Communications on Applied Nonlinear Analysis
ISSN: 1074-133X
Vol 32 No. 3 (2025)

Phishing URL Detection with Gradient Boosting Classifier

Dr. Narayana Rao Appini[1], Mr. V. Bhuvana Kumar[2], Mr. N. Yedukondalu[3]

[1]Professor & Head, AIDS & IT, NBKR Institute of Science and Technology, India.

E-mail: narayanaraoappini@gmail.com

[2,3]Assistant Professor, Computer Science & Engineering, N.B.K.R Institute of Science and Technology, India.

[2]E-mail: vbkumar1993@gmail.com, [3]E-mail: sevenhills.559@gmail.com

Article History:

Received: *09-08-2024*

Revised: *29-09-2024*

Accepted: *10-10-2024*

Abstract:

Phishing attacks continue to pose a serious and persistent risk to internet security, jeopardizing people's and businesses' financial stability and privacy. The precise and timely identification of these malicious websites remains a crucial challenge in the field of cyber security. This paper provides a new strategy employing Gradient Boosting Classifiers (GBCs) to overcome this issue. Phishing websites are purposely built to mimic legitimate sites to fool users into providing sensitive information. As a result, many malicious sites exhibit tiny traits that identify them from real ones. Traditional rule-based systems typically struggle to adequately identify these complex variances. In contrast, Gradient Boosting provides an ensemble learning framework that utilises the combined strength of weak classifiers, resulting in a robust model capable of recognising these elusive characteristics. Our experimental findings show that our suggested strategy performs better than others in a number of parameters, including accuracy, precision, recall, and F1-score. Crucially, our model demonstrates exceptional resistance to aggressive tactics that are frequently employed to hide the actual purpose of phishing websites. This study represents a substantial step forward in the fight to improve internet security and shield consumers from the constant danger of phishing scams.

Keywords: phishing attacks; machine learning; internet security; cybersecurity; phishing URLs.

1. Introduction

Phishing attacks are cyber threats that attempt to deceive people into disclosing sensitive information such as login passwords, financial information, or personal data [1]. Attackers frequently impersonate trusted companies such as banks, online retailers, or government organisations, sending emails or messages including links to fraudulent websites [2]. When victims enter their personal information on these bogus websites, attackers might use it for nefarious purposes like identity theft or unauthorised financial transactions [3].

Phishing attacks have become increasingly complex and difficult to detect, emphasising the need for enhanced awareness and safeguards [4]. Phishing can take place via phone calls, text messages, or social media communications, in addition to emails [5]. Attackers may use hurry, fear, or emotional manipulation to get victims to move swiftly and reveal critical information. To detect phishing, people should check the origins of emails and messages, look for indications such as poor grammar, strange links, or requests for personal information, keep software and security systems up to date, and use multi-factor authentication when possible.

Organisations may prevent phishing attempts by educating employees about the risks, putting in place security measures such as firewalls and anti-virus software, and routinely monitoring systems for signs of compromise [6,7]. Email filters and URL reputation systems are two examples of technical solutions that can assist detect and thwart phishing attempts.

Finally, phishing assaults pose a significant hazard to both individuals and organisations, needing vigilance and proactive actions to protect against them. Using phishing URL detection software and following best practices can dramatically lower your chances of falling victim to these assaults.

1.1 Motivation

The reason for focusing on phishing URL detection stems from the pressing need for improved cybersecurity measures as phishing assaults become more prevalent and sophisticated. The expanding use of digital platforms and online interactions has increased fraudsters' attack surface, magnifying the impact of phishing on individuals and organisations [8].

Understanding the serious repercussions of successful phishing attempts, which include financial losses, reputational harm, and the compromise of sensitive data, emphasises the need of creating efficient detection methods. The flexibility and complexity of phishing strategies emphasise the limitations of standard detection methods and the need for new alternatives.

Machine learning and artificial intelligence provide interesting options for supplementing existing security measures. Machine learning has the potential to greatly improve phishing URL detection by using sophisticated algorithms to examine large datasets, spot trends, and find minute signs of criminal intent. Proactive detection techniques can interrupt phishing efforts early on, reducing their impact on victims.

The main objective is to strengthen the security posture of online environments, making themsafer for all users globally.Strengthening defences against phishing attempts and improving the resilience of digital infrastructuresare consistent with the project's commitment to protecting sensitive data and sustaining trust in digital interactions.

1.2 Problem Statement

This study seeks to use a Gradient Boosting Classifier (GBC) to detect phishing URLs by combining insights from many weak learners to uncover subtle patterns in datasets. The research will collect a broad dataset of both phishing and authentic URLs for model training and assessment. Using rigorous pre-processing and feature engineering, information such as domain age, URL length, and HTTPS presence will be retrieved to assist the classifier in distinguishing between dangerous and benign URLs.

Ensemble learning using GBC provides a strong solution for the difficult challenge of detecting phishing URLs. Its iterative improvement approach brings together several views, improving forecast accuracy and resistance to complex phishing attempts. The classifier's performance and efficacy in minimising phishing attacks will be closely monitored through rigorous validation and assessment, utilising criteria like accuracy, precision, recall, and F1 score.

The project aims to strengthen cybersecurity defences by creating a sophisticated detection system capable of spotting phishing URLs ahead of time, enabling a safer digital environment for consumers and organisations while also protecting sensitive information.

1.3 Objectives

- Create and apply machine learning algorithms for effective and sophisticated phishing detection.
- Develop a system that can dynamically react to changing phishing strategies, assuring continuous efficacy.
- Create solutions that integrate smoothly into current platforms while prioritising user experience.
- Improve the Gradient Boosting Classifier model's accuracy and efficiency in phishing detection.
- Create scalable and effective techniques for combating global phishing attacks.
- Ensuring dependability requires validating the Gradient Boosting Classifier model's performance on separate datasets.
- To determine the Gradient Boosting Classifier model's efficacy, compare its performance to that of existing approaches.
- Create a user-friendly interface or integrate with current prediction algorithms to ensure smooth URL detection.

