National College of Ireland

Leveraging Machine learning to Enhance phishing URL detection.

MSc Research Project

Msc cyber security

Manpritcour Rathor

Student ID: x21215286

School of Computing

National College of Ireland

Supervisor: Vanessa Ayala-Rivera

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	ManpritcourRathor…… …
Student ID:	X21215286
Programme:	MSc Cyber Security **Year:** 2023.
Module:	MSCCYB1_Jan23A_I
Supervisor:	Vanessa Ayala-Rivera
Submission Due Date:	14/12/2024
Project Title:	Leveraging Machine learning to Enhance phishing URL detection
Word Count:8335 including appendices	**Page Count 26**

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Manpritcour Rathor

Date: 14/12/2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

ABSTRACT

This study adopts a supervised learning approach, concentrating on the identification of phishing websites through a diverse range of machine learning techniques. The process encompasses acquiring, processing, and visualizing a comprehensive dataset containing 1000 URLs for each category (phishing and legitimate). Utilizing Python libraries like urllib and whois, 19 features including URL length and domain age are extracted. The dataset undergoes preprocessing, addressing null values, and transforming categorical data into a numerical format. Four machine learning models—Logistic Regression, AdaBoost Classifier, Gradient Boosting Classifier, and Stacking Classifier—are both trained and assessed using metrics like accuracy, precision, recall, and F1-score. The ISCX-URL2016 dataset, encompassing 45,225 URLs, from which we used 1000 URLS for Phishing and 1000 for legitimate which ensures that the model is trained on a vast and varied dataset,. Preprocessing involves managing null values, converting data to NumPy arrays, and employing correlation-based feature selection. The proposed phishing detection system encompasses webpage generation, feature extraction, and the training of machine learning models, with a 90:10 split for training and testing. Results highlight varying model performances, with the Stacking Classifier demonstrating notable accuracy and balance. Individual model experiments provide in-depth insights into their specific strengths and weaknesses.

Keywords**:** phishing attacks, machine learning, cybersecurity, dataset (ISCX-URL2016), URL detection system, feature extraction, logistic regression, gradient boosting classifier, stacking classifier, adaboost classifier, Stacking Classifier, anti-phishing solutions

1. Introduction

Nowadays, since communicating digitally is commonplace, cybercrime particularly phishing attacks poses a threat to the security of personal and corporate data. A sort of cybercrime known as "phishing" occurs when offenders purposefully deceive targets into disclosing personal data, including usernames and passwords, or even crucial financial details. The rising reliance of people on digital transactions and the popularity of online commerce are the two main targets of these fraudulent crimes.

As individuals increasingly conduct transactions, pay bills, and transfer money online, the importance of identifying and countering phishing websites becomes paramount. Reports from the Anti-Phishing Working Group reveal a staggering 647,592 unique phishing sites reported until September 2018, underscoring the scale and pervasiveness of this cybersecurity threat. The deceptive nature of phishing, often camouflaging itself as a trustworthy entity through the creation of fake websites with HTTPS certification, makes it a formidable challenge for users to distinguish between genuine and malicious platforms.

This master's thesis report is dedicated to distinguish the complexities of phishing attacks and proposes an innovative solution through the integration of machine learning principles into cybersecurity measures. Traditional cybersecurity defenses, although effective against certain threats, often fall short in addressing the nuanced and socially engineered nature of phishing attacks. With a particular focus on countering phishing, this research explores four distinct approaches: Rule-based or Heuristics-based, Blacklisting, Content-based, Machine Learning-based, and a Hybrid approach.(Almousa et al., 2022)

In this article, related work Section 3 describes deep learning models used for Machine Learning, a relatively recent and innovative approach in mitigating phishing attacks. It leverages algorithms and data analysis to detect and prevent phishing attempts, offering the potential to significantly reduce this threat. This research is centered around the application of Machine Learning principles in countering phishing attacks, The research methodology is described in Section 4. Section 5 describes the design components of our machine learning algorithms framework. The implementation of this study is described in Section 5. Section 6 presents and discusses the evaluation results. Section 7 concludes the study and discusses future work.

a. Research Question

"How can the precision of identifying spoofed websites in phishing attacks be enhanced through the integration of diverse supervised learning techniques, advanced feature extraction, and dynamic adaptation to emerging threats?"

2. Related Work

3.1 Random Forest and Support Vector Machine Phishing Website Detection

The study leverages a dataset sourced from the Faculty of Computer Science and Information Technology (FSIT) at the University Malaysia Sarawak (UNIMAS), comprising 30,000 websites, evenly divided into phishing and legitimate categories. Focused on raw HTML code, the dataset includes additional files such as SCREEN-SHOT,

URL, WEBPAGE, and WHOIS. The research methodology integrates tokenization through Byte Pair Encoding (BPE) and Term Frequency-Inverse Document Frequency (TFIDF) scoring for feature extraction from HTML files. Classification employs Random Forest (RF) and Support Vector Machine (SVM) algorithms, with a 70:30 training and testing dataset split. The model's workflow involves user input of a website URL, feature extraction, model training and testing, culminating in a determination of the website's legitimacy. The scikit-learn tools facilitate model training and performance evaluation. Results demonstrate the superiority of Random Forest over SVM, showcasing its efficiency in phishing website detection with a 99.98 percent accuracy. This finding is reinforced by a comprehensive analysis of performance metrics, including precision, recall, F1-score, and the confusion matrix, positioning Random Forest as the preferred model due to its accuracy and processing efficiency. (Almousa et al., 2022)

3.2 Advancements in Phishing Website Detection with PHISHWEB

PHISHWEB is a sophisticated phishing detection system designed with a multi-filter approach, integrating blocklists, allowlists, Similar Domain Detection (SDD), and Domain Generation Algorithm (DGA) detection. The SDD filter focuses on identifying forged domains through homoglyph and typosquatting analysis, utilizing a list of targeted domains and employing similarity metrics. The DGA filter, inspired by reputation value computation, assesses the likelihood of a domain being benign based on n-grams. An ML-driven extension enhances DGA detection by combining reputation value, domain length, and character randomness features through a random forest classifier. Evaluation results showcase the system's efficacy in detecting phishing domains, demonstrating high precision and recall. The study also extends its application to real-world DNS measurements, confirming the practicality of the proposed solution. Overall, PHISHWEB offers a promising avenue for accurate phishing detection without relying on website content features.(Aravena et al., 2023a)

3.3 Detection of Phishing Websites and Spam Content Utilizing Machine Learning Algorithms

The literature emphasizes the pivotal role of algorithms in cybersecurity, particularly in detecting phishing websites and identifying spam content. The Support Vector Machine (SVM) is highlighted for its effectiveness in classification tasks, excelling in scenarios where linear separation is challenging. The incorporation of Natural Language Processing (NLP) enhances the system's text analysis capabilities for spam detection. The proposed methodology includes user registration, login, and phishing website detection using SVM, with NLP contributing to spam content analysis. Comparative analyses favor SVM over Random Forest (RF) due to its superior accuracy (88%). SVM's ability to construct decision boundaries in complex, multidimensional space underscores its efficacy in cybersecurity applications, ensuring robust detection outcomes.(*(PDF) Phishing Website and Spam Content Detection Using Machine Learning Algorithms*, n.d.)

1.4 A methodology for detecting phishing websites, utilizing a Multi-layer Perceptron and employing Mutual Information Feature Selection.

The presented methodology relies on the UCI dataset, encompassing 1353 websites categorized into regular, phishing, and suspicious labels. The dataset undergoes a 7:3 training-test split. Mutual Information Feature Selection is employed for its relevance to label and low redundancy with other features. The algorithm sequentially calculates mutual information, sorts features, and generates a new dataset, D'. The overall process involves feature selection and model training using a multi-layer perceptron (MLP) with a three-layer structure. The algorithm's pseudo-code outlines steps for feature selection and MLP model training. Evaluation indicators include accuracy, precision, recall, and F1 score. Comparative analysis demonstrates the method's superior performance in phishing website detection, outperforming other approaches in accuracy, recall, and F1 score.(Yang et al., 2022a)

1.5 PhiKitA introduces a dataset, specifically designed for identifying phishing websites through phishing kit attacks.

The literature review explores two main categories related to phishing kits: understanding their behavior and using their analysis for phishing identification. Studies include Cova's analysis of phishing kits' stolen information destinations and Oest et al.'s examination of anti-phishing groups' blocklists.

In the second category, Britt et al. proposed a method using MD5 values for similarity, creating brand-consistent clusters. Orunsolu and Sodiya achieved 85% accuracy with a Naive Bayes classifier using phishing kit features. Tanaka et al. used website structure signatures, and Bijmans et al. proposed a fingerprint representation.

Feng et al. employed web structure analysis for phishing identification. Various data collection methods, such as distribution sites and server honeypots, are discussed in the literature. The paper introduces the PhiKitA dataset, created for phishing website identification through phishing kit attacks, and the experiments conducted contribute to advancing research in phishing detection methodologies.(Castano et al., 2023)

1.6 Detection of Phishing in URLs through Machine Learning Techniques Utilizing Lexical Analysis

In the fight against phishing attacks, researchers have developed various anti-phishing techniques to achieve high-accuracy detection while reducing false positives. A review of these methods reveals significant advancements. For instance, one study achieved an 83% accuracy rate using J48, SVM, and Logistic Regression classifiers with a substantial dataset for real-time URL detection. Another middleware system achieved an impressive 86% accuracy rate employing Random Forest, SVM, and KNN algorithms on a carefully curated dataset. Another approach using a Random Forest-based strategy achieved an 87% accuracy with a select set of URL features. Content analysis and URL feature extraction led to an 86.01% accuracy using Artificial Neural Network. While Random Forest performed well with an 86.9% accuracy in a specific study, an innovative Extreme Learning Machine (ELM) based on the Random Forest algorithm reached 85.34% accuracy. The "PhishStorm" system detected phishing URLs with an accuracy rate of 84.91%. Collectively, these studies highlight the effectiveness of machine learning approaches in analyzing URL lexical features for robust phishing detection, potentially surpassing previous methods and laying the groundwork for further research in developing more potent anti-phishing solutions.(Abutaha et al., 2021a)

3.7 A systematic exploration of phishing detection methods coupled with a structured approach to develop an anti-phishing framework. The literature emphasizes the significance of website features in phishing detection, categorizing them into HTML and JavaScript-based, Address bar-based, Abnormal-based, and Domain-based features. The study modifies and refines existing feature definitions, discarding outdated criteria like the use of '@' symbol, right-click disablement, and hiding suspicious links. The URL length criterion is adjusted for improved accuracy. The architecture involves URL and feature collection, feature selection, and classification, utilizing machine learning algorithms such as Naive Bayes, SVM, AdaBoost, Random Forest, JRIP, PART, PRISM, C4.5, and CBA. The observation recommends heuristic and hybrid approaches for superior accuracy, while acknowledging challenges like the blacklisting technique's limitations with newly registered domains. The study underscores the importance of diverse datasets, feature selection, and continuous adaptation to evolving phishing techniques.(Patil & Dhage, 2019a)

3.8. A machine learning technique utilizing Support Vector Machines to identify phishing websites. The proposed methodology outlines a systematic process for phishing detection using a Support Vector Machine (SVM) algorithm. The key steps involve collecting a phishing dataset, implementing the SVM algorithm in Python, and enhancing the prediction model's performance. The methodology includes data selection, pre-processing, dataset splitting, feature extraction, classification using SVM, prediction, and evaluation. Evaluation metrics such as accuracy, precision, recall, and classification error are employed. The study aims to forecast phishing websites by refining prediction performance through SVM. The algorithm involves filtering null values, sorting data, and selecting the best features for classification. The evaluation phase assesses accuracy, precision, recall, and F1-score, with results generating metrics like error rates. The comprehensive process seeks to provide an effective and accurate approach to phishing website detection.(Jain & Gupta, 2023)

3.9. Identifying Phishing Websites Using Domain and Content Analysis

The literature review covers phishing detection techniques, including domain blacklisting, NLP algorithms, and Machine Learning. URL-based detection targets typosquatting and subdomains mimicking legitimate services, while content-based detection focuses on common elements. Challenges include evolving phishing techniques and frequent model retraining.

