IDENTIFICATION OF PHISHING URLS AND WEBSITES USING ML

Mr Sriram Parabrahma Chari[1], Dhritisree Chhabra[2], Gampa Shashi Preetham[3], Gopaluni Ketan[4], Hanumandla Pavani[5],

[2,3,4,5] Sreyas Institute of Engineering and Technology [1] Assistant Professor (AIML)

Abstract:

The internet has become an essential part of our lives. The rise of the internet also opens opportunities to various scams and malicious activities like phishing. Phishing attacks are the practice of sending fraudulent communications that appear to come from a reputable source. Phishers try to deceive their victims by social engineering and creating mockup websites to steal information such as account ID, username, password from individuals and organizations. The identification process encompasses various techniques, including scrutinizing URLs for misspellings, inspecting SSL certificates for secure connections, analyzing website design and content quality, and avoiding unsolicited emails or pop-up windows requesting personal data. Many methods have been proposed to detect phishing websites and URLS yet the phishers have evolved their methodology and have managed to escape from these detection methods. One of the most successful methods for detecting these malicious activities is Machine Learning. This is because most Phishing attacks have some common characteristics which can be identified by machine learning methods. These systems exhibit the capability to adapt to emerging phishing tactics and work efficiently. The proposed approach employs various comparative analyses to select the most efficient algorithm. The anticipated outcome is an effective and adaptable ML-based phishing URL detection system, contributing to the ongoing efforts in safeguarding users from cyber threats.

Keywords:

Phishing, Cybersquatting, Typosquatting, Gradient boosting algorithm, random forest classifier, CatBoost classifier

1. Introduction:

Phishing is the act of deceiving people by impersonation, a malicious cyber activity, represents a persistent and evolving threat in the digital landscape. In its essence, phishing involves the deceptive practice of luring individuals into divulging sensitive information, such as usernames,passwords, and financial details, by masquerading as a trustworthy entity. These deceptive attempts can occur through various channels, including emails, social media messages, and fraudulent websites. This is broadly classified into two types- Cybersquatting and Typosquatting. These practices involve the registration or use of domain names that are either similar to well-known trademarks or intentionally misspelled versions of popular websites to deceive users for various purposes, often with malicious intent. Cybersquatting: This is a malicious practice where phishers or scammers register their domain names such that they resemble famous organisations, brands or entities with the intention of profiting from the confusion. The objectives of this may be monetary gain, diverting web traffic and fraud. Businesses have to engage in legal battles to regain control of their online identity. Typosquatting: This is also called URL hijacking. In this variation of phishing, phishers register their domain names with intentionally wrong spellings hoping that the users won't notice. The objectives of this may be ad revenue, fraud and brand confusion.

To address this pressing issue, the application of Machine Learning has emerged as a groundbreaking solution for identifying phishing URLs and websites. Machine Learning, with its ability to analyze vast datasets and learn intricate patterns, offers a powerful means to stay one step ahead of cybercriminals. This approach enables the automatic detection of phishing attempts with a level of accuracy and efficiency that was previously unattainable. The current systems outline the machine learning algorithms used for phishing URL detection, including Decision Tree, Random Forest, Light GBM, Logistic Regression, and SVM. The features extracted from URLs are categorized into Address Bar-based Features and Domain-based Features. The systems use different but limited machine learning models to analyze the data. The systems use a dataset that isn't comprehensive enough to understand the breadth of areas of phishing attacks. There is little space for comparison as only few ML models are used.

The proposed system uses different machine learning algorithms like Gradient Boosting Classifier, CatBoost classifier, Random forest classifier, Multi-layer Perceptron, XGBoost Classifier and trains and tests them against the dataset to achieve their accuracy score and based on that, the most efficient model is identified. The most efficient model will be charted and then be used in the final interface. The advantages of the system are that It will provide different machine learning algorithms along their workings such as accuracy score, F1 score, recall and precision. It will provide an in depth analysis of feature importance which will help understandthe most common domains used in phishing. It will also provide training and testing comparison maps. The system will also represent various comparisons of different classifiers so that the algorithm with the best accuracy score can be chosen. In an advanced application of this project, an interface will be provided where a user can enter a URL or a website to check whether it's a begnin website or a phishing website.