2. Review of Literature

The literature on malicious URL detection has advanced tremendously, with a variety of approaches being used to improve detection accuracy and efficiency. [9] Proposed a unique technique to detecting phishing URLs that relies on URL content rather than typical string properties. This technique starts with simple feature screening, then moves on to simulated environmental evaluation to extract page content, and finally to CNN identification. Previous research, like as [10], shown the efficacy of CNNs in text categorization, emphasising its potential in URL analysis. The transition to deep learning models, including work by [11] and [12], established CNNs' superiority in capturing complex patterns in URLs. Comparative investigations, such as those conducted by [13], demonstrated that CNNs outperform classical classifiers and other deep learning architectures in URL classification. [14] Expands on this foundation, addressing the shortcomings of prior approaches by using CNNs to analyse web page content, resulting in greater detection accuracy and resilience against advanced phishing strategies.

The literature on phishing URL detection includes substantial research into categorization approaches to improve performance and accuracy. [15] Performed a detailed performance research, comparing several classification methods for identifying phishing URLs. This study adds to earlier research by investigating the efficacy of machine learning algorithms in detecting phishing attacks. Previous research, such as that of [16], has emphasised the need of using pattern recognition skills to discriminate between phishing and legal URLs. Furthermore, [17] found that classification algorithms, particularly tree-based classifiers, may achieve high accuracy in detecting phishing URLs. [18] Adds

to this corpus of research by thoroughly comparing the performance of several classifiers. Using a dataset of 4,500 URLs. The findings provide light on the efficacy of categorization approaches and highlight the need of using machine learning to tackle phishing threats in today's changing cybersecurity scene.

In the field of cybersecurity, detecting phishing websites is critical, and [19] provides a unique technique in their article, "A New Method for Detection of Phishing Websites: URL Detection." Building on prior studies, they discusses the necessity for increased security measures to counteract emerging phishing strategies. The study builds on previous research, such as that of [16], which emphasises the importance of identifying fake URLs in preventing data breaches and identity theft. Parekh's technique detects phishing websites using the Random Forest algorithm and consists of three major phases: parsing, heuristic data classification, and performance analysis. This unique technique represents a significant improvement in the industry, providing a viable answer to the persistent difficulty of detecting and reducing phishing attacks in online contexts.

[20] Addresses the growing problem of phishing attempts by utilising machine learning methods. This study expands on previous studies, including those by [16] and [17], which emphasise the need of using sophisticated algorithms to detect fake websites. This work efficiently detects phishing websites by extracting and assessing numerous elements of real and phishing URLs using CNN LSTM, CNN Bi-LSTM, Logistic Regression, and XGBoost algorithms. The study's goal is to discover the best dependable machine learning algorithm by comparing accuracy, false positive, and false negative rates. Their study, with its comprehensive methodology, provides vital insights into the creation of strong phishing detection systems, addressing the crucial need for proactive steps to protect against cyber-attacks.

[21] Investigates the serious issue of phishing assaults and proposes a novel solution based on machine learning methods. Their findings are consistent with prior research, including studies by [16] and [17], which emphasise the need of using sophisticated algorithms to battle phishing attacks. Their method seeks to successfully identify and prevent phishing assaults by combining methods of blacklisting and semantic analysis. The research entails creating a database of phishing websites and analysing text, links, graphics, and other site data for pattern detection. Through extensive testing and comparison with existing methodologies, [21] proves the usefulness of the suggested solution in significantly minimising the phishing problem, adding to ongoing efforts to improve cyber security in the digital realm.

3. Existing System

The present method for identifying phishing emails involves evaluating 17 components, some of which are out of date and may result in false positives. URLs with the "@" symbol, for example, or mismatches between anchor text and URL properties, might be identified as phishing attempts, despite acceptable usage such as website registration. To solve this, a more advanced identification approach is developed that uses neural networks, MIME properties, and contextual analysis. The approach seeks to adapt to contemporary phishing methods by taking into account six contextual factors, including the overall number of links in an email. However, differentiating real emails from phishing efforts remains difficult. Furthermore, the use of JavaScript functions in the detection process may offer extra

information about phishing behaviour. Detecting harmful JavaScript methods such as eval() or exec() is critical in identifying possible phishing sites, however this method may be ineffective if the functionality of these functions is rewritten by other ways. Additionally, because there are less features accessible for website identification than there are for phishing emails, identifying fake websites presents a harder problem. This complication emphasises the significance of our work, which focuses largely on improving accuracy in detecting phishing sites.

4. Proposed System

This section presents the suggested paradigm for phishing attack detection. The approach is intended to detect phishing attempts by examining the features of phishing websites and cross-referencing them against a blacklist. The distinguishing between harmful and genuine web pages under our proposed system relies heavily on certain attributes. These characteristics include URLs, domain identities, page aesthetics and content, the look of the online address bar, and the impact of human behavior. Nonetheless, the properties pertaining to URLs and domain names are the major emphasis of this article. We evaluate these properties using a variety of standards, such as IP addresses, URL length, the existence of redirect symbols "//," and URLs with "mail" or "mail-to" attributes. To assess these characteristics and distinguish phishing URLs from those linked to potentially dangerous websites, a broad array of criteria is used. There are numerous crucial phases in the detecting process:

- Making use of a blacklist database that gathers URLs linked to well-known phishing websites
- Examining the IP Address: If an IP address is found in the URL, such as "http://125.98.3.123/fake.html," it may be a sign of an attempted theft of private information.
- Examining mail/mail-to attributes: If they are present in the URL, there may be a risk that they are being used to get user data.