The proposed solution introduces URL and content analysis, emphasizing simplicity and flexibility. Python scripts perform feature extraction and comparison offline. Testing reveals high accuracy against known malicious domains, with limitations in detecting certain attack vectors. Content-based analysis using offensive tools demonstrates high title similarity and authentication keyword identification. (Pascariu & Bacivarov, 2021a)

3.10 Detecting Phishing Websites through Machine Learning

The literature review covers diverse approaches to phishing website detection using machine learning. Studies explore semantic data as a social engineering indicator, URL identification with Random Forest (75% accuracy), and flexible feature extraction with neural networks (84.18% accuracy). Comparative analyses highlight the success of deep learning techniques like CNN-LSTM (78% accuracy) and reduced feature selection with SVM outperforming Logistic Regression. Other studies focus on phishing detection in Chinese web pages, efficient C4.5 decision tree-based URL detection (89.40% accuracy), and heuristic-based methods achieving accuracies of 87% and 84.91%. The review also introduces models like PhishChecker for domain-based phishing URL detection (86% accuracy). Overall, these studies showcase machine learning's effectiveness across diverse phishing detection scenarios and datasets.The Table 1 shows the research niche

Research Niche

Index	Research Papers	Authors and date	Strengths	Limitations
1	Phishing Website Detection Using Random Forest and SVM: A Comparison	(Noh & Nazmi Bin M Basri, 2021)	The paper enhances accuracy using a 500 site UNIMAS dataset. BPE and TFIDF improve performance, while RF achieves 87.98% accuracy. A 70:30 split boosts generalization. These strengths underscore the study's effective methodology.	The study's limitations include potential struggles with dynamic content due to reliance on raw HTML. Generalizability beyond the dataset is discussed minimally, and there's a lack of details on model interpretability and computational resources for training, limiting scalability.
2	Phish Me If You Can – Lexicographic Analysis and Machine Learning for Phishing Websites Detection with PHISHWEB	(Aravena et al., 2023b)	Aravena et al.'s (2023) "Advancements in Phishing Website Detection with PHISHWEB" showcases a precise multi-filter system, accurate SDD, DGA assessment, and ML-driven extension. Results reveal high precision and recall, extending to real-world DNS measurements for accurate phishing detection without content reliance.	PHISHWEB lacks detailed computational info for real-world use. Adaptability insights are limited. ML-driven extension specifics are missing, potential vulnerabilities unexplored. Real-world DNS measurement details are absent. Ethical considerations and biases in training data aren't addressed, impacting reliability. Yet, the study provides valuable insights into advanced phishing detection.
3	Detecting Phishing Websites through Machine Learning	(Kiruthiga & Akila, 2019)	The literature review on phishing detection using machine learning emphasizes strengths such as leveraging semantic data, high accuracy with methods like Random Forest (85%) and neural networks (84.18%), and success in diverse scenarios like phishing detection in Chinese web pages (89.40% accuracy).	The literature review on machine learning for phishing detection highlights promising results but lacks in-depth discussions on interpretability, ethics, and generalizability. It could be strengthened by addressing these aspects and exploring benefits from combining detection approaches.
4	Detection of Phishing Websites and Spam Content Utilizing Machine Learning Algorithms	(Kiruthiga & Akila, 2019)	The literature emphasizes SVM's effectiveness in complex cybersecurity classification, especially with challenging linear separations, and its superiority over RF (88% accuracy). Integrated NLP enhances spam detection in the proposed methodology.	The literature emphasizes algorithms in cybersecurity, favoring SVM for complex classifications and NLP for spam detection. The proposed methodology includes user registration, login, and phishing detection using SVM and NLP. SVM outperforms RF with 88%

				accuracy, but limitations include interpretability concerns, biases, and scalability issues.
5	Detection of Phishing in URLs through Machine Learning Techniques Utilizing Lexical Analysis	(Abutaha et al., 2021b)	The study excels in data handling, feature extraction, and classification using versatile algorithms like Gradient Boosting, SVM, Random Forest, and Neural Network. Robust evaluation metrics ensure accurate classification, and high-performance computing enhances efficiency.	The study lacks a literature review, and while its methodology is systematic, details on performance metrics are limited. Addressing imbalanced classes may introduce biases, and feature selection could impact predictive power. External factors influencing phishing detection are not explored.
6	A methodology for detecting phishing websites, utilizing a Multi-layer Perceptron and employing Mutual Information Feature Selection	(Yang et al., 2022b)	The study optimizes phishing detection with a 1353-website dataset, emphasizing Mutual Information Feature Selection and a three-layer MLP. Achieving superior performance in key metrics, it showcases robust dataset handling and feature selection.	Phishing detection has UCI dataset biases and limited scalability. Fixed split ratios and feature selection sensitivity impact real-world use. Generalizing findings to diverse datasets needs further exploration.
7	A systematic exploration of phishing detection methods coupled with a structured approach to develop an anti-phishing framework	(Patil & Dhage, 2019b)	The study adeptly refines phishing detection features, demonstrating versatility in machine learning algorithms. It emphasizes the importance of diverse datasets, feature selection, and ongoing adaptation to evolving phishing techniques.	The study faces challenges with blacklisting for new domains and potential misclassification in heuristics due to predefined thresholds. It stresses diverse datasets and adapting to evolving phishing techniques.
8	A machine learning technique utilizing Support Vector Machines to identify phishing websites.	(Abdulwakil et al., 2017)	The SVM-based phishing detection method excels with Kaggle datasets and Python (Spyder 2.7), emphasizing precision, recall, F1-score, and error rates for a robust evaluation. Its strength lies in optimized feature extraction for enhanced accuracy and reliability.	The SVM-based phishing detection approach faces limitations in dataset sensitivity, potential feature inadequacy, adaptability concerns to evolving tactics, reliance on traditional metrics, and computational challenges, highlighting the need for a nuanced evaluation.
9	PhiKitA introduces a dataset, specifically designed for identifying phishing websites	Castaño, Felipe and Fernañdez, Eduardo Fidalgo and Alaiz-Rodríguez, Rocío and Alegre,	The review covers phishing kit studies, emphasizing MD5 and Naive Bayes for identification, culminating in the PhiKitA dataset, providing a comprehensive overview of phishing detection approaches.	Limitations involve dataset specificity, metric inconsistencies, and potential oversight of emerging techniques. Scalability, temporal validity, biases, and ethics also need consideration for robust phishing detection methodologies.

	through phishing kit attacks	Enrique.(*IEEE Xplore Full-Text PDF:*, n.d.)		
10	Identifying Phishing Websites Using Domain and Content Analysis	(Pascariu & Bacivarov, 2021b)	The review and solution excel in phishing detection, emphasizing URL and content analysis. Strengths include simplicity, flexibility, and high accuracy against known malicious domains. It offers an efficient, adaptable approach for automated and manual analysis.	The solution effectively detects known malicious domains but may struggle with evolving phishing tactics, potential oversights in specific attack vectors, and title discrepancies based on location. Regular model retraining is needed, and accuracy may be compromised in URL shortener scenarios.

Table 1

Literature Review Conclusion:

In our study on spotting phishing websites, we have noticed the use of various machine learning (ML) tools. Interestingly, our research focuses on finding unique combinations of ML algorithms, including ensemble methods, often called hybrid algorithms. We are keen to see how well we can combat phishing by teaming up machine learning with heuristic approaches. Our main aim is to check how effective these hybrid algorithms can be in identifying phishing attempts. This research aims to shed light on the world of phishing detection, emphasizing the collaboration between machine learning and heuristic methods.

3. Research Methodology and Design Specification

In my research, I employ a supervised learning approach, where the algorithm is informed about the expected outcomes. The model is trained to make accurate predictions by constructing a mathematical model based on provided input data. This type of learning, encompassing both input and output data in a dataset, is known as Supervised Learning. For instance, in determining the legitimacy of an email (a binary 0 or 1 scenario), this falls under supervised learning.

In a broader context, Supervised Learning includes Classification and Regression algorithms. Classification algorithms are utilized when the output is limited within a specific range, such as identifying whether an email is phishing. On the other hand, Regression algorithms are applied when the output involves continuous changes within a defined range, such as predicting temperature fluctuations.(*What Is Machine Learning? Definition, Types, Tools & More | DataCamp*, n.d.)

"The suggested method aims to enhance the precision of identifying spoofed websites by employing diverse supervised learning techniques. The methodology employed in this study involves a systematic approach to handling and analysing the dataset (containing 1000 URLs for each class, distinguishing between legitimate and phishing websites. Through Python libraries like urllib and whois, 19 features are extracted, encompassing characteristics such as URL length, domain age, and more. Labels are assigned to indicate the target class of each URL, and the resulting dataset is saved into a CSV file for subsequent analysis. The essential Python libraries are imported and installed to ensure the availability of the required functionalities. The Pandas library is utilized to load the dataset, followed by a thorough data cleaning process that addresses null values and removes unnecessary columns. Further preprocessing involves converting categorical data into a numerical format suitable for machine learning algorithms. The dataset is split into training and testing sets with a 90:10 ratio. The machine learning models selected for training include Logistic Regression, AdaBoost Classifier, Gradient Boosting Classifier, and a Stacking Classifier combining Gradient Boosting and AdaBoost with a meta-classifier Logistic Regression. The final step involves evaluating the model performance using standard metrics such as confusion matrix and classification report to provide a comprehensive understanding of the effectiveness of the models in detecting phishing websites. The figure provided outlines the procedural steps of the methodology.

