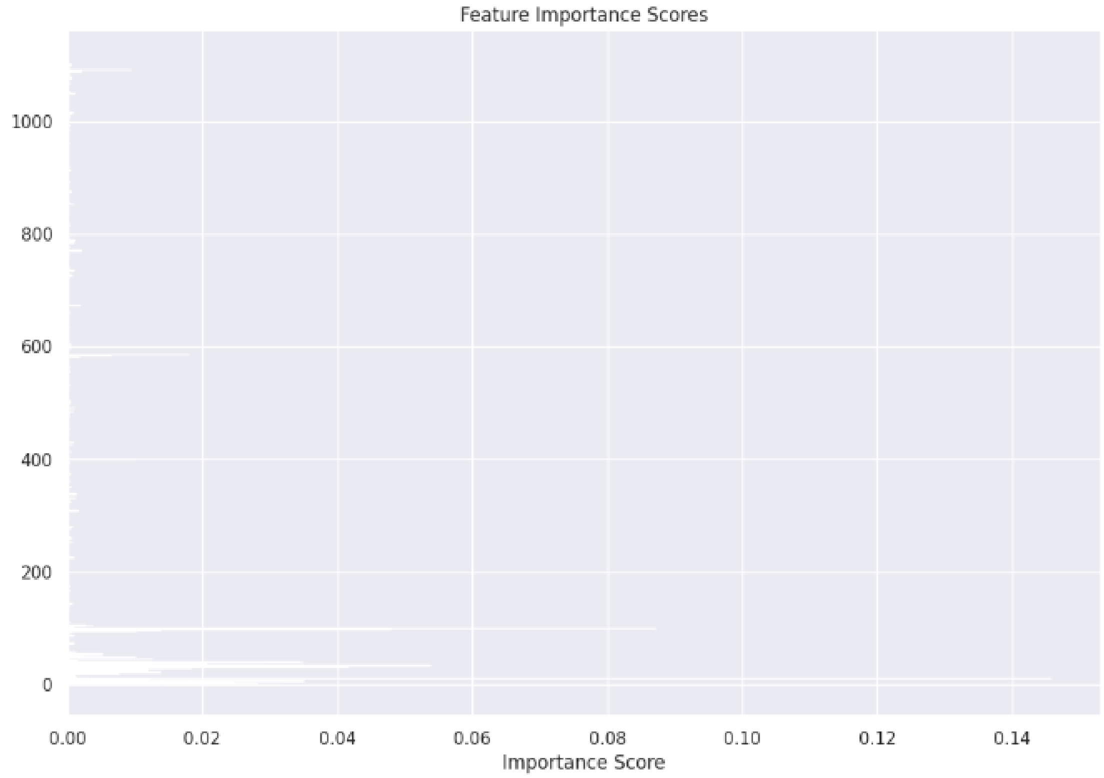


Figure 3.4.4: Feature importance scores

3.4.5 Data splitting

For our work on detecting phishing attacks, we followed a common approach in machine learning by dividing our dataset into separate training and test sets. This division, also known as an 80-20 split, entails assigning 80% of the data for training the machine learning model and setting aside the remaining 20% for assessing its performance. The justification and intricacies of this data partitioning approach may be elucidated in our study as follows:

The main objective of dividing the dataset is to evaluate the efficacy of our machine learning model on data that it has yet to be trained on. This exercise replicates real-life situations where the model meets unfamiliar examples and allows us to assess its ability to apply knowledge beyond the training data.

The training set included 80% of the dataset. This component is the foundation for training our machine learning model, allowing it to understand patterns, correlations, and decision limits from the supplied examples.

The test set comprised 20% of the dataset. This subset was left unaltered during the training phase and was exclusively set aside to assess the model's performance. It denotes a collection of examples that the model has yet to encounter.

Utilizing a distinct test set safeguards against overfitting, in which a model demonstrates good performance on the training data but has difficulties when presented with novel, unforeseen occurrences. By assessing the model on a unique test set, we acquire insights into its capacity to generalize and generate correct predictions on various data.

We applied randomization approaches to ensure the representativeness of the training and test sets. Random selection helps avoid biases if, for example, cases with specified features are clustered together in one collection.

The approach utilized to perform the data split comprises leveraging published libraries or methods inside machine learning frameworks, ensuring transparency and repeatability of our experimental setup.

By employing an 80-20 data splitting technique, we intended to strike a compromise between training our model on a suitably large dataset and evaluating its performance on a representative selection of unseen examples. This method allows us to assess our model's performance in identifying phishing assaults and contributes to the trustworthiness of our study findings.

3.4.6 Data validation

Data validation is a vital step in the preparation phase of a dataset for providing an ideal environment to run multiple models. It entails reviewing and guaranteeing the data's quality, correctness, and consistency to increase the dependability of future studies and model training. In our dataset, we applied five data validation approaches to ensure our data's quality, correctness, and consistency.

(i) Correlation matrix

While testing our dataset for phishing attack detection, we applied a fundamental analytical technique, the correlation matrix. This matrix briefly depicts the pairwise correlations between characteristics, delivering valuable insights into the interdependencies within the data.

The primary purpose of applying the correlation matrix was to uncover links and dependencies between distinct variables in our dataset. This investigation was essential in analyzing the possibility of multicollinearity, a vital aspect in assuring the resilience of our machine learning models.

We chose the Pearson correlation coefficient to generate the correlation matrix, given the nature of our characteristics and their distribution. This coefficient measures linear correlations between variables, providing a quantitative estimate of the degree and direction of these interactions.

Interpreting the correlation coefficients obtained from the matrix is critical for understanding the nature of feature connections. A coefficient near 1 implies a high positive relationship, whereas a number close to -1 denotes a strong negative correlation. Coefficients approaching 0 show little or no linear association between characteristics.