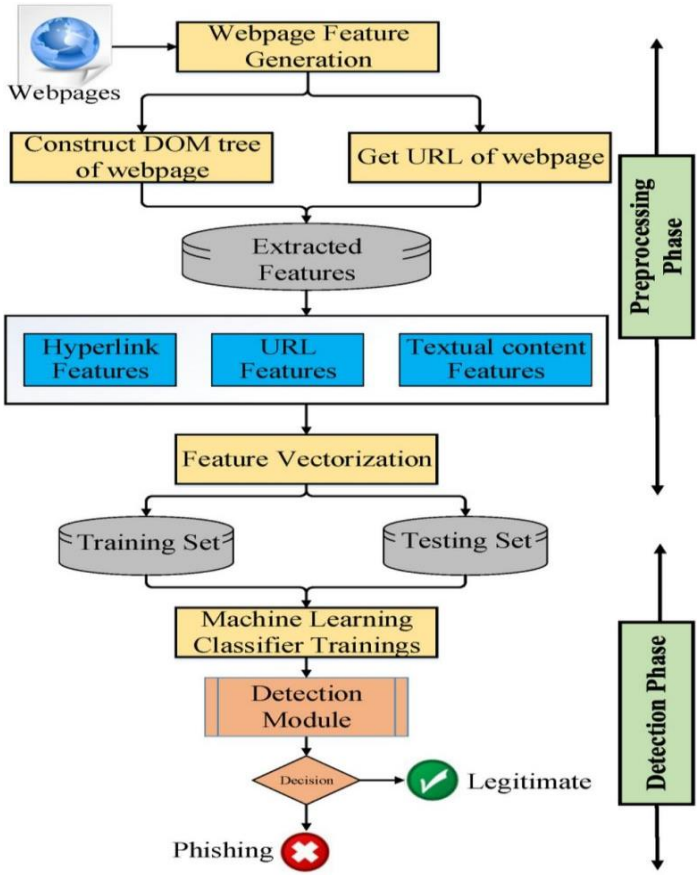

Figure 1: Overall proposed architecture

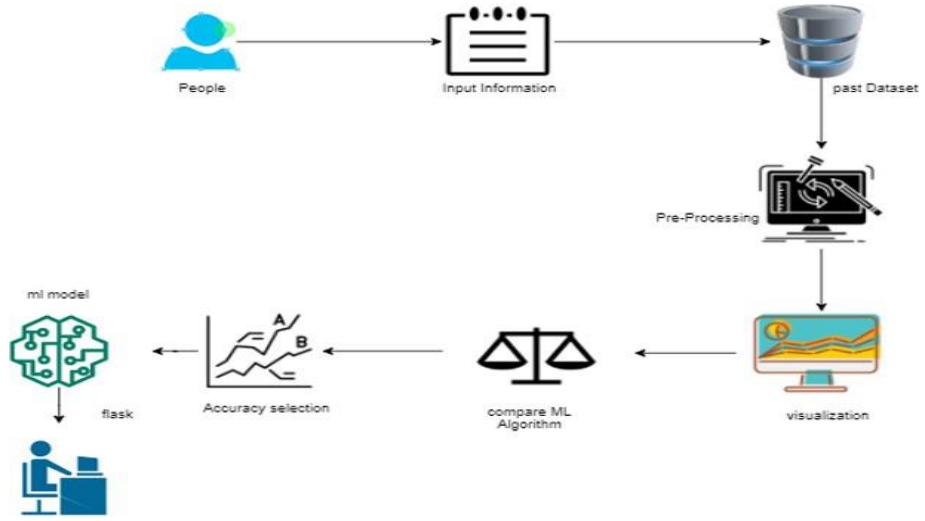

Figure 2: Model Architecture

5. Methodology

Phishing is a serious cybersecurity threat because its offenders use dishonest internet methods to get private information without authorization. These attacks provide access gates for more serious cyber threats, such as identity theft, ransomware attacks, and denial-of-service attacks. Unfortunately, human

susceptibility—which is typified by bad judgment, ignorance, and inattention to detail—is the source of both the prevalence and effectiveness of phishing.

Phishers employ a wide range of techniques to trick their victims, from intricate social engineering schemes to the deft construction of false websites. While social engineering techniques take use of psychological weaknesses to trick people into disclosing private information, fraudulent websites stealthily impersonate trustworthy ones to trick visitors into inadvertently disclosing personal information. Conventional detection approaches have a significant problem in effectively countering these emerging strategies.

Machine learning offers an anticipatory approach to counteract phishing by identifying recurrent patterns found in various phishing efforts. By examining distinguishable characteristics and actions characteristic of phishing websites, machine learning algorithms are able to quickly identify and stop these attacks as they happen. This proactive strategy ensures a strong defense against the constantly changing panorama of harmful activity online and allows for real-time detection and mitigation, enabling ongoing adjustments to new phishing strategies.

Machine learning algorithms for phishing detection need the extraction of key information from URLs. These attributes include the domain name, subdomain, route, query parameters, and other relevant information. Machine learning algorithms are able to identify patterns and traits typical of phishing websites by analyzing these attributes. Unusual domain patterns, such as misspellings or suspicious top-level domains, are good indicators of possible phishing sites. Similarly, query parameters that request sensitive information such as passwords or credit card numbers are strong evidence of criminal intent.

5.1 Overview Of Methodology

5.1.1 Data Loading and Familiarization:

- The dataset from Kaggle is imported as a CSV file.
- To understand the dataset's composition and properties, EDA approaches are used.
- This includes analyzing the dataset's size, columnar layout, data kinds, and undertaking descriptive statistical analysis.