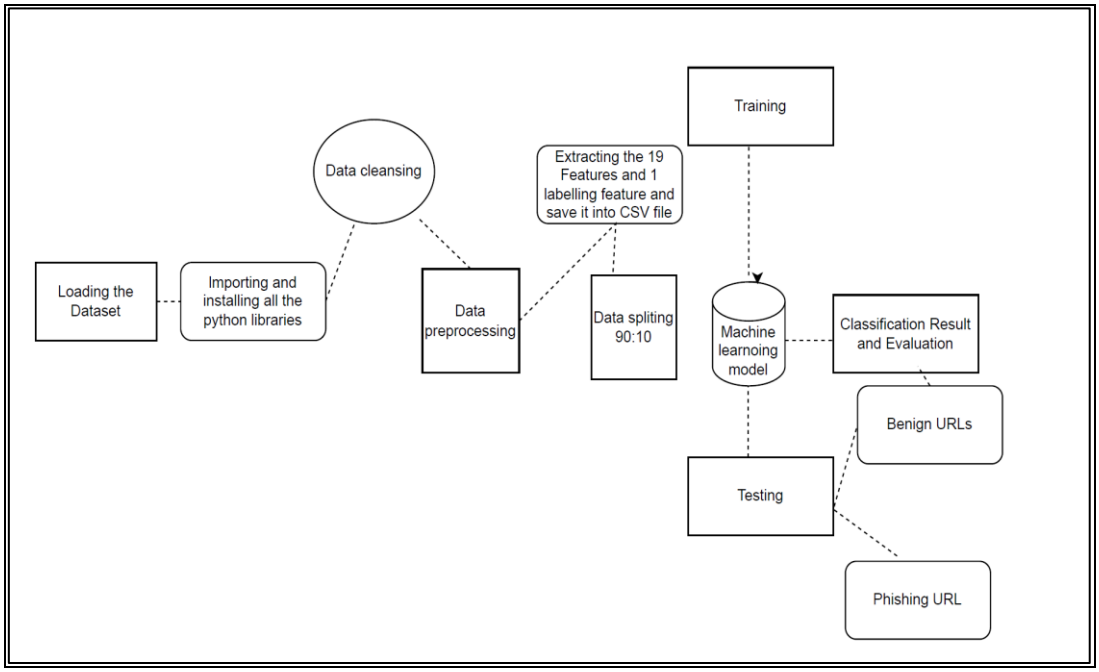


Figure 1 Process flow of proposed Model

Below is the process flow of the project:

Step 1: **Dataset acquisition**
Load a dataset containing 1000 URLs from each class, ensuring a balanced representation.
Step 2: **Data Processing and Visualization**
The extracted features are saved into a CSV file for easy accessibility. The dataset is loaded using pandas, and data cleaning involves handling null values and removing unnecessary columns. Categorical data is converted. into numerical format for effective machine learning model training.
Step 3: Data Splitting
The dataset undergoes a division into training and testing sets, adhering to a ratio of 90:10. This specific split ratio was chosen after experimenting with various alternatives, and it consistently yielded the highest accuracy. The rationale behind selecting the 90:10 split is to ensure a robust evaluation of the model's performance. This ratio is opted for as it allows the model to extensively learn from a significant portion of the dataset, thereby effectively capturing the majority of phishing patterns during the training process.

Step 4: Machine Learning Model Training
Four machine learning models are employed for training
Logistic Regression, AdaBoost Classifier, Gradient Boosting Classifier, Stacking Classifier (combination of Gradient Boosting and AdaBoost, with a meta-classifier Logistic Regression)
The Machine Learning Algorithms used in developing the models are:

Algorithms

ADABOOST:
AdaBoost, or Adaptive Boosting, stands as an ensemble machine learning algorithm applicable across diverse classification and regression tasks. Functioning as a supervised learning method, AdaBoost classifies data by amalgamating numerous weak or base learners, such as decision trees, into a potent learner. The algorithm operates by assigning weights to instances in the training dataset, influenced by the accuracy of prior classifications.(*Understanding the AdaBoost Algorithm | Built In*, n.d.)

GRADIENT BOOSTING
Gradient boosting, categorized under machine learning boosting, operates on the principle that the optimal subsequent model, when integrated with preceding models, works towards minimizing the overall prediction error.

The fundamental concept involves establishing the target outcomes for this subsequent model with the aim of reducing the error to its minimum.(*What Is Gradient Boosting? - Gradient Boosting Explained - Displayr*, n.d.)

LOGISTIC REGRESSION

Logistic regression serves as a classification algorithm employed for categorizing observations into distinct classes. Instances of classification problems include determining if an email is spam or not, identifying online transaction fraud, or classifying tumors as malignant or benign. Logistic regression utilizes the logistic sigmoid function to transform its output, providing a probability value for the classification task at hand.(*An Introduction to Logistic Regression in Python*, n.d.)

STACKING CLASSIFIER

The stacking Classifier represents an ensemble technique that merges predictions from multiple models to enhance overall performance. In this specific case, the stacking classifier amalgamates predictions from both Gradient (*How Stacking Technique Boosts Machine Learning Model's Performance - Dataaspirant*, n.d.)Boosting and AdaBoost classifiers. What distinguishes stacking is the inclusion of a meta-classifier, in this instance, Logistic Regression. The meta-classifier is trained on the outputs of the underlying models, synthesizing their predictions to provide a refined and informed final prediction. This approach leverages the strengths of each base model, resulting in a more robust and accurate classification.

Step 5: In this step, we are going to evaluate all the four models performance using the following metrics the detailed explanation is given in Evaluation

a. Confusion Matrix
b. Classification Report

4. Implementation

In this thesis, the acquisition and utilization of a comprehensive URL dataset (ISCX-URL2016) play a pivotal role in understanding and preventing online criminal activities, particularly in the realm of phishing. The URL dataset (ISCX-URL2016) employed encompasses 45,225 instances, distinguished into two versions, with 35,260 instances classified as legitimate and 9,965 instances labeled as phishing websites. Following the removal of the target phishing property, 19 features are utilized to differentiate instances, assigning a value of 1 for phishing and 0 for legitimacy. The significance of this dataset stage lies in its contribution to gathering examples and warnings about phishing and legal proceedings, establishing it as not only important but also the initial step in the research process.(*URL 2016 | Datasets | Research | Canadian Institute for Cybersecurity | UNB*, n.d.)

Dataset Pre-processing

After installing the necessary software and importing data, the initial step involves data pre-processing. This includes providing a basic statistical overview of each feature and refining the dataset to ensure the absence of missing values. Once null columns are removed and null values are replaced with appropriate identifiers (-1 for missing, 0 for phishing URLs, and 1 for authentic URLs), data and labels are extracted. The removal of object-type columns follows, given the focus on integer values. As object-type columns may contain both text and numeric data, the subsequent transformation converts labels and data into NumPy arrays.

Feature Selection

One of the most important tasks during model training in the field of machine learning classification is the selection of significant features. This research employs feature selection to ensure our model is trained with an optimal set of features, discarding those deemed insignificant. The correlation-based feature selection (CFS) method is employed for feature evaluation. CFS assesses features in a dataset based on their correlation with the target variable. Features with weak correlations are considered less suitable for prediction compared to those strongly connected to the target variable. CFS examines the relationship between each feature and the desired outcome, arranging features based on their closeness to the target variable. Subsequently, it selects a subset of features to endow the machine learning model with the most relevant attributes. After extracting features from the 19-target set, the dataset used to train our machine learning models undergoes the final feature selection process, with the selected features displayed in the image below.(*Introduction to The Correlation Matrix | Built In*, n.d.)

	Have_IP	Have_At	URL_Length	URL_Depth	Redirection	https_Domain	TinyURL	Prefix/Suffix	DNS_Record	Web_Traffic	Domain_Age	Domain_
count	2000.0	2000.000000	2000.0	2000.000000	2000.00000	2000.0	2000.000000	2000.000000	2000.0	2000.000000	2000.0	20(
mean	0.0	0.005500	1.0	2.782000	0.01250	1.0	0.042500	0.337000	1.0	0.596500	1.0	
std	0.0	0.073976	0.0	2.101591	0.11113	0.0	0.201777	0.472803	0.0	0.490722	0.0	
min	0.0	0.000000	1.0	0.000000	0.00000	1.0	0.000000	0.000000	1.0	0.000000	1.0	
25%	0.0	0.000000	1.0	1.000000	0.00000	1.0	0.000000	0.000000	1.0	0.000000	1.0	
50%	0.0	0.000000	1.0	2.000000	0.00000	1.0	0.000000	0.000000	1.0	1.000000	1.0	
75%	0.0	0.000000	1.0	4.000000	0.00000	1.0	0.000000	1.000000	1.0	1.000000	1.0	
max	0.0	1.000000	1.0	16.000000	1.00000	1.0	1.000000	1.000000	1.0	1.000000	1.0	

Figure 2 Statistics of data

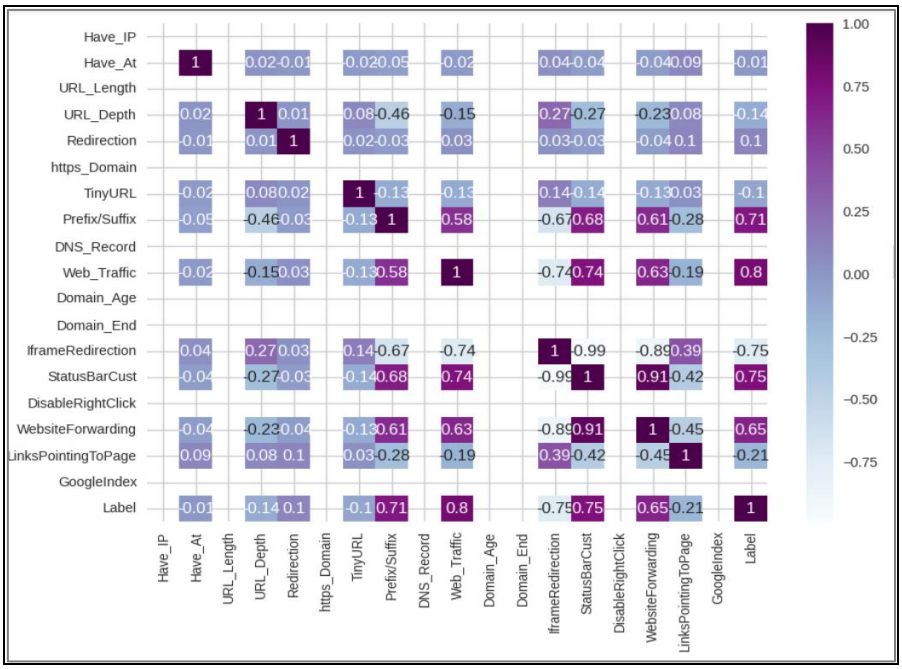


Figure 3 Statistic of correlation matrix

Data Feature Extraction

This research uses an advanced feature extraction method in the ever-evolving realm of cybersecurity to recognize and handle any phishing risks within URLs. There are a total of 19 features which are Divided into three primary areas of focus: Address Bar-based, Domain-based, and HTML & JavaScript-based features. Nine aspects of the Address Bar are examined, including URL length, "http/https" manipulations, and suspicious character identification. The three domain-based features use online traffic, domain age, and remaining lifespan to assess the authenticity of a domain. Finally, the seven HTML & JavaScript-based capabilities reveal content-based nuances, such as IFrameRedirection, Customizing the Status Bar, and Turning off Right-Click strategies. This thorough feature extraction method greatly improves proactive cybersecurity measures by providing a more advanced understanding of phishing risks.