Identifying strong correlation coefficients between specific characteristics raised awareness about potential multicollinearity. We appreciate the impact multicollinearity might have on model stability and interpretability. Strategies were established to alleviate or reduce this problem, assuring the integrity of our following analyses.

To increase our comprehension and facilitate communication, we added visual aids such as heatmaps to show the correlation matrix. Figure 3.4.6 gives an accessible overview of correlation patterns within our dataset.

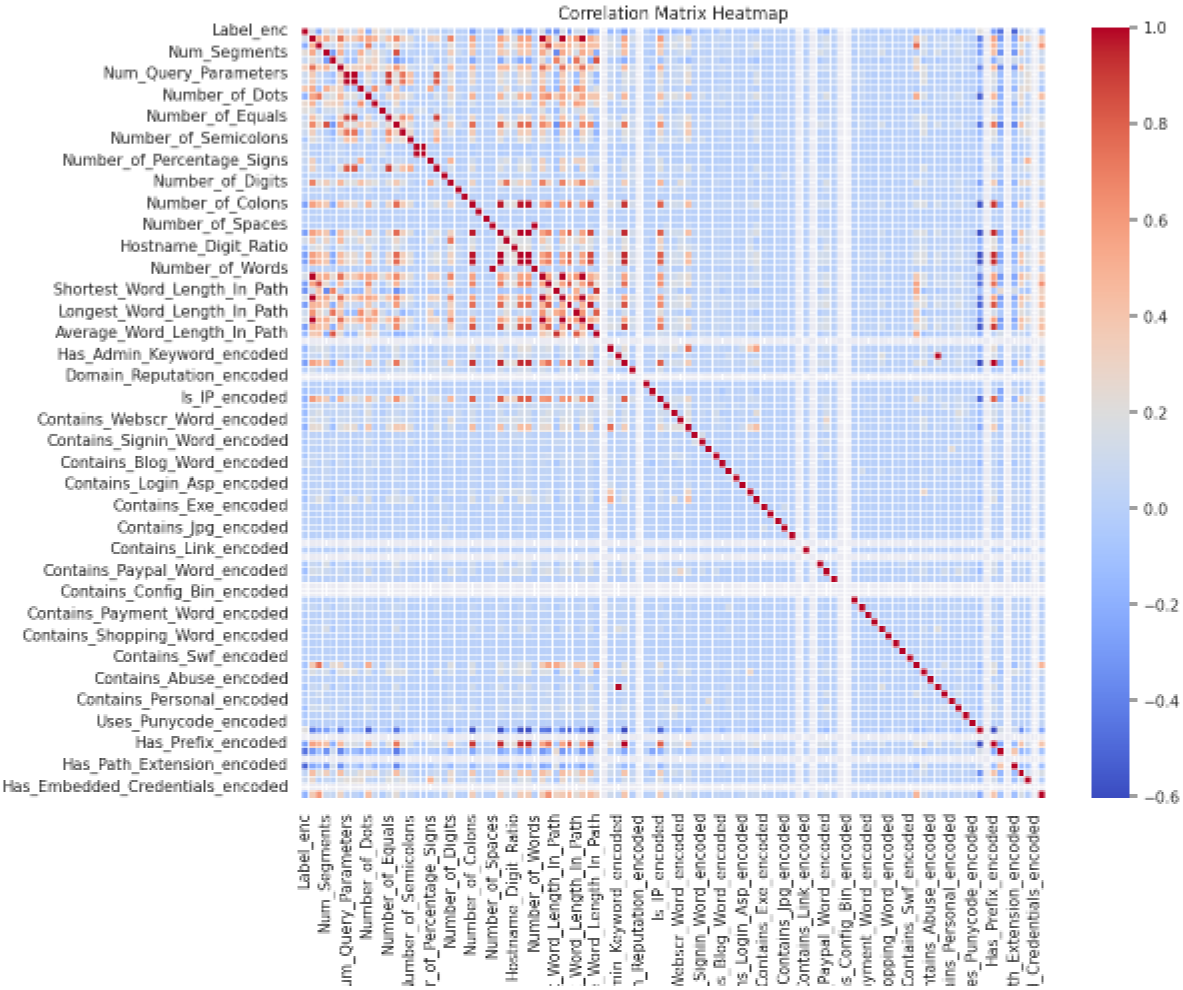


Figure 3.4.6: Correlation matrix

(ii) Null or Missing Values

In validating our dataset, fixing null or missing values surfaced as a vital step in guaranteeing the quality and completeness of our data. This part of data validation is vital to eliminate any biases and mistakes that might jeopardize the trustworthiness of our studies.

The primary purpose of resolving null or missing values was to increase the quality and consistency of our dataset. We intended to provide a stable basis for later machine learning model creation and analysis by carefully addressing missing data.

As our dataset was picked carefully, it did not include any missing or null values. We would have eliminated that entire row if any null or missing data existed.

(iii) Data Types

In the meticulous process of validating our dataset, a key facet involved addressing and validating the data types of our features. This step was undertaken to guarantee the consistency and appropriateness of data representations, thus fortifying the foundation for subsequent analyses and machine learning model development.

Our initial scrutiny involved a thorough examination of the data types assigned to each feature in the dataset. This encompassed understanding the nature of the information each variable was intended to capture and ensuring that the assigned data types align with the expected formats.

The primary objective of addressing data types was to ensure uniformity and accuracy in the representation of information across all features. This step is pivotal for preventing potential errors in computations, fostering interpretability, and facilitating seamless integration into machine learning models.

After all the pre-processing and post-processing, our dataset includes float values, which are excellent for running a model; we did not have to modify any data types subsequently.