	count	mean	std	min	25%	50%	75%	max
UsingIP	11054.0	0.313914	0.949495	-1.0	-1.0	1.0	1.0	1.0
LongURL	11054.0	-0.633345	0.765973	-1.0	-1.0	-1.0	-1.0	1.0
ShortURL	11054.0	0.738737	0.674024	-1.0	1.0	1.0	1.0	1.0
Symbol@	11054.0	0.700561	0.713625	-1.0	1.0	1.0	1.0	1.0
Redirecting//	11054.0	0.741632	0.670837	-1.0	1.0	1.0	1.0	1.0
PrefixSuffix-	11054.0	-0.734938	0.678165	-1.0	-1.0	-1.0	-1.0	1.0
SubDomains	11054.0	0.064049	0.817492	-1.0	-1.0	0.0	1.0	1.0
HTTPS	11054.0	0.251040	0.911856	-1.0	-1.0	1.0	1.0	1.0
DomainRegLen	11054.0	-0.336711	0.941651	-1.0	-1.0	-1.0	1.0	1.0
Favicon	11054.0	0.628551	0.777804	-1.0	1.0	1.0	1.0	1.0
NonStdPort	11054.0	0.728243	0.685350	-1.0	1.0	1.0	1.0	1.0
HTTPSDomainURL	11054.0	0.675231	0.737640	-1.0	1.0	1.0	1.0	1.0
RequestURL	11054.0	0.186720	0.982458	-1.0	-1.0	1.0	1.0	1.0
AnchorURL	11054.0	-0.076443	0.715116	-1.0	-1.0	0.0	0.0	1.0
LinksInScriptTags	11054.0	-0.118238	0.763933	-1.0	-1.0	0.0	0.0	1.0
ServerFormHandler	11054.0	-0.595712	0.759168	-1.0	-1.0	-1.0	-1.0	1.0
InfoEmail	11054.0	0.635788	0.771899	-1.0	1.0	1.0	1.0	1.0
AbnormalURL	11054.0	0.705446	0.708796	-1.0	1.0	1.0	1.0	1.0
WebsiteForwarding	11054.0	0.115705	0.319885	0.0	0.0	0.0	0.0	1.0
StatusBarCust	11054.0	0.762077	0.647516	-1.0	1.0	1.0	1.0	1.0
DisableRightClick	11054.0	0.913877	0.406009	-1.0	1.0	1.0	1.0	1.0
UsingPopupWindow	11054.0	0.613353	0.789845	-1.0	1.0	1.0	1.0	1.0
IframeRedirection	11054.0	0.816899	0.576807	-1.0	1.0	1.0	1.0	1.0
AgeofDomain	11054.0	0.061335	0.998162	-1.0	-1.0	1.0	1.0	1.0
DNSRecording	11054.0	0.377239	0.926158	-1.0	-1.0	1.0	1.0	1.0
WebsiteTraffic	11054.0	0.287407	0.827680	-1.0	0.0	1.0	1.0	1.0
PageRank	11054.0	-0.483626	0.875314	-1.0	-1.0	-1.0	1.0	1.0
GoogleIndex	11054.0	0.721549	0.692395	-1.0	1.0	1.0	1.0	1.0
LinksPointingToPage	11054.0	0.343948	0.569936	-1.0	0.0	0.0	1.0	1.0
StatsReport	11054.0	0.719739	0.694276	-1.0	1.0	1.0	1.0	1.0
class	11054.0	0.113986	0.993527	-1.0	-1.0	1.0	1.0	1.0

Figure 3: Description of Data set

5.1.2 Data Visualization

- Visual representations such as correlation heatmaps, pair plots, and pie charts help understand data distribution and feature correlations.
- Determining the importance of characteristics and their correlations with the target variable are made easier by these representations.