Index	Feature	Description
1	length_of_url	Computes the length of the URL, aiding in detecting potential phishing attempts.
2	http_has	Identifies the presence of "http/https" in the domain, exposing deceptive use of the "HTTPS" token.
3	suspicious_char	Detects the '@' symbol, revealing potentially misleading addresses.
4	prefix_suffix	Examines the presence of '-' in the domain, a potential indicator of phishing.
5	dots	Quantifies the number of dots in the URL, aiding in identifying patterns.
6	slash	Checks for the presence of "//" in the URL path, indicating redirection.
7	phis_term	Identifies phishing terms within the URL.
8	sub_domain	Examines the presence of subdomains, aiding in distinguishing legitimate from potentially fraudulent URLs.
9	ip_contain	Detects the presence of IP addresses in the URL, a potential red flag for phishing attempts.
10	Web Traffic check	Assesses website popularity based on visitor count and page visits.
11	Domain Age	Calculates the survival time of a domain by determining the difference between termination and creation times.
12	Domain End	Measures the remaining lifespan of a domain by determining the difference between termination time and the current time.
13	IFrameRedirection	Identifies the use of invisible iframe tags for redirection.
14	Status Bar Customization	Detects changes to the status bar via JavaScript, potentially indicating attempts to display a fake URL.
15	Disabling Right Click	Flags the disabling of the right-click function using JavaScript.
16	Website Forwarding	Analyzes the number of redirects, with phishing websites typically having multiple redirects compared to legitimate sites.
17	IframeRedirection	Similar to the previous feature, this checks for the presence of iframe tags used for redirection.
18	LinksPointingToPage	Examines internal links, differentiating them from external links.
19	GoogleIndex	Determines if the webpage is indexed by Google, a potential indicator of legitimacy.

Figure 4 Feature extraction detail

Architecture

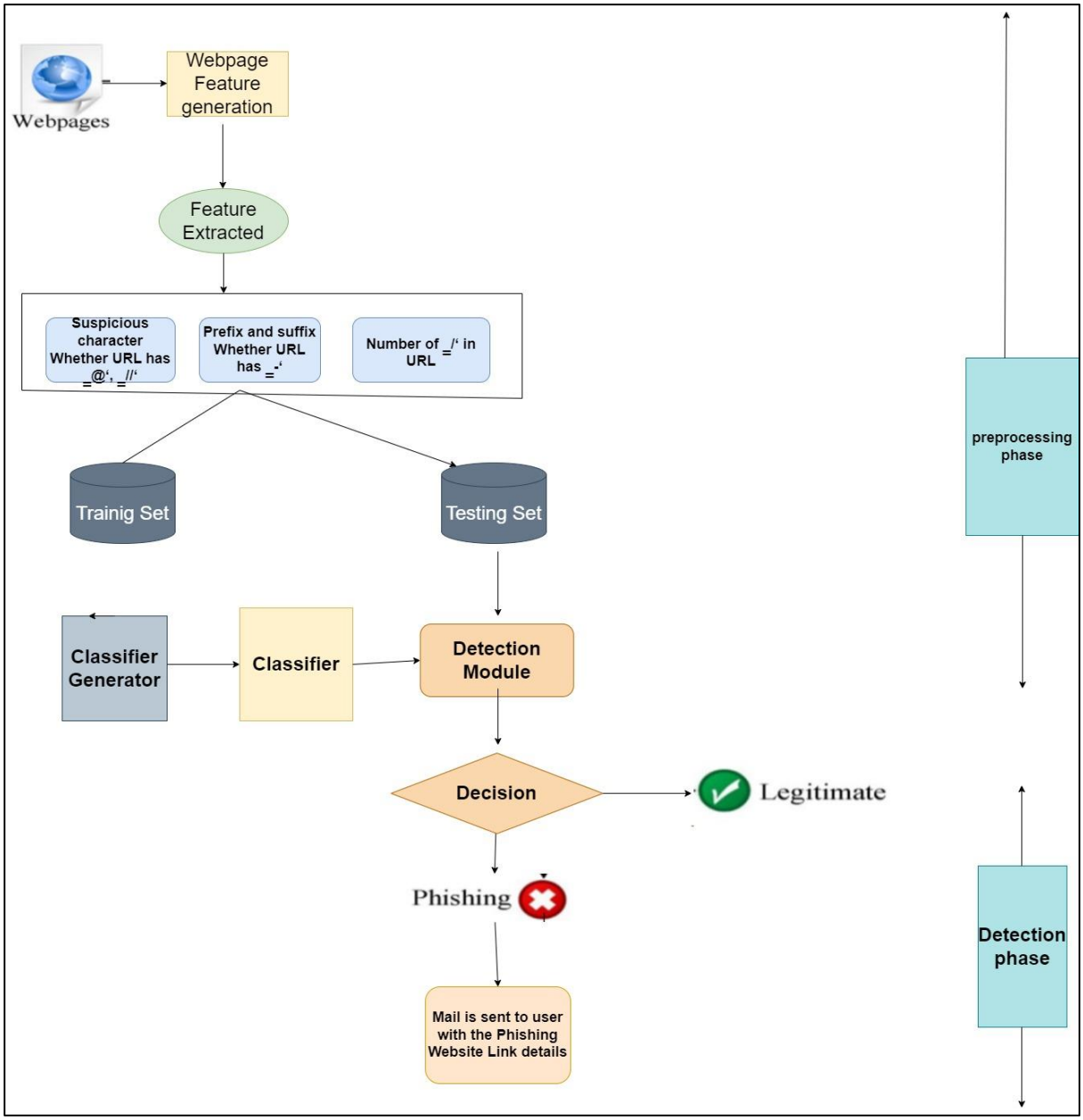


Figure 5 Implementation process flow

The proposed phishing detection system involves a multi-phase approach, beginning with webpage generation to construct a diverse dataset for training and testing. The preprocessing phase includes feature extraction, focusing on elements like the presence of suspicious characters, the count of slashes in URLs, and the existence of prefixes, suffixes, or hyphens. In the training phase, a dataset is collected, and a machine learning algorithm, such as AdaBoost classifier, gradient boosting classifier, logistic regression, and stacking classifier, is employed to generate a classifier. During the training phase, a classifier is generated with the help of a data set, which consists of phishing and legitimate website URLs. This collection of URLs is passed on to the feature extractor. The job of the feature extractor is to extract all features from these URLs. This feature extracting job depends upon the features we have selected for our feature's extractor. Now, these extracted features act as input and are passed to the classifier generator. The classifier generator generates a classifier in return, with the help of this newly generated input and some machine learning algorithms that have been selected. 1

We have discussed the algorithms in the Methodology Phase in detail.

Modelling (Training and Testing)

The approach employed for data partitioning involved the train-test split, dividing the data and labels. The dataset was evenly divided, allocating 90% for training purposes and reserving 10% for testing. The objective of this study was to construct an ensemble learning model for categorizing Phishing URL detection software into malicious or benign categories. Our ensemble learning model integrates the ADABOOST classifier, Gradient Boosting classifier, Logistic Regression classifier, and Stacking classifier. Following model development, the split data and labels were fitted to the created machine learning model, which underwent training using the training dataset. The trained model was saved, and subsequently, the accuracy and F1 score of the model were evaluated using the testing dataset.

In the final phase, our system underwent testing on the dataset created for fairness. The initial dataset was split into 90%-10%, with 90% used for system training and the remaining 10% for testing. This 10% dataset included both phishing and legitimate websites. The testing results are comprehensively presented in this chapter, featuring detailed outcomes from each individual algorithm. Each algorithm's results include a Confusion Matrix, and associated metrics like recall, precision, f1-score, accuracy, macro-average, and weighted average.

Precision: Precision is the measure of accurately identified phishing URLs out of all the URLs that the classifier labeled as phishing. (Shung, n.d.) .
The formula for calculating precision is (Shung, n.d.):

$$\text{Precision} = \frac{\text{True Positive}}{(\text{True Positive} + \text{False Positive})}$$

Precision

Recall:
Recall is the measure of accurately identified phishing URLs out of all the phishing URLs present in the dataset.

$$\text{Recall} = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

$$= \frac{True\ Positive}{Total\ Actual\ Positive}$$

The formula for calculating Recall

F1-score: The F1-score is calculated by incorporating both the values of recall and precision, representing a harmonic mean between the two. It serves as a metric that reflects a balance between recall and precision. The formula for the F1-score is expressed as: (*Accuracy, Precision, Recall or F1? | by Koo Ping Shung | Towards Data Science*, n.d.)

$$\text{F1-score} = \left(\frac{\text{Recall}^{-1} + \text{Precision}^{-1}}{2}\right)^{-1} = 2 * \frac{(\text{Precision} * \text{Recall})}{(\text{Precision} + \text{Recall})}$$

Confusion Matrix
The primary tool for assessing the efficiency of a classification algorithm is the confusion matrix, widely recognized for its effectiveness. Several essential parameters will be employed to gauge the performance of the models implemented in the proposed research. The model's effectiveness is unveiled through a comprehensive examination of the confusion matrix. Moreover, diverse performance metrics can be derived by utilizing the individual components of the confusion matrix, providing additional insights and detailed information.

True Positive (TP): "The percentage of values correctly identified as true by the model, considering both the actual truth and the predicted outcomes."

True Negative (TN): "The true negative represents the percentage of values that are genuinely negative, and the model correctly predicts a negative outcome as well.".

False Positive (FP): The False Positive is the count of truly negative values for which the model predicted a positive outcome.

False Negative (FN):
The false negative is the percentage of truly negative values that the model erroneously predicted to be true. (*Confusion Matrix in Machine Learning - GeeksforGeeks*, n.d.)

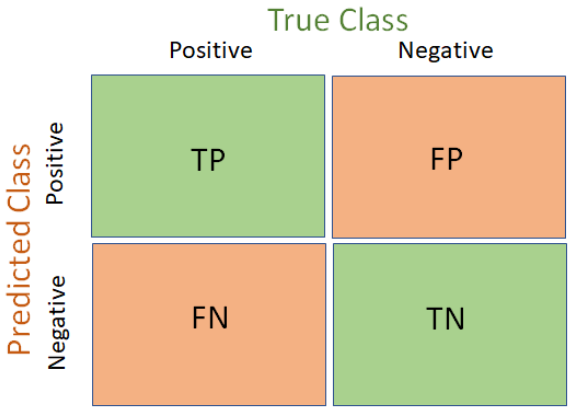
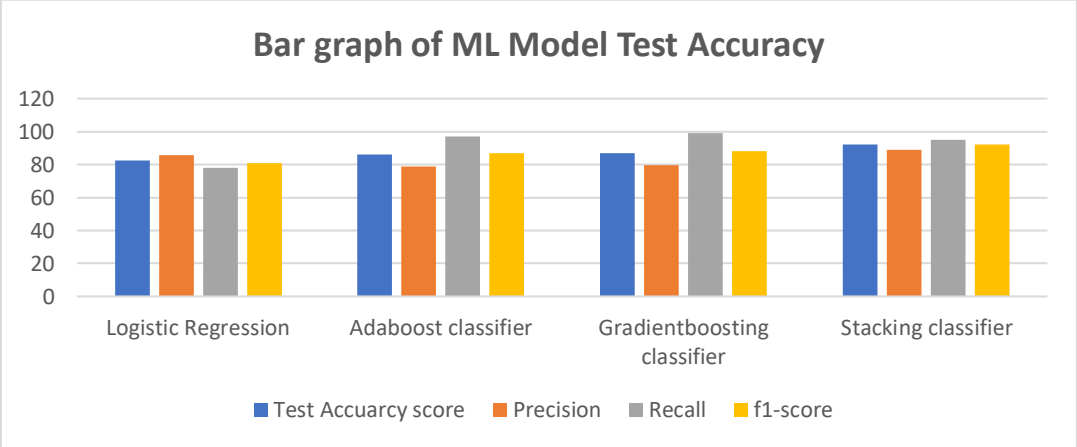


Figure 6 Confusion Matrix(*Confusion Matrix for Your Multi-Class Machine Learning Model | by Joydwip Mohajon | Towards Data Science*, n.d.)