(iv) Cross-Validation

As part of our extensive data validation efforts for phishing URL detection, we routinely employed cross-validation—a proven approach to check the efficacy of our machine learning model. The cross-validation scores give vital insights into the model's consistency and generalization across different subsets of the dataset.

We adopted a k-fold cross-validation approach, where the dataset was partitioned into k subsets (folds). The model was trained and validated k times, each time using a new fold for validation while the remaining folds were utilized for training. This approach was continued until each fold had served as the validation set precisely once.

The resultant cross-validation scores, notably [0.99923629, 0.99969452, 0.99954177, 0.99908355, 0.99908355], indicate the performance measures (such as accuracy, precision, recall, or F1-score) attained throughout each iteration of the cross-validation procedure. These scores are quantifiable assessments of the model's efficacy in generalizing to unseen data.

The extraordinarily high scores imply that our machine learning model consistently performed at a very high level across diverse subsets of the dataset. Each score is close to 1, showing high prediction powers and few mistakes in categorization.

The consistency of scores across folds shows that our model generalizes effectively to varied data sets. This resilient performance is critical for ensuring that the model is not overfitting to specific subsets but rather capturing patterns that reflect the entire dataset.

(v) Consistency Checks

In our thorough data validation procedure, we established a series of consistency checks to ensure the dependability and coherence of our dataset. The results of these checks, given in terms of mean, standard deviation, minimum, 25th percentile, median (50th percentile), 75th percentile, and maximum values, provide a thorough picture of the distribution and consistency of characteristics inside our dataset.

The mean values, standard deviations, and percentiles offered offer insights into our characteristics' central tendency, spread, and distribution. Let us interpret the results:

The mean values around zero suggest that, on average, the characteristics have a balanced distribution. Positive and negative means imply that values are scattered around the center.

The standard deviations represent the data's degree of dispersion or variability. Higher standard deviations reflect more scattered values, whereas lower values indicate a more concentrated distribution.

These percentiles give information on the data spread. For instance, the 25th percentile (Q1) and 75th percentile (Q3) assist indicate the interquartile range, showing where most data resides.

The minimum and maximum values show each characteristic's range of observed values. Understanding these extremes is critical for recognizing any outliers or abnormalities in the dataset.

3.5 Model implementation

The model training and deployment approach entailed feeding our rigorously preprocessed dataset with several models. Our study required the distinct installation of multiple machine learning models, each bringing a unique set of capabilities and traits to the task. The models considered for this study include a broad spectrum of approaches, allowing us to analyze and compare their unique performances. We employed a total of seven models to figure out the best potential result. The models utilized in our investigation include:

Random forest:

In our study on phishing attack detection, the Random Forest Classifier emerges as a vital component of our machine learning process. This ensemble learning method, noted for its adaptability and firm performance, plays a significant role in exploiting the collective intelligence of numerous decision trees. The Random Forest Classifier is particularly well-suited for binary classification tasks, such as discriminating between phishing and nonphishing cases within our dataset. At its heart, the Random Forest Classifier relies on the concepts of ensemble learning, where several decision trees are generated individually and combined to build a more robust and accurate prediction model. Each decision tree is trained on a random portion of the training data, and during the prediction phase, the various tree outputs are pooled to form a final, aggregated result.

In our phishing URL detection model, the Random Forest Classifier acts as a trustworthy and effective tool, utilizing the strength of ensemble learning to recognize subtle patterns indicative of phishing cases. Its durability, adaptability, and interpretability make it a significant asset in our search for precise and reliable phishing detection models, as it offers the highest result in our model.

XGBoost:

XGBoost, a gradient boosting method, was used independently to exploit its capacity to grasp complex patterns and correlations within the data. Known for its resilience and efficiency, XGBoost was applied to find subtle traits indicative of phishing assaults.

LightGBM:

LightGBM, a gradient-boosting framework designed for efficiency, was implemented as a standalone model. Its power to handle big datasets and high-dimensional feature spaces was utilized to discover patterns associated with phishing occurrences.

Neural Network:

A classic Neural Network, with its layered design capable of capturing non-linear interactions, was trained separately. The focus on deep learning methods helped us to find detailed patterns within the dataset linked with phishing attacks.

Logistic Regression:

A standard linear model was performed separately to offer an essential baseline. Its interpretability and simplicity assisted in understanding linear relationships within the data, establishing a standard for model comparison.

Decision Tree:

The Decision Tree paradigm, noted for its interpretability and transparent decision-making, was implemented in isolation. This model provides insights into feature relevance and explicit rules linked with phishing detection.

Deep Neural Network:

A Deep Neural Network, defined by its several hidden layers, was trained as a standalone model. This deep learning method allows for extracting hierarchical representations, reflecting intricate relationships within the dataset.

CHAPTER 4

Experimental Results and Discussion

4.1 Experimental Setup

The experimental setup in our paper on phishing attack detection using the Random Forest Classifier is a crucial component that outlines the procedures, tools, and configurations employed to conduct the experiments. The setup aims to ensure reproducibility, transparency, and a controlled environment for evaluating the model's performance. Here are key elements of the experimental setup:

Datasets Selection:

We selected diverse phishing datasets for training and testing the Random Forest Classifier. The choice of datasets considers different characteristics of phishing attacks, ensuring a comprehensive evaluation of the model's generalizability.

Data Preprocessing:

Data preprocessing steps are essential for preparing the datasets for model training. This involves handling missing values, addressing data imbalance, and performing any necessary transformations. The details of preprocessing steps, including tokenization and text cleaning, are explicitly described.

Feature Engineering:

Feature engineering involves selecting or creating relevant features for input to the Random Forest Classifier. This includes the extraction of meaningful information from raw data, such as URL structure, presence of certain keywords, and other relevant indicators of phishing attacks.