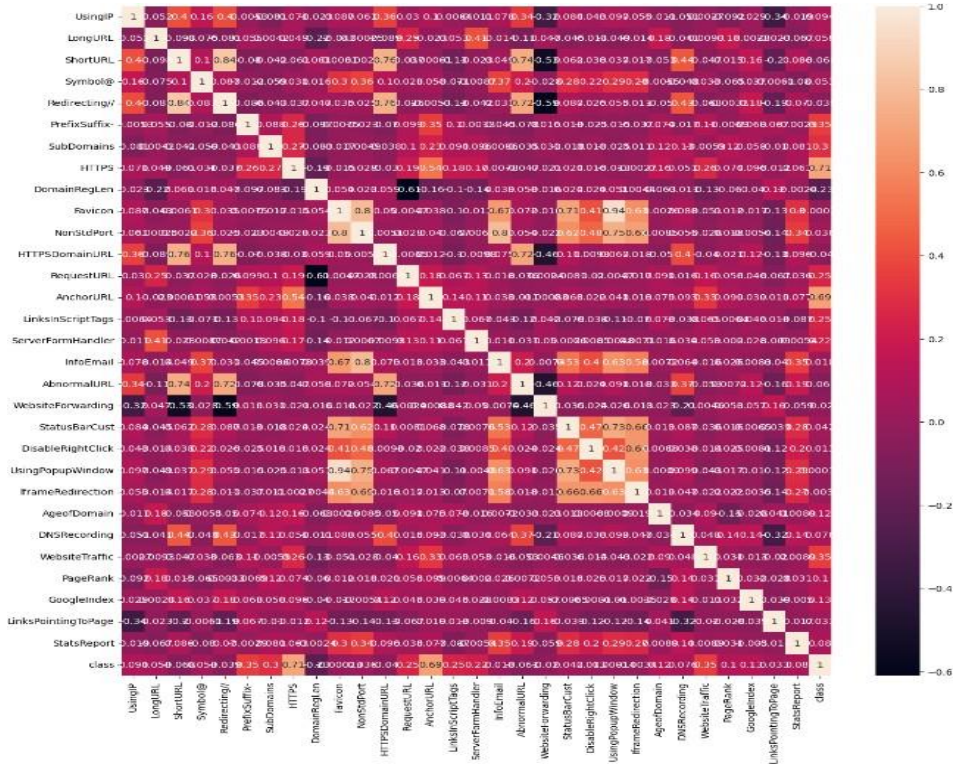

Figure 4: Data Visualization

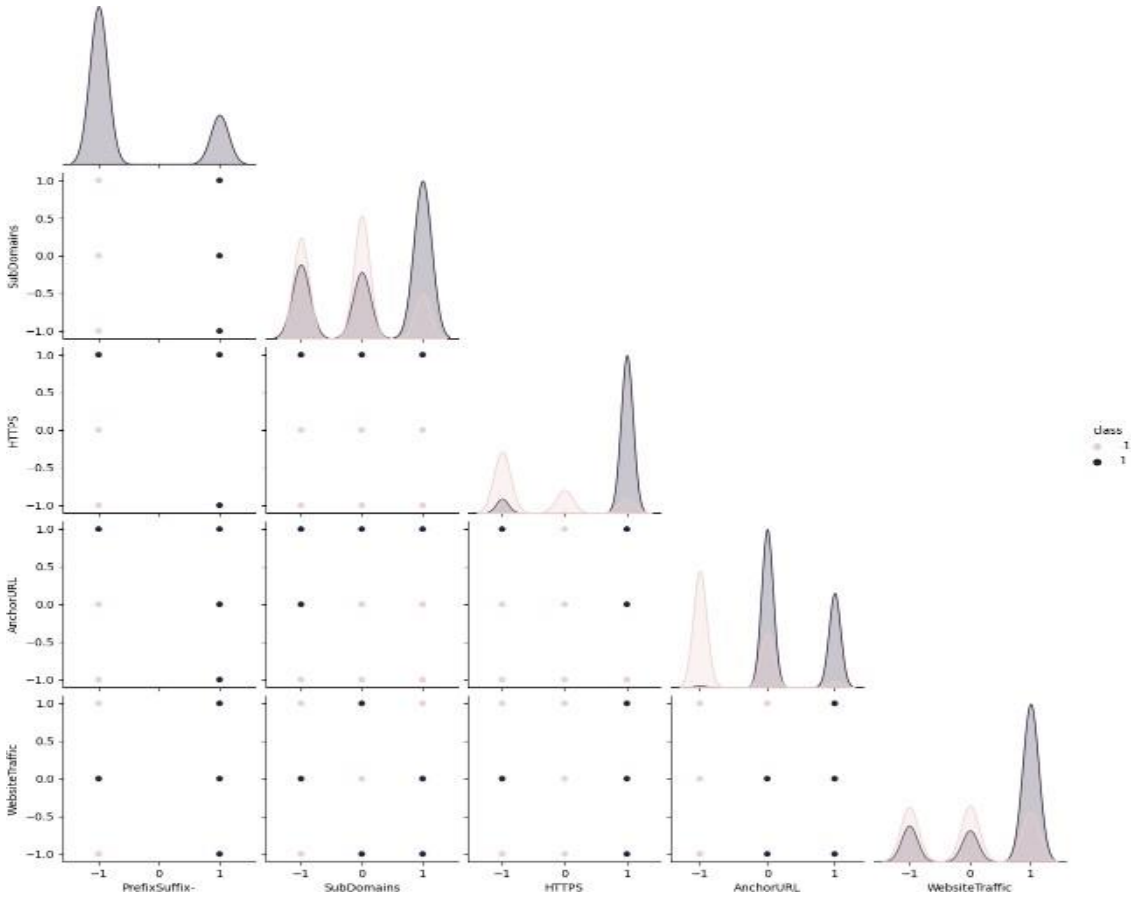

Figure 5: Features of the dataframe

5.1.3 Data Splitting

- The dataset is partitioned between training and testing sets using an 80-20 split ratio.
- By ensuring that the models are trained on a part of the data and then assessed on untested data, the generalization capacity of the models may be determined.

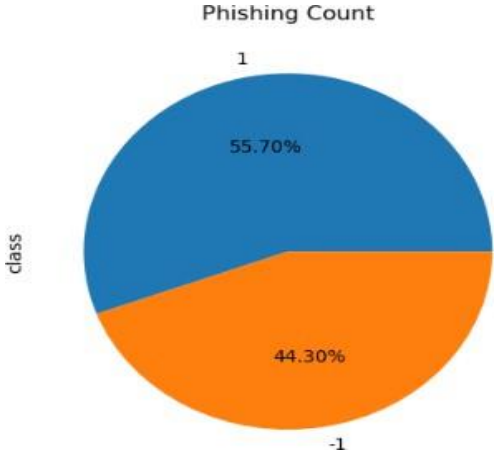

Figure 6. Phishing Count

5.1.4 Model Building and Training

Five different supervised machine learning methods are used to build models: Support Vector Classifier, Random Forest Classifier, Gradient Boosting Classifier and CatBoost Classifier. For every model, the subsequent actions are carried out:

- The model is initialized and trained with the training data.
- Target values are predicted for training and testing datasets. Evaluation of model performance measures on training and testing datasets, including as accuracy, F1-score, recall, and precision.
- For certain models, hyperparameter tweaking is done using methods such as GridSearchCV. The model's performance is evaluated comprehensively through a classification report.

5.1.5 Model Comparison

- A comparison data frame is created to assess each model's performance in terms of precision, recall, accuracy, and F1-score.
- The data frame is then arranged in order to identify the best-performing model using these measures.

5.1.6 Storing Best Model

- Re-instantiate and retraining of the Gradient Boosting Classifier, which was found to be the best-performing model, takes place.
- The model is then stored in an archive for further use.
- Combined, the technique follows an organized sequence that includes data loading, pre - processing, model assessment, and selection. This promotes consistency and resilience across the model building process.

5.2 Parameters

- URL Length: Indicates the number of characters in the domain name. Phishing URLs frequently have altered or extended domain names to seem like real ones. Secure-BankofAmerica-Login-Verify-Account-Information.com.
- Special Characters in URLs: Phishing URLs may use special characters such as hyphens or underscores to make them appear more authentic. Payp@l.com.
- IP Address Presence: Phishing URLs may use IP addresses rather than conventional domain names. 203.0.113.1/example.com/login is an example.
- HTTPS Presence: If a URL does not contain HTTPS, especially on websites that are supposed to be safe, this might be a sign of possible phishing. Using https://secure-bankofamerica-login.com as an example.
- URL Redirectors: Redirectors are a common tool used by phishing URLs to trick people. Using http://short.link/redirect with the URL http://malicious-site.com as an example.
- Subdomain Count: An excessive amount of subdomains is a common feature of phishing URLs. Using sub1.sub2.example.com as an example