5. Evaluation

This section presents the outcomes of implementing and evaluating machine learning models for phishing URL detection, using metrics such as accuracy, F1-score, and confusion matrix. The results indicate that Logistic Regression achieved an accuracy of 82.25%. AdaBoost Classifier demonstrated balanced accuracy, with 85.7% for phishing and 79.8% for legitimate URLs, accompanied by F1 scores of 81.7% and 83.3%, respectively. Decision Tree exhibited high accuracy at 86.7% and an F1 score of 86.12%. Gradient Boosting Classifier excelled in identifying legitimate URLs, boasting 98% accuracy for that class and an F1 score of 88% for phishing URLs. The Stacking Classifier, combining Gradient Boosting and AdaBoost, reached an accuracy of 89% for phishing and 94% for legitimate URLs, with F1 scores of 92% and 91%, respectively. These findings offer nuanced insights into the models' effectiveness, contribution.
Below is the table for a machine learning model result.

Model ML	Test Accuracy score	Precision	Recall	f1-score
Logistic Regression	82.5	83	82	82
Adaboost classifier	86	88	86	86
Gradientboosting classifier	87	89	87	87
Stacking classifier	92	92	89	89



Experiment 1: Evaluating Adaboost classifier.

Here the experiment aims to evaluate the performance of the AdaBoost classifier in terms of precision, recall and F1 score.The confusion matrix visually encapsulates the classifier's ability to categorize instances. Rows represent predicted labels, where "Phishing" is denoted by 1 and "Legitimate" by 0. Columns correspond to actual labels. True Positive (TP), False Positive (FP), False Negative (FN), and True Negative (TN) are presented with counts of 75, 25, 3, and 97, respectively.

The bar graph above illustrates key metrics from the classification report. Precision, recall, and F1 score are presented for both classes. For legitimate URLs (class 1), precision is 79.5%, recall is 97%, and the F1 score is 87. For phishing URLs (class 0), precision is 96.2%, recall is 75%, and the F1 score is 84.3.

These visualizations offer a concise overview of the AdaBoost Classifier's proficiency in distinguishing between legitimate and phishing URLs, providing valuable insights into its classification performance.

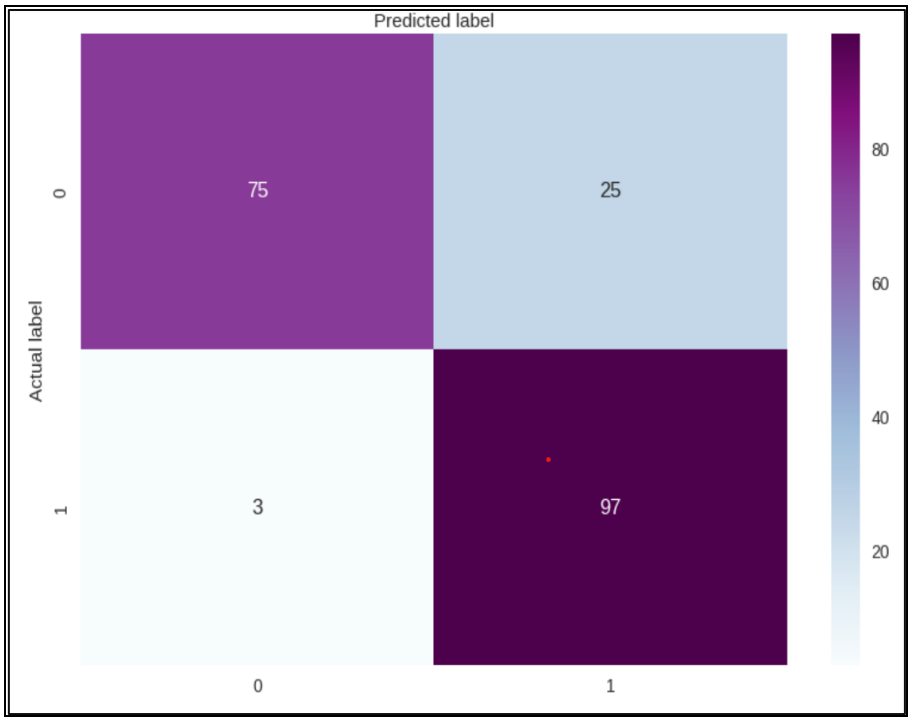


Figure 7 confusion matrix adaboost classifier

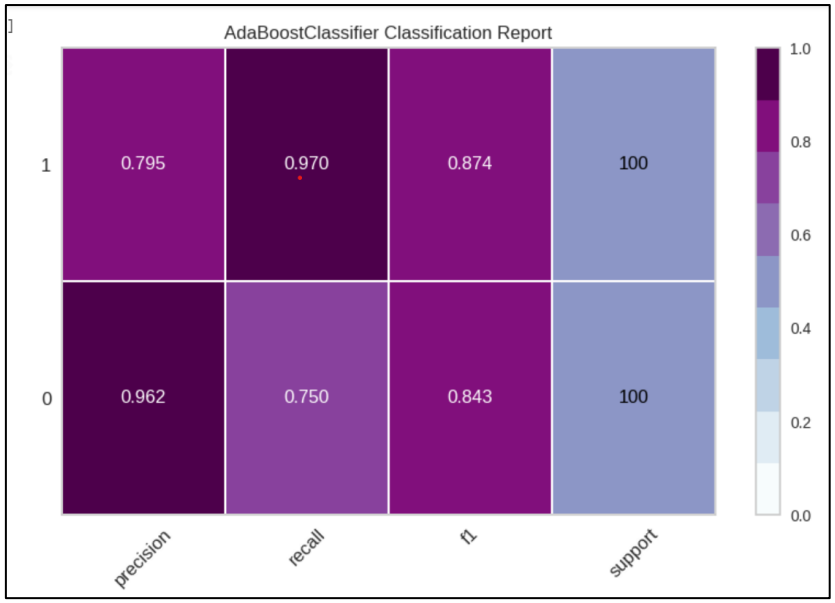


Figure 8 Adaboost classifier classification Report

Experiment 2: Evaluating Logistic Regression

In this experiment, the objective is to assess the effectiveness of the logistic classifier by measuring its precision, recall, and F1 score. The confusion matrix for the Logistic Regression classifier reveals key performance metrics in classifying phishing and legitimate URLs. With 87 true positives (legitimate URLs correctly identified), 13 false positives (phishing URLs misclassified as legitimate), 22 false negatives (legitimate URLs misclassified as phishing), and 78 true negatives (phishing URLs correctly identified), the model demonstrates a nuanced understanding of both classes.

The classification report further highlights precision, recall, and F1 score for each class. For legitimate URLs, precision is 85.7%, indicating accurate identification among predicted legitimate instances. The recall of 78% emphasizes the model's ability to capture the majority of actual legitimate instances, with an overall F1 score of 81%, reflecting balanced performance.

Regarding phishing URLs, precision stands at 79.8%, showcasing accurate identification among predicted phishing instances. The high recall of 87% demonstrates the model's effectiveness in capturing the majority of actual phishing instances, resulting in an impressive F1 score of 83%. These metrics collectively underscore the Logistic Regression classifier's robust performance and its utility in accurately discerning between phishing and legitimate URLs.