Model Selection:

The Random Forest Classifier is chosen as the primary machine learning model for phishing attack detection. The rationale behind this selection, including its versatility, ensemble nature, and suitability for the problem at hand, is thoroughly explained.

Hyperparameter Tuning:

Optimal hyperparameter tuning is crucial for the Random Forest Classifier's performance. The specific hyperparameters selected, such as the number of trees, maximum depth, and minimum samples split, are detailed. The tuning process may involve techniques like grid search or random search.

Training Procedure:

The training procedure outlines how the Random Forest Classifier is trained on the selected datasets. It includes the allocation of data for training and validation, the convergence criteria, and any specific considerations taken to enhance model convergence.

Validation and Testing:

The validation process is explained, including the use of validation sets to fine-tune the model during training. The testing procedure, using separate test datasets not seen during training, is detailed to assess the model's performance in real-world scenarios.

Performance Metrics:

The evaluation metrics used to assess the model's performance are clearly defined. This includes accuracy, precision, recall, F1 score, and potentially additional metrics relevant to phishing attack detection. The rationale for choosing these metrics is provided.

Comparative Analysis:

For comparative analysis with other classifiers, details on the selection of benchmark models, datasets used for comparison, and the statistical tests employed are explained. This ensures a fair and comprehensive assessment of the Random Forest Classifier's performance.

Tools and Frameworks:

The specific tools, libraries, and frameworks used for implementing the Random Forest Classifier and conducting experiments are mentioned. This ensures transparency and facilitates the reproducibility of the study.

By providing a detailed description of the experimental setup, our paper ensures that readers can replicate the experiments, understand the conditions under which the Random Forest Classifier was evaluated, and critically assess the validity and reliability of the study's findings.

4.1.1 Performance measurement matrices

In our study, a detailed evaluation of our deployed models was undertaken utilizing a complete set of performance measurement measures. These metrics give deep insights into our algorithms' efficiency in different categorization elements. The following section summarizes and interprets the critical performance measures considered in our investigation.

Accuracy:

Accuracy is a crucial measure measuring the overall accuracy of our models. It is determined as the ratio of successfully predicted cases to the total instances in the dataset. High accuracy reflects a model's ability to produce reliable predictions across positive and negative classifications.

Precision:

Precision evaluates the accuracy of positive predictions provided by our models. It is determined as the ratio of accurate positive predictions to the total positive predictions (true positives + false positives). High accuracy suggests a low rate of false positives.

Recall:

Recall, also known as sensitivity or true positive rate, assesses the capacity of our models to catch all relevant occurrences of the positive class. It is determined as the ratio of genuine positives to the total real positives (true positives + false negatives). High recall means a low rate of false negatives.

F1 Score:

The F1 score is the harmonic mean of accuracy and recall, evaluating a model's performance. It considers both false positives and negatives and is especially beneficial when there is an imbalance between the classes.

Confusion Matrix:

The confusion matrix is a tabular representation of the model's predictions, breaking down the number of true positives, true negatives, false positives, and false negatives. It gives a deep insight into the model's performance across multiple classes.

AUC Score (Area Under the Curve):

The AUC score is a statistic used to analyze the performance of a classification model at various threshold settings. It measures the area under the receiver operating characteristic (ROC) curve, offering insights into the model's capacity to distinguish across classes.

ROC Score (Receiver Operating Characteristic):

The ROC curve obtains the ROC score, indicating the trade-off between a true positive rate and a false positive rate at various threshold values. A higher ROC score shows enhanced discriminative capacity of the model.

Misclassification Error:

The misclassification error is the ratio of erroneously categorized occurrences to the total instances. It offers a direct indication of the model's total error rate.

Jaccard Score:

The Jaccard score analyzes the similarity between expected and actual sets, notably suited to cases where the intersection of true and predicted positives is critical.

4.2 Experimental Results & Analysis

In our study on phishing URL detection utilizing a broad group of seven machine learning models, the Random Forest Classifier emerged as the top-performing model based on a comprehensive examination of several performance indicators with an outstanding accuracy of 99.98%. Table 4.2 shows the outcome of all the models we have employed based on the performance measurement matrices:

Table 4.2: Performance Evaluation

No.	**Models**	**Accuracy**	**precision**	**Recall**	**F1 score**	**AUC score**	**M. Error**	**Jackard score**	**Confusion matrix**
1	Random forest	99.98%	100%	99.95%	99.97%	0.9997	0.0002	0.9995	[4267 0] [2 3915]
2	XGBoost	99.95%	99.92%	99.97%	99.95%	0.9995	0.0005	0.9990	[4264 3] [1 3916]
3	LightGBM	99.94%	99.92%	99.95%	99.94%	0.9994	0.0006	0.9987	[4264 3] [2 3915]
4	Neural Network	99.87%	99.90%	99.82%	99.86%	0.9986	0.0013	0.9972	[4263 4] [7 3910]
5	Logistic Regression	99.61%	99.59%	99.59%	99.59%	0.9996	0.0039	0.9919	[4251 16] [16 3901]
6	Decision Tree	99.77%	99.64%	99.87%	99.76%	0.9977	0.0023	0.9952	[4253 14] [5 3912]
7	Deep Neural Network	99.84%	99.85%	99.82%	99.83%	0.9996	0.0016	0.9967	[4261 6] [7 3910]

4.3 Discussion

The table displays the highest accuracy achieved by our model, which is 99.98%. The utilization of a random forest classifier obtained this accuracy. All the other accuracy results are likewise rather satisfactory, with the lowest being 99.61%. Random forest achieved a remarkable 100% accuracy, demonstrating its precision. While the XGBoost model has a recall rate of 99.97%, the Random Forest model outperforms all other models in several performance metrics. These include a 99.97% F1 score, a 0.9997 AUC score, a 0.0002 misclassification error, and a 0.9995 Jaccard score. While XGBoost has a false negative rate of 1, lower than the false negative rate of 2 for Random Forest, Random Forest outperforms all other models with a false positive rate of 0.