5.3 Tools And Technologies

5.3.1 Python

Python is a prominent interpretive high-level programming language that is widely used in a wide range of applications. Python is well-liked by both inexperienced and seasoned developers because to its simplicity, readability, and flexibility. Its ease of usage is by far its greatest benefit. Python has a simple, user-friendly architecture with a syntax that puts emphasis on readability and simplicity. Python's natural simplicity makes it the perfect language for beginners learning programming, since it allows them to understand basic ideas without getting bogged down by complex grammar intricacies. Moreover, Python is incredibly versatile. Python's versatility across a wide range of fields, including web development, scientific computing, data analysis, artificial intelligence, and automation, confirms its reputation as the preferred language for complex tasks. With a strong developer community, Python has a rich ecosystem of libraries and tools that further expand its capabilities. Python stands out as a flexible language that can be used with a variety of coding styles because of its strong support for functional programming, object-oriented programming, and other programming paradigms. Its popularity is due to its strength and adaptability, which make it essential for a wide range of uses. Python is a top option for novices and experts alike because of its intuitive interface, wide range of features, and strong community support. Python reigns dominant, particularly in the arena of machine learning, providing a plethora of benefits that push progress in this sector.

- A vibrant and active developer community exists for Python, which is beneficial since it produces a wide range of tools and modules specifically designed for machine learning applications. Top Python machine learning libraries include Scikit-learn, PyTorch, TensorFlow, Keras, and Pandas.

- Python is known for its simple, easy-to-read syntax, which makes machine learning code generation and understanding easier. This feature reduces mistakes, speeds up development, and boosts productivity—especially when it comes to complex machine learning projects.
- Python's versatility is shown in its outstanding flexibility across a wide range of applications, including the ever-evolving field of machine learning. Python is able to work with a variety of programming paradigms, including object-oriented and functional programming. It can handle both organized and unstructured data quite well.
- Python's large community, simplicity, and flexibility make it the perfect tool for machine learning quick prototyping. Developers can quickly experiment with different methods and approaches because to its intuitive interface, which allows them to repeatedly refine their models until they provide the best results.

5.3.2 Libraries

- NumPy: Facilitates numerical computations within the Python environment.
- Pandas: Enables manipulation and analysis of data structures.
- Matplotlib: A versatile plotting library supporting static, interactive, and animated visualizations.
- Seaborn: Enhances Matplotlib's capabilities by creating visually appealing and informative statistical graphics.
- Scikit-learn (sklearn): Offers efficient tools for data mining and analysis within the Python machine learning ecosystem.
- CatBoost: Developed by Yandex, excels in handling categorical features within datasets.
- XGBoost: A renowned gradient boosting library known for its efficiency and scalability.
- Warnings: Used to manage warning messages effectively.

5.3.3 Flask

Flask is a lightweight and versatile web framework crafted for Python. Its primary aim is to facilitate rapid web application development with minimal coding efforts. Offering a suite of tools and libraries including URL routing, templates, and request/response management, Flask empowers developers to build applications swiftly and efficiently. Its simplicity makes it easy to grasp and a popular choice for small to medium-sized projects. Moreover, Flask offers developers the flexibility to cherry-pick components and structure their applications as they see fit. With a vibrant developer community, Flask boasts a wide array of extensions and plugins, enhancing its functionality with features like database integration, authentication, and form handling. Its lightweight design further boosts its appeal, ensuring efficient performance for web applications of varying scales.

Flask boasts a concise codebase and few dependencies, simplifying deployment and scalability. This feature has earned it popularity among developers aiming to craft efficient, uncomplicated web applications. With its focus on simplicity, Flask stands as a powerful and adaptable framework suited for small to medium-sized web projects. Its inherent flexibility and extensibility add to its allure, allowing

developers to rapidly create web applications with minimal code, while retaining the freedom to scale as required.

5.3.4 Jupyter Notebook

Jupyter Notebook is an open-source web application that enables users to create and share documents integrating live code, equations, visualizations, and narrative text. It is based on the Python interactive shell and offers a robust toolkit for data analysis and visualization. The notebook structure is organized into cells, accommodating code, text, or multimedia content, and supports various programming languages like Python, R, and Julia. Users can execute code cells interactively and view results directly within the notebook. Widely adopted by data scientists, researchers, and educators, Jupyter Notebook provides a flexible environment for data analysis and experimentation. Noteworthy features include support for data visualization libraries such as Matplotlib, Seaborn, and Bokeh, simplifying the creation of interactive plots and charts. Collaboration features facilitate real-time sharing and project collaboration, particularly beneficial for remote teams. Additionally, Jupyter Notebook aids reproducibility by allowing documentation of analysis and code, simplifying the verification process. Its versatility, interactivity, data visualization capabilities, collaborative functionalities, and support for reproducibility make it a favored tool among data professionals and educators.

6. Result and Discussion

Gradient boosting classifiers (GBC) are a set of machine learning algorithms that combine multiple weak learning models to form a powerful predictive model. Typically, decision trees are employed in gradient boosting. Boosting algorithms are essential for managing the bias-variance trade-off. Unlike bagging algorithms, which focus solely on reducing variance, boosting algorithms handle both bias and variance, making them more effective in model optimization.