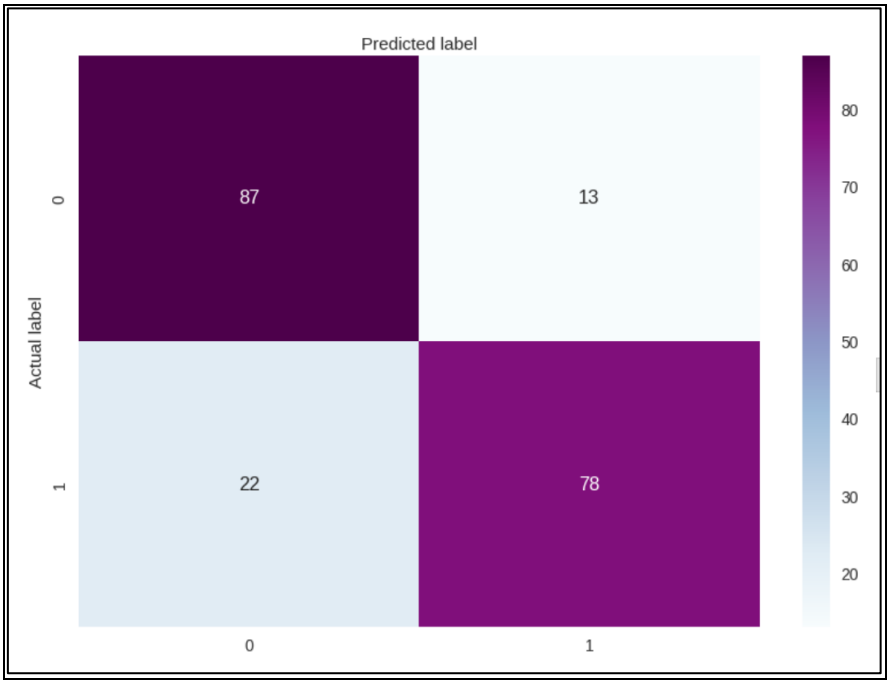


Figure 9 Confusion matrix for logistic regression

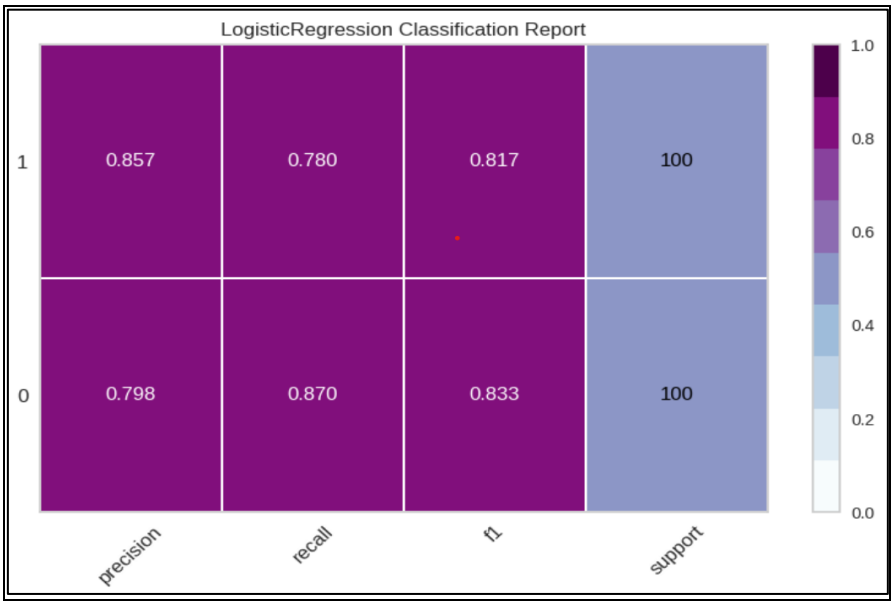


Figure 10 Classification report for logistic regression

Experiment 3: Evaluating Gradient Boosting Classifier

The aim of this experiment is to evaluate the performance metrics of the gradient boosting classifier in terms of precision, recall, and F1 score in order to determine how effective it is. The Gradient Boosting Classifier's confusion matrix visually represents its performance in distinguishing between phishing and legitimate URLs. Columns represent actual labels (0 for phishing, 1 for legitimate), and rows signify predicted labels. True positives (correctly identified legitimate URLs) total 75, while 25 false positives indicate phishing URLs misclassified. Impressively, only 1 false negative occurred, denoting a legitimate URL misclassified as phishing, and 99 true negatives signify correctly identified phishing URLs. The classification report refines these metrics. For legitimate URLs, precision stands at 79.8%, emphasizing accurate positive predictions. A 99% recall demonstrates the model's proficiency in identifying legitimate instances, yielding an 88.4 F1 score. Phishing URLs exhibit a 98.7%

precision, showcasing precise positive predictions, with a 75% recall capturing a substantial proportion. The F1 score for phishing URLs is 85.2, attesting to the model's balanced precision and recall across both classes.

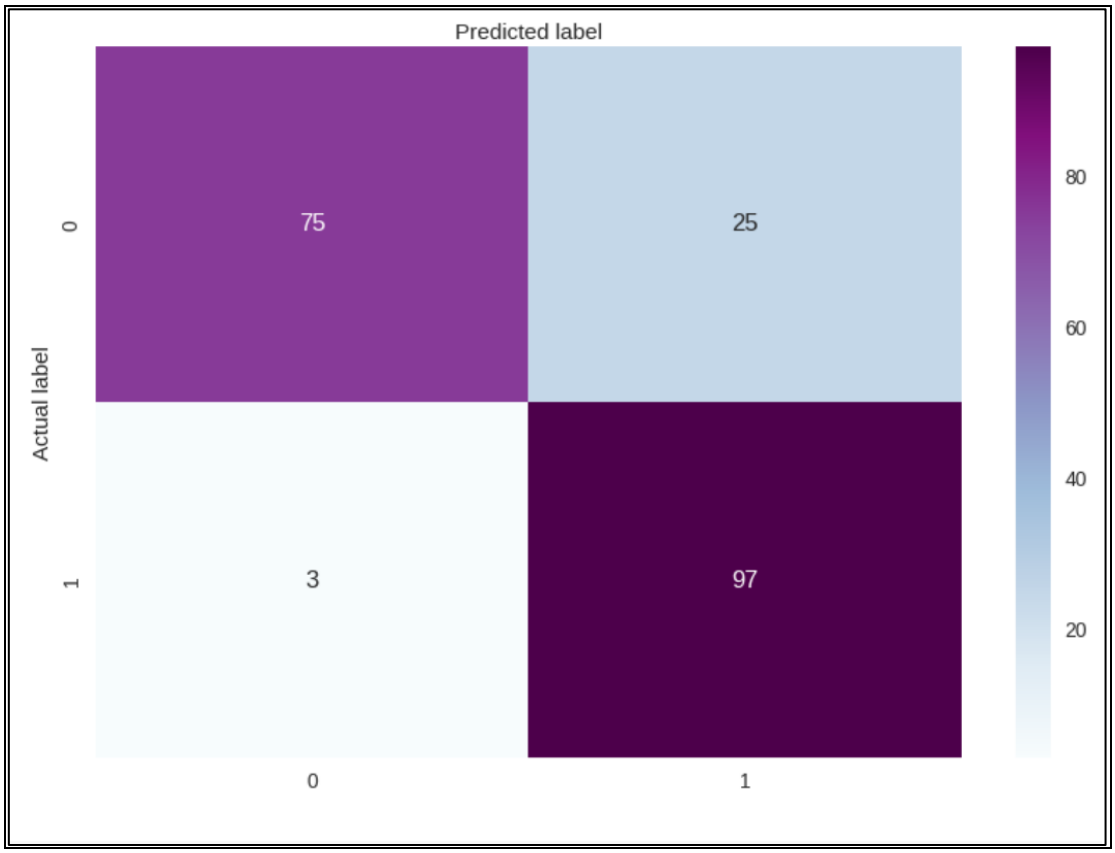


Figure 11 Confusion matrix for Gradient boosting classifier

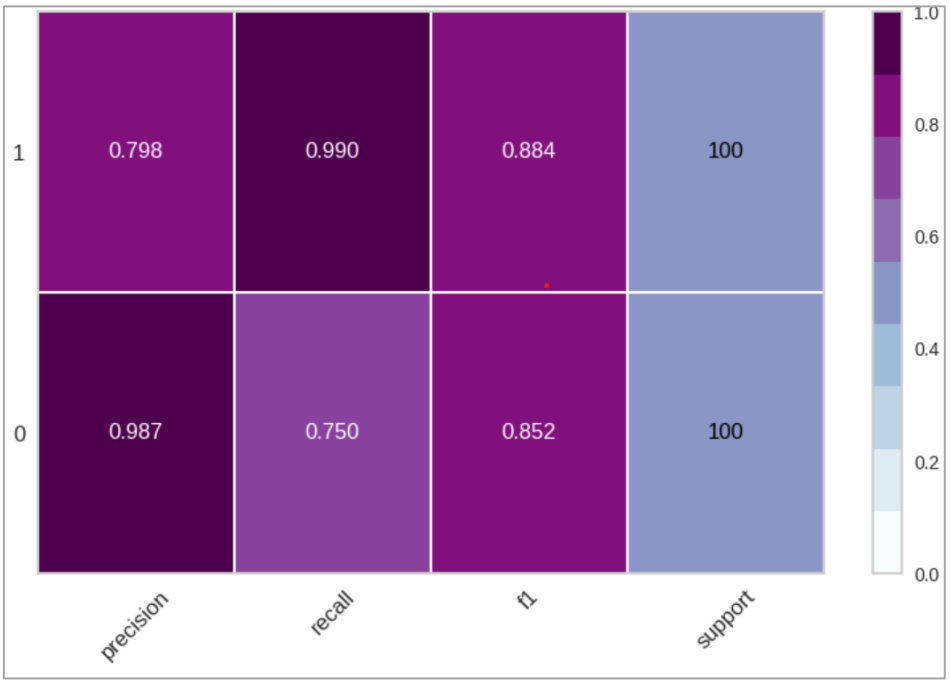


Figure 12 classification report for gradient boosting classifier

Experiment 4: Evaluating Stacking Classifier

This experiment's goal is to to measure performance parameters such as precision, recall, and F1 score in order determine the effectiveness of the stacking classifier. The confusion matrix visually encapsulates the performance of the Stacking Classifier in distinguishing between phishing (label 0) and legitimate (label 1) URLs. Columns represent actual labels, while rows indicate predicted labels. True positive instances, where phishing URLs were correctly identified, total 89, with 11 false positives. Five instances of false negatives indicate the model's failure to identify actual phishing URLs, while the true negative count is 95, denoting correct identification of legitimate URLs.

In the classification report, precision for legitimate URLs is 89.6%, showcasing accuracy in correctly identifying them, with a recall of 95%, indicating the model's effectiveness in capturing a high proportion of actual legitimate URLs. The F1 score for legitimate URLs is 92.2%, emphasizing balanced performance. For phishing URLs, precision is 94.7%, highlighting accuracy in their identification, while the recall is 89%, showing the model's effectiveness in capturing a significant proportion of actual phishing URLs. The F1 score for phishing URLs is 91.80%, illustrating a balanced performance in classifying phishing URLs.