4.3.1 Visual representation

Confusion matrix:

Figure 4.3.1 showcases the result of confusion matrix of the Random forest classifier model:

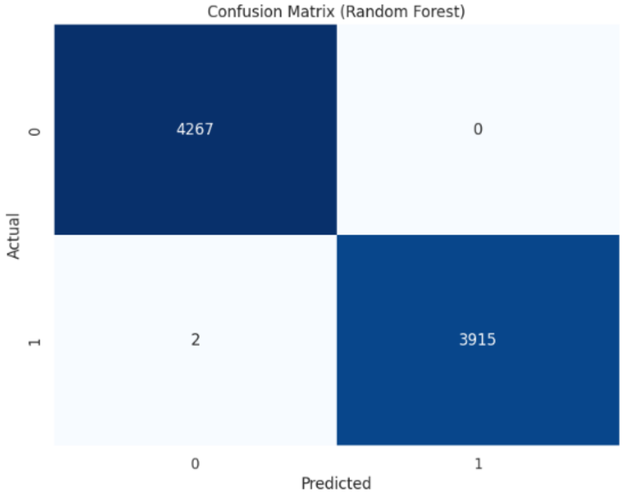


Figure 4.3.1: Confusion matrix of Random Forest classifier

ROC Curve:

Figure 4.3.2 showcases the result of ROC curve of the Random forest classifier model:

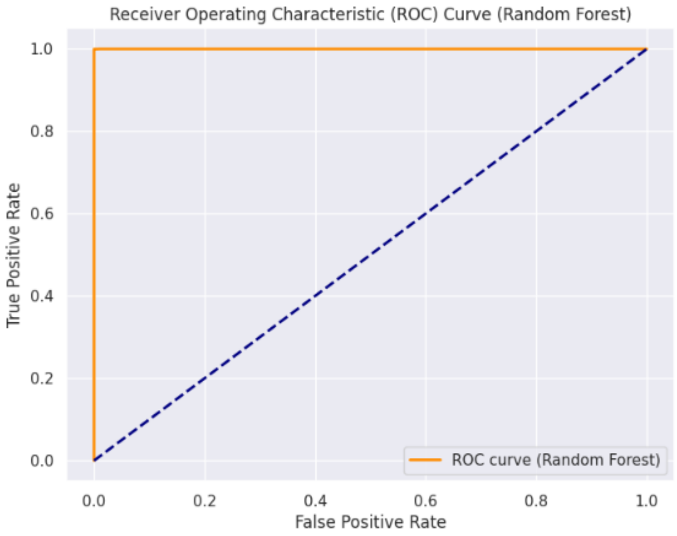


Figure 4.3.2: ROC Curve of Random Forest classifier

CHAPTER 5

Impact on Society, Environment and Sustainability

5.1 Impact on Society

Our study on phishing attack detection using the Random Forest Classifier can positively affect society by contributing to cybersecurity. The influence extends to numerous stakeholders and societal aspects:

The adoption of an efficient phishing attack detection system aids in enhancing overall cybersecurity defenses. The research contributes to building improved systems capable of recognizing and combating phishing attacks by employing machine learning, notably the Random Forest Classifier.

Phishing attacks generally target people seeking to acquire sensitive personal information. The research conclusions have a direct influence on keeping individuals from falling victim to such assaults and securing their personal and financial data.

Phishing attacks may lead to substantial financial losses for people, corporations, and organizations. As described in our article, the adoption of effective detection techniques aids in limiting these financial losses by preventing illegal access to financial information.

Successful phishing assaults diminish confidence in online communication and commerce. Our research contributes to sustaining online trust by minimizing the success rates of phishing attacks. This, in turn, generates a more secure and trustworthy online environment.

By adding user-centric assessment to the research, our work tackles the relevance of user awareness and experiences. Empowering people with knowledge and strengthening their awareness of phishing dangers can lead to a more cautious and resilient online community.

The research underlines ethical issues in implementing machine learning models for cybersecurity. By supporting responsible AI practices, it helps to the development of technologies that value privacy, fairness, and openness in their implementation.

The distribution of research findings adds to educational initiatives in cybersecurity. Academic institutions, researchers, and practitioners can exploit the findings to strengthen curricula, training programs, and professional development activities.

The research conclusions inform policy and regulatory issues relating to cybersecurity and the implementation of machine learning in sensitive sectors. Policymakers might consider the findings in creating legislation that balances security demands with ethical considerations.

Including machine learning, notably the Random Forest Classifier, for phishing detection adds to improvements in the technology sector. This may stimulate more developments and improvements in cybersecurity solutions across multiple sectors.

The research findings can motivate international collaboration among scholars, cybersecurity experts, and governments. This collaborative approach is vital for building common best practices, standardized standards, and a unified front against growing cyber threats.

As phishing attacks are minimized, customer confidence in online platforms is anticipated to rise. Users may feel more safe partaking in online activities, positively influencing e-commerce, online communication, and digital services.

Phishing attempts sometimes act as a prelude to identity theft. Effective detection techniques can play a crucial role in avoiding identity theft and safeguarding individuals from the terrible implications of stolen personal information.

In short, the societal effect of our study extends beyond the technical world, impacting how individuals, corporations, and politicians approach cybersecurity. By tackling the various issues posed by phishing assaults, the research contributes to developing a safer and more resilient digital environment for society.