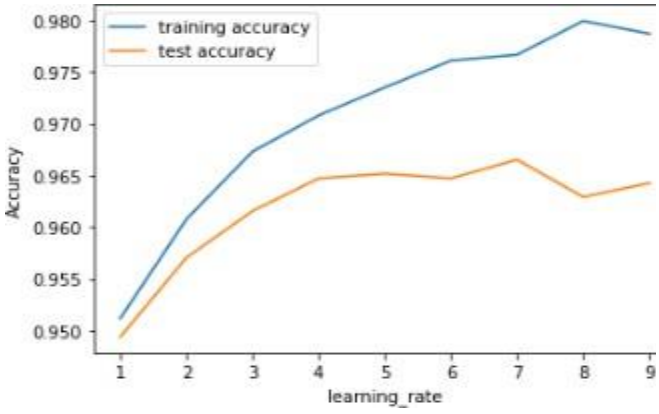

Figure 7. Learning rate and Accuracy in Gradient Boosting Classifier

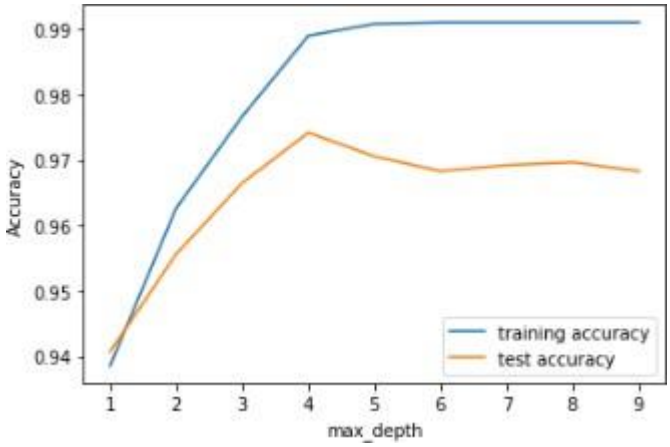

Figure 8. Max depth and Accuracy

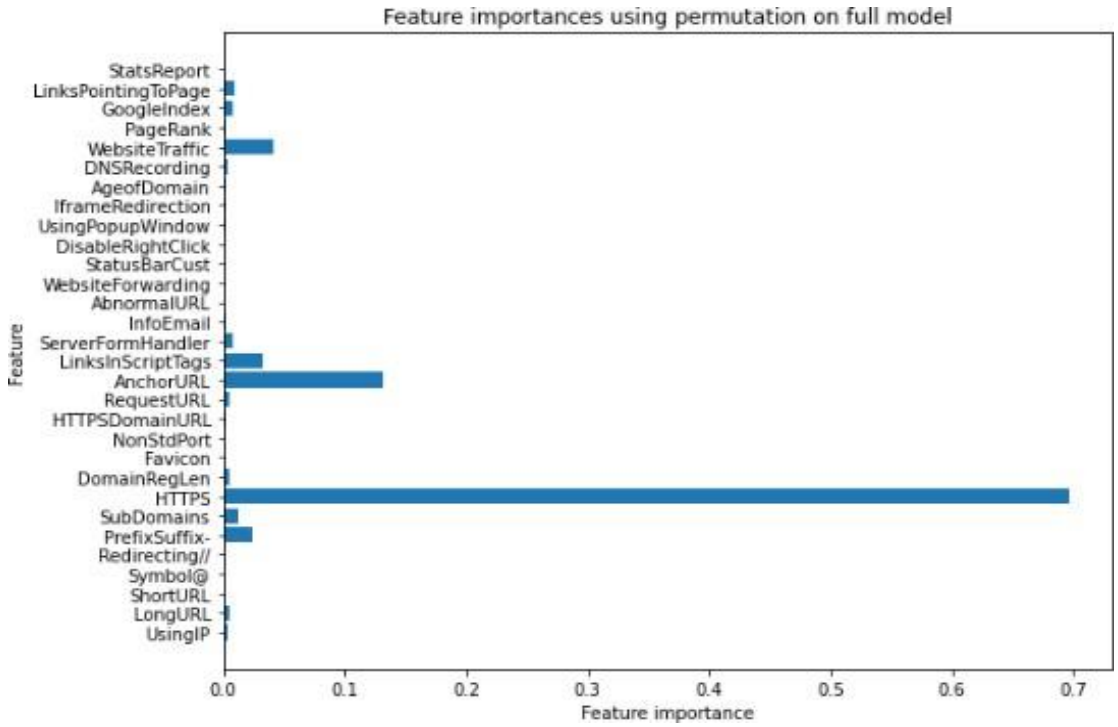

Figure 9. Features importance's using permutation on full model

6.1 Analysis

- Accuracy: The Gradient Boosting Classifier showed the best performance on both the training and test sets, achieving an accuracy of 98.9% on the training data and 97% on the test data.
- F1 Score: The Gradient Boosting Classifier also attained the highest F1 score on both the training and test sets, scoring 99% on the training data and 97% on the test data.
- Recall: Furthermore, the Gradient Boosting Classifier achieved the highest recall on both the training and test sets, with rates of 99.4% on the training data and 98% on the test data.
- Precision: Moreover, the Gradient Boosting Classifier demonstrated the highest precision on both the training and test sets, with precision rates of 98.9% on the training data and 96% on the test data.

6.2 Performance Metrics

Table 1: Performance metrics on Training data

Metrics / Methods	Precision	Recall	F-Measure	Accuracy
SVM	93.18182	89.1304	91.11111	89.61039
Random Forest	96.22642	94.4444	95.3271	93.50649
CatBoost Classifier	97.95918	94.1176	96.0000	94.80519
GBC	98.9000	99.4010	99.1410	98.9123

Table 2: Performance metrics on Testing data

Metrics / Methods	Precision	Recall	F-Measure	Accuracy
SVM	90.2510	87.4584	92.5484	86.6454
Random Forest	92.1450	92.4425	96.3158	91.5478
CatBoost Classifier	95.4745	92.5546	94.0080	92.7954
GBC	96.5498	98.5423	97.2345	97.1584

7. Conclusion

Phishing attacks have surged in recent years, posing grave risks to individuals and organizations alike. These schemes often involve fraudulent emails, texts, or websites designed to coax users into revealing sensitive information like passwords or credit card numbers. An effective countermeasure against such threats is the use of phishing URL detection tools. These tools employ advanced algorithms and machine learning to scrutinize website content and behavior, identifying suspicious patterns indicative of fraudulent activity. However, while these tools can be effective, they are not foolproof, as phishing tactics continuously evolve. Attackers constantly devise new methods to deceive users, underscoring the importance of vigilance and additional protective measures. Remaining cautious, avoiding suspicious links and emails, and implementing two-factor authentication are essential steps in safeguarding against phishing attacks. In conclusion, while phishing URL detection tools serve as valuable assets in identifying potentially harmful websites, maintaining awareness and adopting comprehensive protective strategies are imperative in combating phishing threats.