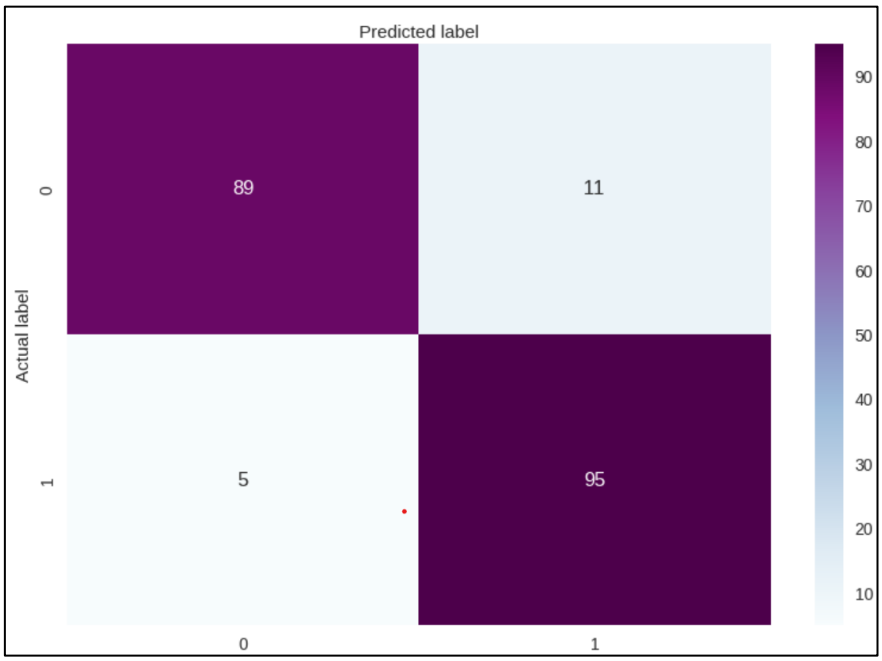


Figure 13 Confusion Metrix stacking classifiers

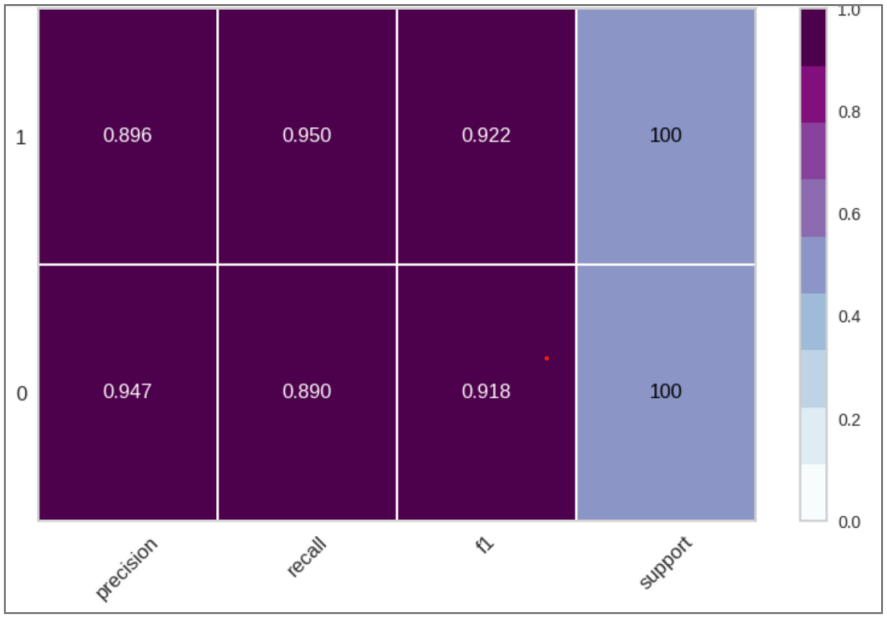


Figure 14 classipacifiers classification report for classifiers

6. Discussion

The Extra Trees Classifier's feature importance analysis highlights key contributors to the model's performance in classifying phishing and legitimate URLs. Notably, 'Web_Traffic' emerges as the most influential feature, accounting for 30% importance. This emphasizes the critical role of web traffic patterns in discerning between the two classes. Following closely, 'StatusBarCust' (22%) and 'Prefix/Suffix' (18%) underscore the significance of URL customization and structure in the classification process.

Other crucial features include 'IframeRedirection' (17%), 'URL_Depth' (15%), and 'WebsiteForwarding' (12%), indicating the importance of features related to redirection and URL structure. Features such as 'LinksPointingToPage' (8%) and 'Redirection' (5%) also contribute meaningfully. However, features like 'TinyURL' (2%), 'Have_At' (1%), and the remaining features ('GoogleIndex,' 'URL_Length,' 'DNS_Record,' 'https_Domain,' 'Domain_Age,' 'Domain_End,' 'DisableRightClick,' 'Have_IP') have limited impact on the model's decision-making process.

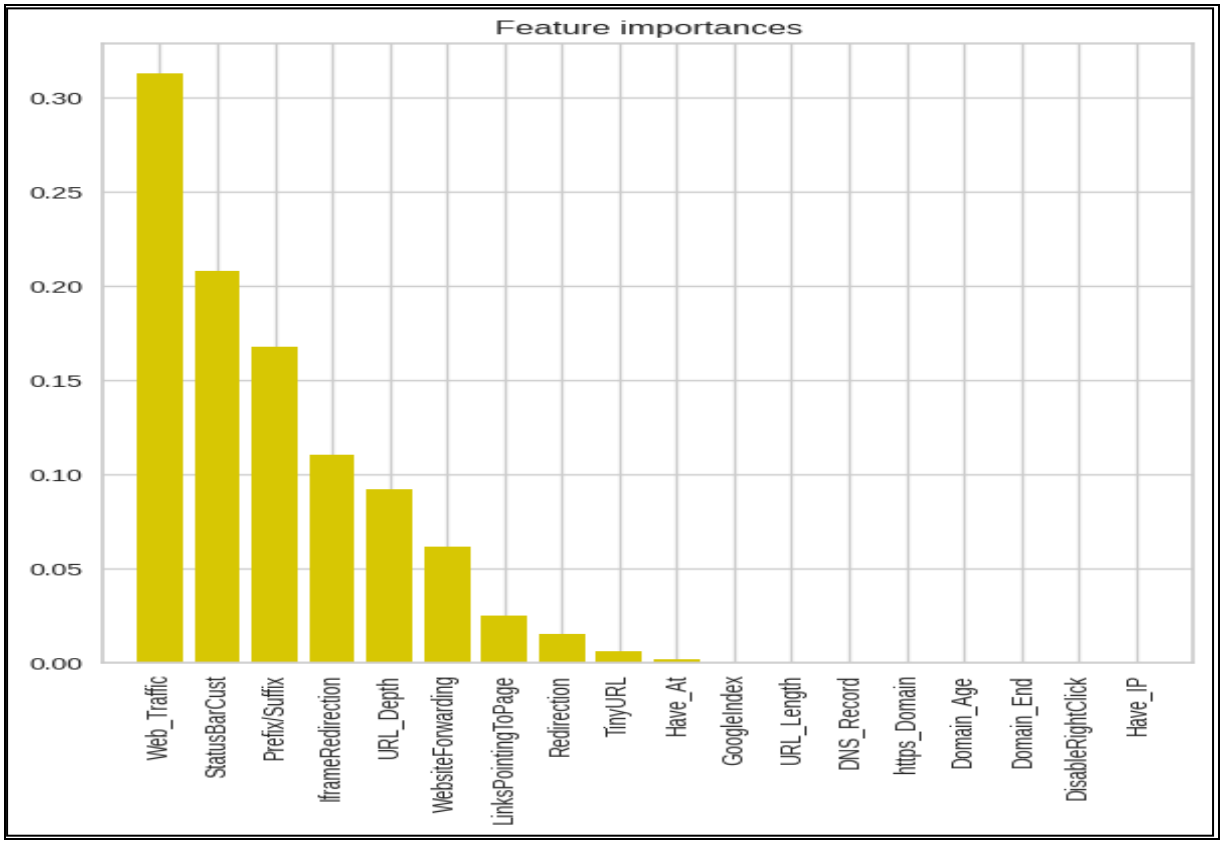


Figure 15 Important features

7. Conclusion and Future Work

The study utilized from https://www.unb.ca/cic/datasets/url-2016.html , revealing that the stacking classifier model proved effective in predicting phishing websites through feature extraction. The selected features provided substantial data for the algorithms, resulting in an accuracy of approximately 92%. This implementation demonstrates the potential of machine learning as a significant solution for phishing detection. While the achieved accuracy is commendable, there is room for improvement through better training. The research highlighted challenges in obtaining high-quality datasets containing both phishing and legitimate URLs, crucial for enhancing algorithm understanding of the nuanced boundary between legitimate and fake websites

Future work in phishing detection research involves enhancing the model's feature set for improved accuracy. Exploring advanced feature extraction techniques will contribute to a more robust model. Dynamic adaptation to emerging phishing trends in real-time is crucial for continuous effectiveness against evolving cyber threats. Behavioural dataanalysis, incorporating user interactions with URLs, can provide valuable insights for model refinement. Cross-dataset validation ensures the model's generalization across diverse cyber environments. Improving model interpretability enhances user trust, and integrating user feedback mechanisms refines the system based on practical insights. Real-time threat intelligence integration and collaboration with the cybersecurity community further strengthen the model's efficacy. Usability focus and user education programs empower individuals to recognize and report phishing threats. Finally, deploying the system in real-world environments and conducting field trials assess its performance in practical scenarios, contributing to the development of resilient anti-phishing solutions.

8. References

Abdulwakil, A., Aydin, M. A., & Aksu, D. (2017). Detecting phishing websites using support vector machine algorithm. *Pressacademia*, *5*(1), 139–142. https://doi.org/10.17261/PRESSACADEMIA.2017.582

Abutaha, M., Ababneh, M., Mahmoud, K., & Baddar, S. A. H. (2021a). URL Phishing Detection using Machine Learning Techniques based on URLs Lexical Analysis. *2021 12th International Conference on Information and Communication Systems, ICICS 2021*, 147–152. https://doi.org/10.1109/ICICS52457.2021.9464539

Abutaha, M., Ababneh, M., Mahmoud, K., & Baddar, S. A. H. (2021b). URL Phishing Detection using Machine Learning Techniques based on URLs Lexical Analysis. *2021 12th International Conference on Information and Communication Systems, ICICS 2021*, 147–152. https://doi.org/10.1109/ICICS52457.2021.9464539

Accuracy, Precision, Recall or F1? | by Koo Ping Shung | Towards Data Science. (n.d.). Retrieved December 2, 2023, from https://towardsdatascience.com/accuracy-precision-recall-or-f1-331fb37c5cb9

Almousa, M., Furst, R., & Anwar, M. (2022). Characterizing Coding Style of Phishing Websites Using Machine Learning Techniques. *Proceedings - 2022 4th International Conference on Transdisciplinary AI, TransAI 2022*, 101–105. https://doi.org/10.1109/TRANSAI54797.2022.00025

An Introduction to Logistic Regression in Python. (n.d.). Retrieved December 2, 2023, from https://www.simplilearn.com/tutorials/machine-learning-tutorial/logistic-regression-in-python

Aravena, L. T., Casas, P., Bustos-Jimenez, J., Capdehourat, G., & Findrik, M. (2023a). Phish Me if You Can - Lexicographic Analysis and Machine Learning for Phishing Websites Detection with PHISHWEB. *2023 IEEE 9th International Conference on Network Softwarization: Boosting Future Networks through Advanced Softwarization, NetSoft 2023 - Proceedings*, 252–256. https://doi.org/10.1109/NETSOFT57336.2023.10175503

Aravena, L. T., Casas, P., Bustos-Jimenez, J., Capdehourat, G., & Findrik, M. (2023b). Phish Me if You Can - Lexicographic Analysis and Machine Learning for Phishing Websites Detection with PHISHWEB. *2023 IEEE 9th International Conference on Network Softwarization: Boosting Future Networks through Advanced Softwarization, NetSoft 2023 - Proceedings*, 252–256. https://doi.org/10.1109/NETSOFT57336.2023.10175503

Castano, F., Fernandez, E. F., Alaiz-Rodriguez, R., & Alegre, E. (2023). PhiKitA: Phishing Kit Attacks dataset for Phishing Websites Identification. *IEEE Access*. https://doi.org/10.1109/ACCESS.2023.3268027

Confusion Matrix for Your Multi-Class Machine Learning Model | by Joydwip Mohajon | Towards Data Science. (n.d.). Retrieved December 14, 2023, from https://towardsdatascience.com/confusion-matrix-for-your-multi-class-machine-learning-model-ff9aa3bf7826

Confusion Matrix in Machine Learning - GeeksforGeeks. (n.d.). Retrieved December 2, 2023, from https://www.geeksforgeeks.org/confusion-matrix-machine-learning/

How Stacking Technique Boosts Machine Learning Model's Performance - Dataaspirant. (n.d.). Retrieved December 11, 2023, from https://dataaspirant.com/stacking-technique/

IEEE Xplore Full-Text PDF: (n.d.). Retrieved January 29, 2024, from https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=10103863

Introduction to The Correlation Matrix | Built In. (n.d.). Retrieved December 11, 2023, from https://builtin.com/data-science/correlation-matrix

Jain, S., & Gupta, C. (2023). A Support Vector Machine Learning Technique for Detection of Phishing Websites. *2023 6th International Conference on Information Systems and Computer Networks, ISCON 2023*. https://doi.org/10.1109/ISCON57294.2023.10111968