5.2 Impact on Environment

While the primary focus of our research is cybersecurity and phishing attack detection, it is crucial to address any indirect implications on the environment. The environmental effect is not a direct product of the research, but it can be impacted by specific characteristics of the study and its implications:

Implementing machine learning models, like the Random Forest Classifier, may have ramifications for energy usage, especially if done on resource-intensive hardware. The training step of complicated models can be computationally intensive. However, the magnitude of this influence depends on the scale and infrastructure employed for deployment.

The choice of hardware infrastructure for implementing the machine learning models might influence the environmental effect. Energy-efficient hardware and cloud-based systems developed with sustainability in mind help offset possible detrimental impacts on the environment.

The scalability and efficiency of the adopted solution have a part in determining the environmental effect. A more scalable and efficient system consumes fewer computing resources, decreasing the overall environmental footprint associated with model training and deployment.

If the research involves exploiting cloud computing resources, the environmental effect is tied to the data centers' energy consumption. Cloud service providers implementing green computing methods can contribute to minimizing the carbon footprint connected with the research.

The dissemination of research findings generally entails preparing and distributing academic publications. Choosing electronic and paperless ways for disseminating research results aids in lowering the environmental impact involved with printing and sending physical copies.

Ethical issues in the research extend to environmental principles. Addressing environmental challenges in developing and deploying machine learning models corresponds with ethical values and sustainability aims.

Adopting green technology and sustainable practices in developing machine learning models can contribute favorably to the environmental effect. This involves using renewable energy sources and ecologically responsible computing methods.

It is vital to recognize that the environmental effect of our research is fundamentally tied to broader factors about technology, computer infrastructure, and research distribution methods. While the immediate impact on the environment may be minimal compared to disciplines with more direct environmental repercussions, implementing sustainable methods in the execution of the study corresponds with a more significant commitment to responsible and ethical conduct. Researchers may decrease the environmental impact by making mindful decisions in technology usage and dissemination techniques.

5.3 Ethical Aspects

Our study on phishing attack detection using the Random Forest Classifier covers ethical issues at several phases to ensure responsible and conscientious research. Ethical implications in the context of our study cover the following essential considerations:

We value the preservation of privacy throughout the research. This requires treating sensitive data responsibly, anonymizing information where appropriate, and ensuring that the model implementation does not violate the privacy of individuals.

Recognizing the possible flaws in machine learning models, we take methods to limit bias in the Random Forest Classifier. This requires careful selection and data preparation to prevent reinforcing existing biases and employing fairness-aware algorithms.

If the research incorporates user-centric assessment, gaining informed consent from participants is a crucial ethical practice. Participants are told about the nature of the study, any hazards, and how their data will be utilized. Consent is requested willingly and publicly.

Given the complexity of machine learning models, notably the Random Forest Classifier, we attempt to give straightforward model explanations. Ensuring that users, stakeholders, and the larger community understand how the model arrives at conclusions promotes openness and ethical accountability.

Ethical data handling methods are maintained throughout the research. This involves safe storage, limited access to sensitive information, and adherence to data protection standards. Data usage matches the intended purpose and is disclosed correctly.

The research focuses on user-centric assessment to understand user experiences and perceptions. This contributes to model improvement and empowers users by improving their knowledge of phishing hazards and enhancing their capacity to make educated decisions in online interactions.

A social impact assessment is undertaken to determine the potential repercussions of the study on persons and communities. Ethical issues extend beyond technical elements to incorporate the enormous societal ramifications of adopting phishing detection techniques.

Maintaining the integrity of the publication process is vital. The study is presented correctly, and outcomes are conveyed truthfully. The scientific community fully acknowledges any possible conflicts of interest or constraints.

Ethical considerations are not static. Continuous contemplation on ethical implications and the shifting environment of technology ensures that the study remains consistent with ethical ideals. Adjustments are made when needed to address developing ethical problems.

By including these ethical issues in our research, we want to contribute ethically to cybersecurity and machine learning. This strategy assures that developing and implementing the Random Forest Classifier for phishing attack detection matches ethical norms, creating trust and responsible innovation.

5.4 Sustainability Plan

A sustainability strategy for our study on phishing attack detection using the Random Forest Classifier contains measures to maintain the life, accessibility, and responsible effect of the research. The strategy incorporates different factors to enhance sustainability:

Ensuring open access to the work and accompanying resources: We seek to publish the paper in open-access journals or repositories to make it freely available to the scientific community, practitioners, and the public. Share datasets, code, and supplemental information on platforms that promote open access.

Enabling the reproducibility of the research: We intend to offer extensive documentation on datasets, techniques, and code implementation. This provides thorough directions, explanations, and version information to allow other researchers to replicate studies.

Ensuring the continuous functioning and relevance of the codebase: We want to utilize version control systems to manage and track changes in the code. Regularly update the codebase to fix errors, implement enhancements, and react to software dependencies or framework changes.

Upholding ethical standards in AI research and deployment: We seek to adhere to ethical norms, such as those defined by professional organizations or industry standards, and frequently evaluate and update the research processes to fit with emerging ethical issues in AI and machine learning.

Contributing to educational initiatives in the field: We aim to engage with educational institutions, workshops, and training programs and share insights from the research to contribute to the education and skill development of students, researchers, and professionals interested in cybersecurity and machine learning.

Minimizing the carbon footprint associated with research activities: We strive to examine environmentally sensitive methods in computing, such as adopting energy-efficient hardware, optimizing code for performance, and using cloud services with a dedication to sustainability.

Ensuring the availability of research outputs over the long term: We seek to deposit essential artifacts, including papers, datasets, and code, in trustworthy long-term repositories and adhere to archival best practices to maintain the integrity and accessibility of the research for future generations.

By adding these sustainability measures, our study intends to contribute to the present understanding of phishing attack detection and the long-term progress of ethical and responsible behaviors in cybersecurity and machine learning.