8. Future Directions

Continual advancement in detection algorithms is imperative due to the evolving nature of phishing attacks. This involves delving into sophisticated machine learning methods like deep learning to refine detection accuracy. Integrating various detection techniques, such as analyzing website content, behavior, and user interactions, expands the scope of phishing URL detection tools, enhancing their efficacy. Concurrently, investing in user education and awareness programs is vital to empower individuals in recognizing and thwarting phishing attempts. This includes developing educational resources, disseminating informative content, and fostering partnerships with organizations to promote cybersecurity best practices. Moreover, maintaining vigilant monitoring of phishing trends and updates ensures that detection tools remain responsive to emerging threats. Collaboration with industry partners, including cybersecurity firms and organizations, facilitates knowledge exchange and resource sharing to strengthen phishing detection capabilities collectively. Additionally, improving the user interface and experience of detection tools enhances accessibility and usability, contributing to overall effectiveness. Finally, ongoing evaluation and validation of detection tools through rigorous testing and real-world deployment scenarios uphold their reliability in defending against phishing attacks.

References

[1] Jain, A. K., & Gupta, B. B. (2022). A survey of phishing attack techniques, defence mechanisms and open research challenges. *Enterprise Information Systems*, *16*(4), 527-565.
[2] Ali, M. M., & Mohd Zaharon, N. F. (2024). Phishing—A cyber fraud: The types, implications and governance. *International Journal of Educational Reform*, *33*(1), 101-121.
[3] Schmitt, M., & Flechais, I. (2024). Digital Deception: Generative artificial intelligence in social engineering and phishing. *Artificial Intelligence Review*, *57*(12), 1-23.
[4] Tamal, M. A., Islam, M. K., Bhuiyan, T., Sattar, A., & Prince, N. U. (2024). Unveiling suspicious phishing attacks: enhancing detection with an optimal feature vectorization algorithm and supervised machine learning. *Frontiers in Computer Science*, *6*, 1428013.
[5] Butt, U. A., Amin, R., Aldabbas, H., Mohan, S., Alouffi, B., & Ahmadian, A. (2023). Cloud-based email phishing attack using machine and deep learning algorithm. *Complex & Intelligent Systems*, *9*(3), 3043-3070.
[6] Alauthman, M., Almomani, A., Alweshah, M., Omoush, W., & Alieyan, K. (2019). Machine learning for phishing detection and mitigation. In *Machine Learning for Computer and Cyber Security* (pp. 48-74). CRC Press.
[7] Apandi, S. H., Sallim, J., & Sidek, R. M. (2020, February). Types of anti-phishing solutions for phishing attack. In *IOP Conference Series: Materials Science and Engineering* (Vol. 769, No. 1, p. 012072). IOP Publishing.
[8] Dodiya, K. R., Varayogula, S. N., & Gohil, B. V. (2024). Rising Threats, Silent Battles: A Deep Dive Into Cybercrime, Terrorism, and Resilient Defenses. In *Cases on Forensic and Criminological Science for Criminal Detection and Avoidance* (pp. 123-150). IGI Global.
[9] Chen, Y. C., Ma, Y. W., & Chen, J. L. (2020, July). Intelligent malicious URL detection with feature analysis. In *2020 IEEE Symposium on Computers and Communications (ISCC)* (pp. 1-5). IEEE.
[10] Zhang, Y., Marshall, I., & Wallace, B. C. (2016, November). Rationale-augmented convolutional neural networks for text classification. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing. Conference on Empirical Methods in Natural Language Processing* (Vol. 2016, p. 795). NIH Public Access.
[11] Saxe, J., & Berlin, K. (2017). eXpose: A character-level convolutional neural network with embeddings for detecting malicious URLs, file paths and registry keys. *arXiv preprint arXiv:1702.08568*.
[12] Vinayakumar, R., Soman, K. P., & Poornachandran, P. (2018). Evaluating deep learning approaches to characterize and classify malicious URL's. *Journal of Intelligent & Fuzzy Systems*, *34*(3), 1333-1343.
[13] He, C., Kang, H., Yao, T., & Li, X. (2019). An effective classifier based on convolutional neural network and regularized extreme learning machine. *Mathematical biosciences and engineering*, *16*(6), 8309-8321.
[14] Jayapradha, J., Vineethkumar, S., Vigneshwaran, R., & Ramprasath, A. (2024). Intrusion Detection System For Phising Detection Using Convolution Neural Network. *Educational Administration: Theory and Practice*, *30*(5), 5565-5575.
[15] Abad, S., Gholamy, H., & Aslani, M. (2023). Classification of malicious URLs using machine learning. *Sensors*, *23*(18), 7760.
[16] Ma, J., Saul, L. K., Savage, S., & Voelker, G. M. (2009, June). Identifying suspicious URLs: an application of large-scale online learning. In *Proceedings of the 26th annual international conference on machine learning* (pp. 681-688).
[17] Vinayakumar, R., Soman, K. P., & Poornachandran, P. (2018). Evaluating deep learning approaches to characterize and classify malicious URL's. *Journal of Intelligent & Fuzzy Systems*, *34*(3), 1333-1343.
[18] Pradeepthi, C., & Maheswari, B. U. (2024). Network intrusion detection and prevention strategy with data encryption using hybrid detection classifier. *Multimedia Tools and Applications*, *83*(13), 40147-40178.
[19] Parekh, S., Parikh, D., Kotak, S., & Sankhe, S. (2018, April). A new method for detection of phishing websites: URL detection. In *2018 Second international conference on inventive communication and computational technologies (ICICCT)* (pp. 949-952). IEEE.
[20] Bhavani, P. A., Chalamala, M., Likhitha, P. S., & Sai, C. P. S. (2022). Phishing Websites Detection Using Machine Learning. *Madhumitha and Likhitha, Pinnam Sree and Sai, Chanda Pranav Sai, Phishing Websites Detection Using Machine Learning (September 2, 2022)*.
[21] Razaque, A., Frej, M. B. H., Sabyrov, D., Shaikhyn, A., Amsaad, F., & Oun, A. (2020, October). Detection of phishing websites using machine learning. In *2020 IEEE Cloud Summit* (pp. 103-107). IEEE.