Kiruthiga, R., & Akila, D. (2019). Phishing websites detection using machine learning. *International Journal of Recent Technology and Engineering*, *8*(2 Special Issue 11), 111–114. https://doi.org/10.35940/IJRTE.B1018.0982S1119

Noh, N. B. M., & Nazmi Bin M Basri, M. (2021). Phishing Website Detection Using Random Forest and Support Vector Machine: A Comparison. *2021 2nd International Conference on Artificial Intelligence and Data Sciences, AiDAS 2021*. https://doi.org/10.1109/AIDAS53897.2021.9574282

Pascariu, C., & Bacivarov, I. C. (2021a). Detecting Phishing Websites through Domain and Content Analysis. *Proceedings of the 13th International Conference on Electronics, Computers and Artificial Intelligence, ECAI 2021*. https://doi.org/10.1109/ECAI52376.2021.9515165

Pascariu, C., & Bacivarov, I. C. (2021b). Detecting Phishing Websites through Domain and Content Analysis. *Proceedings of the 13th International Conference on Electronics, Computers and Artificial Intelligence, ECAI 2021*. https://doi.org/10.1109/ECAI52376.2021.9515165

Patil, S., & Dhage, S. (2019a). A Methodical Overview on Phishing Detection along with an Organized Way to Construct an Anti-Phishing Framework. *2019 5th International Conference on Advanced Computing and Communication Systems, ICACCS 2019*, 588–593. https://doi.org/10.1109/ICACCS.2019.8728356

Patil, S., & Dhage, S. (2019b). A Methodical Overview on Phishing Detection along with an Organized Way to Construct an Anti-Phishing Framework. *2019 5th International Conference on Advanced Computing and Communication Systems, ICACCS 2019*, 588–593. https://doi.org/10.1109/ICACCS.2019.8728356

(PDF) Phishing Website and Spam Content Detection using Machine Learning Algorithms. (n.d.). Retrieved December 12, 2023, from https://www.researchgate.net/publication/362532219_Phishing_Website_and_Spam_Content_Detection_using_Machine_Learning_Algorithms

Understanding the AdaBoost Algorithm | Built In. (n.d.). Retrieved December 2, 2023, from https://builtin.com/machine-learning/adaboost

URL 2016 | Datasets | Research | Canadian Institute for Cybersecurity | UNB. (n.d.). Retrieved December 11, 2023, from https://www.unb.ca/cic/datasets/url-2016.html

What is Gradient Boosting? - Gradient Boosting Explained - Displayr. (n.d.). Retrieved December 2, 2023, from https://www.displayr.com/gradient-boosting-the-coolest-kid-on-the-machine-learning-block/

What is Machine Learning? Definition, Types, Tools & More | DataCamp. (n.d.). Retrieved December 2, 2023, from https://www.datacamp.com/blog/what-is-machine-learning

Yang, Y., Li, H., & Jing, D. (2022a). A Phishing Website Detection Method Based on Multi-layer Perceptron with Mutual Information Feature Selection. *Proceedings - 2022 8th Annual International Conference on Network and Information Systems for Computers, ICNISC 2022*, 117–123. https://doi.org/10.1109/ICNISC57059.2022.00034

Yang, Y., Li, H., & Jing, D. (2022b). A Phishing Website Detection Method Based on Multi-layer Perceptron with Mutual Information Feature Selection. *Proceedings - 2022 8th Annual International Conference on Network and Information Systems for Computers, ICNISC 2022*, 117–123. https://doi.org/10.1109/ICNISC57059.2022.00034

Abdulwakil, A., Aydin, M. A., & Aksu, D. (2017). Detecting phishing websites using support vector machine algorithm. *Pressacademia*, *5*(1), 139–142. https://doi.org/10.17261/PRESSACADEMIA.2017.582

Abutaha, M., Ababneh, M., Mahmoud, K., & Baddar, S. A. H. (2021a). URL Phishing Detection using Machine Learning Techniques based on URLs Lexical Analysis. *2021 12th International Conference on Information and Communication Systems, ICICS 2021*, 147–152. https://doi.org/10.1109/ICICS52457.2021.9464539

Abutaha, M., Ababneh, M., Mahmoud, K., & Baddar, S. A. H. (2021b). URL Phishing Detection using Machine Learning Techniques based on URLs Lexical Analysis. *2021 12th International Conference on Information and Communication Systems, ICICS 2021*, 147–152. https://doi.org/10.1109/ICICS52457.2021.9464539

Accuracy, Precision, Recall or F1? | by Koo Ping Shung | Towards Data Science. (n.d.). Retrieved December 2, 2023, from https://towardsdatascience.com/accuracy-precision-recall-or-f1-331fb37c5cb9

Almousa, M., Furst, R., & Anwar, M. (2022). Characterizing Coding Style of Phishing Websites Using Machine Learning Techniques. *Proceedings - 2022 4th International Conference on Transdisciplinary AI, TransAI 2022*, 101–105. https://doi.org/10.1109/TRANSAI54797.2022.00025

An Introduction to Logistic Regression in Python. (n.d.). Retrieved December 2, 2023, from https://www.simplilearn.com/tutorials/machine-learning-tutorial/logistic-regression-in-python

Aravena, L. T., Casas, P., Bustos-Jimenez, J., Capdehourat, G., & Findrik, M. (2023a). Phish Me if You Can - Lexicographic Analysis and Machine Learning for Phishing Websites Detection with PHISHWEB. *2023 IEEE 9th International Conference on Network Softwarization: Boosting Future Networks through Advanced Softwarization, NetSoft 2023 - Proceedings*, 252–256. https://doi.org/10.1109/NETSOFT57336.2023.10175503

Aravena, L. T., Casas, P., Bustos-Jimenez, J., Capdehourat, G., & Findrik, M. (2023b). Phish Me if You Can - Lexicographic Analysis and Machine Learning for Phishing Websites Detection with PHISHWEB. *2023 IEEE 9th International Conference on Network Softwarization: Boosting Future Networks through Advanced Softwarization, NetSoft 2023 - Proceedings*, 252–256. https://doi.org/10.1109/NETSOFT57336.2023.10175503

Castano, F., Fernandez, E. F., Alaiz-Rodriguez, R., & Alegre, E. (2023). PhiKitA: Phishing Kit Attacks dataset for Phishing Websites Identification. *IEEE Access*. https://doi.org/10.1109/ACCESS.2023.3268027

Confusion Matrix for Your Multi-Class Machine Learning Model | by Joydwip Mohajon | Towards Data Science. (n.d.). Retrieved December 14, 2023, from https://towardsdatascience.com/confusion-matrix-for-your-multi-class-machine-learning-model-ff9aa3bf7826

Confusion Matrix in Machine Learning - GeeksforGeeks. (n.d.). Retrieved December 2, 2023, from https://www.geeksforgeeks.org/confusion-matrix-machine-learning/

How Stacking Technique Boosts Machine Learning Model's Performance - Dataaspirant. (n.d.). Retrieved December 11, 2023, from https://dataaspirant.com/stacking-technique/

IEEE Xplore Full-Text PDF: (n.d.). Retrieved January 29, 2024, from https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=10103863

Introduction to The Correlation Matrix | Built In. (n.d.). Retrieved December 11, 2023, from https://builtin.com/data-science/correlation-matrix

Jain, S., & Gupta, C. (2023). A Support Vector Machine Learning Technique for Detection of Phishing Websites. *2023 6th International Conference on Information Systems and Computer Networks, ISCON 2023*. https://doi.org/10.1109/ISCON57294.2023.10111968

Kiruthiga, R., & Akila, D. (2019). Phishing websites detection using machine learning. *International Journal of Recent Technology and Engineering*, *8*(2 Special Issue 11), 111–114. https://doi.org/10.35940/IJRTE.B1018.0982S1119

Noh, N. B. M., & Nazmi Bin M Basri, M. (2021). Phishing Website Detection Using Random Forest and Support Vector Machine: A Comparison. *2021 2nd International Conference on Artificial Intelligence and Data Sciences, AiDAS 2021*. https://doi.org/10.1109/AIDAS53897.2021.9574282

Pascariu, C., & Bacivarov, I. C. (2021a). Detecting Phishing Websites through Domain and Content Analysis. *Proceedings of the 13th International Conference on Electronics, Computers and Artificial Intelligence, ECAI 2021*. https://doi.org/10.1109/ECAI52376.2021.9515165

Pascariu, C., & Bacivarov, I. C. (2021b). Detecting Phishing Websites through Domain and Content Analysis. *Proceedings of the 13th International Conference on Electronics, Computers and Artificial Intelligence, ECAI 2021*. https://doi.org/10.1109/ECAI52376.2021.9515165

Patil, S., & Dhage, S. (2019a). A Methodical Overview on Phishing Detection along with an Organized Way to Construct an Anti-Phishing Framework. *2019 5th International Conference on Advanced Computing and Communication Systems, ICACCS 2019*, 588–593. https://doi.org/10.1109/ICACCS.2019.8728356

Patil, S., & Dhage, S. (2019b). A Methodical Overview on Phishing Detection along with an Organized Way to Construct an Anti-Phishing Framework. *2019 5th International Conference on Advanced Computing and Communication Systems, ICACCS 2019*, 588–593. https://doi.org/10.1109/ICACCS.2019.8728356

(PDF) Phishing Website and Spam Content Detection using Machine Learning Algorithms. (n.d.). Retrieved December 12, 2023, from https://www.researchgate.net/publication/362532219_Phishing_Website_and_Spam_Content_Detection_using_Machine_Learning_Algorithms

Understanding the AdaBoost Algorithm | Built In. (n.d.). Retrieved December 2, 2023, from https://builtin.com/machine-learning/adaboost

URL 2016 | Datasets | Research | Canadian Institute for Cybersecurity | UNB. (n.d.). Retrieved December 11, 2023, from https://www.unb.ca/cic/datasets/url-2016.html

What is Gradient Boosting? - Gradient Boosting Explained - Displayr. (n.d.). Retrieved December 2, 2023, from https://www.displayr.com/gradient-boosting-the-coolest-kid-on-the-machine-learning-block/

What is Machine Learning? Definition, Types, Tools & More | DataCamp. (n.d.). Retrieved December 2, 2023, from https://www.datacamp.com/blog/what-is-machine-learning

Yang, Y., Li, H., & Jing, D. (2022a). A Phishing Website Detection Method Based on Multi-layer Perceptron with Mutual Information Feature Selection. *Proceedings - 2022 8th Annual International Conference on Network and Information Systems for Computers, ICNISC 2022*, 117–123. https://doi.org/10.1109/ICNISC57059.2022.00034

Yang, Y., Li, H., & Jing, D. (2022b). A Phishing Website Detection Method Based on Multi-layer Perceptron with Mutual Information Feature Selection. *Proceedings - 2022 8th Annual International Conference on Network and Information Systems for Computers, ICNISC 2022*, 117–123. https://doi.org/10.1109/ICNISC57059.2022.00034