CHAPTER 6

Summary, Conclusion, Recommendation and Implication for Future Research

6.1 Summary of the Study

Our study delivers a detailed analysis of phishing attack detection employing the Random Forest Classifier. The study involves numerous facets, including technological approaches, ethical issues, and societal ramifications. Here is a concise description of the critical issues covered in the study:

Our purpose was to create and test a phishing assault detection system using machine learning with an emphasis on obtaining high accuracy, precision, and recall. The study incorporates machine learning-based techniques, notably leveraging the Random Forest Classifier. The technique covers data preprocessing, feature engineering, model training, and assessment of varied datasets. Rigorous preparation procedures, including tokenization, text cleaning, and standard scaling, are employed to assure the quality and relevance of the incoming data. Relevant attributes are picked or developed to boost the model's capabilities to detect phishing assaults. This entails thoroughly assessing elements such as URL structure and keyword presence. Various machine learning models are trained on selected datasets and rigorously assessed using performance measures, including accuracy, precision, recall, and F1 score, where the Random forest classifier got the best result of 99.98% accuracy. A comparative study with different models measures the classifier's efficacy. The paper finishes by reviewing significant findings, underlining the usefulness of the Random Forest Classifier in phishing detection, and reflecting on ethical and societal consequences. This work not only adds technical insights into phishing attack detection but also highlights ethical behavior, societal effects, and sustainability in the quickly developing environment of cybersecurity research.

6.2 Conclusions

To improve cybersecurity defenses against the ever-evolving environment of phishing assaults, our study has traversed the nuances of machine learning models for detection. Through a comprehensive comparative investigation, our technique, anchored by the Random Forest Classifier, has emerged as a beacon of excellence in the area. Our work provides numerous vital advances in the realm of phishing attack detection. The exhibited accuracy of 99.98% utilizing the Random Forest Classifier is a witness to our technique's resilience and usefulness. By exceeding or nearly rivaling the accuracies recorded in top-performing research, our approach places itself at the forefront of improvements in phishing detection.

The consequences of our results transcend the limits of academia, echoing the real-world issues created by harmful phishing attempts. As revealed by our research, a highly accurate detection model offers essential implications for bolstering corporate and individual cybersecurity frameworks.

This success is not a single undertaking but a product of coordinated efforts within the scholarly community. We applaud the many techniques studied by colleague academics, each bringing vital insights to the collective understanding of phishing detection methodologies.

As with any scientific attempt, our study is not without limits. Variations in datasets, feature engineering methodologies, and assessment measures among studies offer concerns that require continuous examination and improvement in future research.

In conclusion, our work emphasizes the efficacy of machine learning, particularly the Random Forest Classifier, in bolstering defenses against phishing attempts. As we reflect on the successes of our research, we acknowledge the dynamic nature of cybersecurity concerns. We are dedicated to extending the frontier of knowledge in pursuit of a more secure digital world. The path continues, and we add to the collective resistance against the ever-adapting menace of phishing attempts with each stride.

6.3 Implication for Further Study

While our research has made tremendous progress in enhancing the efficacy of phishing attack detection, the dynamic nature of cybersecurity necessitates continual innovation and study. Several pathways for future study arise from our findings, allowing the opportunity to increase the robustness and versatility of detection approaches.

Future studies might examine more detailed feature engineering approaches to catch minor distinctions in phishing assaults. Incorporating sophisticated feature extraction approaches like deep learning-based representations may boost the model's capacity to recognize complicated patterns.

Investigating the potential of ensemble techniques and integrating the capabilities of numerous classifiers has promise for further boosting detection accuracy. Ensemble approaches, when carefully combined, can provide a comprehensive and synergistic defense against multiple phishing tactics.

Phishing attackers are noted for their agility and creativity. Future work should incorporate thorough testing for adversarial resilience, ensuring the detection model remains successful even when faced with complex evasion strategies used by malevolent actors.

The development of phishing attacks underscores the significance of real-time detection and response. Future research might focus on constructing models capable of rapidly and precisely identifying phishing attacks, lowering the risk window for people and organizations.

As phishing strategies develop, so should the datasets used for training and testing. Continuous enrichment of datasets with the newest phishing samples and attack vectors will guarantee that detection models remain attentive to emerging threats and maintain their efficacy over time.

Investigating the generalization capabilities of detection models across multiple domains and sectors is vital. Future studies could examine techniques to construct models adaptive to multiple environments, accommodating the specific characteristics of distinct organizational ecosystems.

Recognizing end-user's relevance in the battle against phishing, future work should explore user-centric techniques. Understanding user habits, preferences, and decision-making processes can guide the creation of individualized and successful phishing protection methods.

As we look ahead, the environment of phishing attempts will continue to develop, needing a proactive and adaptable response. By entering these paths for future development, we contribute to the continued growth of cybersecurity techniques, striving towards a more secure digital environment for individuals and companies.

References

[1] https://aag-it.com/

[2] https://www.stationx.net/

[3] Jain, Ankit Kumar, and Brij B. Gupta. "A novel approach to protect against phishing attacks at client side using auto-updated white-list." EURASIP Journal on Information Security 2016 (2016): 1-11.

[4] Moghimi, Mahmood, and Ali Yazdian Varjani. "New rule-based phishing detection method." Expert systems with applications 53 (2016): 231-242.

[5] Medvet, Eric, Engin Kirda, and Christopher Kruegel. "Visual-similarity-based phishing detection." Proceedings of the 4th international conference on Security and privacy in communication netowrks. 2008.

[6] Hara, Masanori, Akira Yamada, and Yutaka Miyake. "Visual similarity-based phishing detection without victim site information." 2009 IEEE Symposium on Computational Intelligence in Cyber Security. IEEE, 2009.

[7] Wenyin, Liu, et al. "Detection of phishing webpages based on visual similarity." Special interest tracks and posters of the 14th international conference on World Wide Web. 2005.

[8] Al-Ahmadi, Saad. "A deep learning technique for web phishing detection combined URL features and visual similarity." International Journal of Computer Networks & Communications (IJCNC) Vol 12 (2020).

[9] Zhou, Yu, et al. "Visual similarity based anti-phishing with the combination of local and global features." 2014 IEEE 13th International Conference on Trust, Security and Privacy in Computing and Communications. IEEE, 2014.

[10] Buber, Ebubekir, Banu Diri, and Ozgur Koray Sahingoz. "NLP based phishing attack detection from URLs." Intelligent Systems Design and Applications: 17th International Conference on Intelligent Systems Design and Applications (ISDA 2017) held in Delhi, India, December 14-16, 2017. springer international Publishing, 2018.

[11] Wenyin, Liu, et al. "Phishing Web page detection." Eighth International Conference on Document Analysis and Recognition (ICDAR'05). IEEE, 2005.

[12] Basnet, Ram B., Andrew H. Sung, and Quingzhong Liu. "Rule-based phishing attack detection." International conference on security and management (SAM 2011), Las Vegas, NV. 2011.

[13] Liu, Gang, Bite Qiu, and Liu Wenyin. "Automatic detection of phishing target from phishing webpage." 2010 20th International Conference on Pattern Recognition. IEEE, 2010.

[14] Rajalingam, Mallikka, Saleh Ali Alomari, and Putra Sumari. "Prevention of phishing attacks based on discriminative key point features of webpages." International Journal of Computer Science and Security (IJCSS) 6.1 (2012): 1.

[15] Rao, Routhu Srinivasa, and Syed Taqi Ali. "Phishshield: a desktop application to detect phishing webpages through heuristic approach." Procedia Computer Science 54 (2015): 147-156.

[16] Mayuri, A., and M. Tech. "Phishing detection based on visual-similarity." International Journal of Scientific and Engineering Research (IJSER) 3.3 (2012): 1-5.

[17] Wardman, Brad, et al. "High-performance content-based phishing attack detection." 2011 eCrime Researchers Summit. IEEE, 2011.

[18] Barlow, Luke, et al. "A novel approach to detect phishing attacks using binary visualisation and machine learning." 2020 IEEE World Congress on Services (SERVICES). IEEE, 2020.

[19] https://www.alexa.com/topsites/

[20] https://www.phishtank.com/

[21] https://www.kaggle.com/

[22] Patil, Dharmaraj R., and Jayantro B. Patil. "Malicious URLs detection using decision tree classifiers and majority voting technique." Cybernetics and Information Technologies 18.1 (2018): 11-29.

[23] Zaiter, Ahmed Salama Abu, and Samy S. Abu-Naser. "Web page phishing detection Using Neural Network." (2023).

[24] IBRAHEEM, NUHA ABUBAKR ABDALRAHMAN. PHISHING WEBSITE DETECTION USING BAGGING ENSEMBLE MACHINE LEARNING. Diss. 2023.

[25] SRUTHI, K. "Real-Time Phishing Threat Detection using Lexical URL Features and Machine Learning Techniques." (2023).

[26] Rani, Liyana Mat, Cik Feresa Mohd Foozy, and Siti Noor Baini Mustafa. "Feature Selection to Enhance Phishing Website Detection Based On URL Using Machine Learning Techniques." Journal of Soft Computing and Data Mining 4.1 (2023): 30-41.

[27] Bhargava, Vaishali. "Leveraging Advanced Machine Learning Techniques for Phishing Website Detection."

[28] Kumari, Machikuri Santoshi, et al. "Viable Detection of URL Phishing using Machine Learning Approach." E3S Web of Conferences. Vol. 430. EDP Sciences, 2023.

[29] Kasim, Ömer. "Automatic detection of phishing pages with event-based request processing, deep-hybrid feature extraction and light gradient boosted machine model." Telecommunication Systems 78.1 (2021): 103-115.

[30] Gupta, Brij B., et al. "A novel approach for phishing URLs detection using lexical based machine learning in a real-time environment." Computer Communications 175 (2021): 47-57.

[31] Hannousse, Abdelhakim, and Salima Yahiouche. "Towards benchmark datasets for machine learning based website phishing detection: An experimental study." Engineering Applications of Artificial Intelligence 104 (2021): 104347.

[32] Moedjahedy, Jimmy, et al. "CCrFS: combine correlation features selection for detecting phishing websites using machine learning." Future Internet 14.8 (2022): 229.

[33] Raj, Mukta Mithra, and J. Angel Arul Jothi. "Hybrid Approach for Phishing Website Detection Using Classification Algorithms." ParadigmPlus 3.3 (2022): 16-29.

[34] Karim, Abdul, et al. "Phishing Detection System Through Hybrid Machine Learning Based on URL." IEEE Access 11 (2023): 36805-36822.

A novel Approach to Phishing Detection and Prevention using URL Features and Machine Learning Techniques

ORIGINALITY REPORT

23%	18%	14%	13%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	dspace.daffodilvarsity.edu.bd:8080 Internet Source	4%
2	Dharmaraj R. Patil, J. B. Patil. "Malicious URLs Detection Using Decision Tree Classifiers and Majority Voting Technique", Cybernetics and Information Technologies, 2018 Publication	2%
3	Submitted to The University of the West of Scotland Student Paper	1%
4	www.researchsquare.com Internet Source	1%
5	Submitted to Higher Education Commission Pakistan Student Paper	1%
6	Submitted to Daffodil International University Student Paper	1%
7	philpapers.org Internet Source	1%