2. LITERATURE SURVEY:

Öner et al.[1] introduce PhishAri—a novel system geared towards the real-time identification of phishing URLs on Twitter. Employing machine learning techniques, the system leverages supervised classification and feature engineering to achieve a commendable accuracy rate of approximately 94% in detecting malicious URLs. The authors address the growing concern of phishing attacks within the dynamic environment of social media, particularly Twitter, where the rapid dissemination of information can amplify the impact of such cyber threats. PhishAri's real-time capabilities make it a valuable tool in proactively identifying and mitigating phishing risks on Twitter, contributing to the ongoing efforts to enhance cybersecurity measures in the realm of social media platforms.

Mallika et al.[2] introduces a comprehensive framework designed for the detection of phishing websites. This innovative approach integrates web scraping and data mining techniques to enhance the efficacy of the detection process. The framework employs cutting-edge machine learning algorithms, contributing to an accuracy ranging from 86% to 95%. By leveraging these advanced techniques, the proposed framework demonstrates its robustness in identifying phishing threats online. The amalgamation of web scraping and data mining not only enriches the feature set for analysis but also facilitates a more nuanced understanding of the intricate patterns associated with phishing websites. This survey not only underscores the significance of integrating multiple technologies for effective detection but also positions the proposed framework as a reliable and advanced solution in the ongoing battle against the pervasive threat of online phishing.

Asadullah Saf et al.[3] serves as a comprehensive examination of various techniques employed in the detection of phishing websites. With a focus on precision, the survey meticulously evaluates the efficacy of multiple detection methods, delving into the nuanced analysis of false positives and false negatives. This critical examination underscores the importance of reliable detection mechanisms in the ever-evolving landscape of phishing attacks. The survey explores a diverse array of algorithms, ranging from traditional methods like decision trees and support vector machines to cutting-edge technologies such as neural networks. This extensive exploration provides a nuanced understanding of the strengths and limitations of different approaches. By prioritizing accuracy and leveraging a wide spectrum of algorithms, the survey contributes to the ongoing discourse on effective phishing detection, fostering advancements in cybersecurity measures.

Alnemari et al. [4] focus on comparatative analysis of different algorithms in the survey. It offers a thorough exploration of the literature surrounding the detection of phishing domains through the lens of machine learning. The surveyed studies reveal an impressive accuracy range of 90-96%, emphasizing a commitment to identifying the most reliable and effective models in the domain. The research delves into the application of various algorithms, including Decision Trees, Random Forest, Support Vector Machines, and Neural Networks, showcasing a diverse and comprehensive approach to phishing domain detection. This literature survey not only sheds light on the current state-of-the-art techniques but also critically examines the existing research, identifying potential limitations and gaps in the field. The paper proposes directions for future studies, emphasizing the need for continued innovation to address emerging challenges in the dynamic realm of phishing domain detection. A synthesis of the findings provides a comprehensive overview of the strategies employed, offering valuable insights into improving the accuracy and efficiency of machine learning-based systems

Mandadi et al. [5] delves into the realm of phishing detection by leveraging user-based and machine-based email header features. The research critically evaluates the effectiveness of these features and highlights their role in achieving remarkable accuracy rates, surpassing 95% in certain instances. The algorithms employed in this study include Random Forest and decision tree classifiers, demonstrating a focused approach to phishing detection. The evaluation of user-based and machine-based email header features, coupled with the application of Random Forest and decision tree classifiers, showcases a targeted and effective approach. The findings not only contribute to the current understanding of phishing detection but also pave the way for further research in leveraging email header features to bolster the accuracy of machine learning-based systems in countering phishing threats.

Shuibin et al. [6] present an innovative approach to detecting phishing websites with a reported accuracy of approximately 96%. The proposed framework introduces a novel methodology centered around structural similarity, utilizing graph-based features and machine learning techniques for effective classification. Unlike traditional methods, this framework leverages the inherent structural characteristics of phishing websites, providing a unique perspective in the realm of cybersecurity. By combining structural similarity analysis with advanced machine learning algorithms, the authors offer a robust and tailored solution for identifying phishing threats, contributing to the continual evolution of effective countermeasures against evolving cyber threats.

Adnan et al. [7] delve into the realm of phishing detection with a reported accuracy of approximately 95-96%. This research employs machine learning algorithms, specifically Random Forest and Support Vector Machine (SVM), to discern phishing websites based on various features, including URLs and website content. The authors' approach signifies a concerted effort to leverage advanced algorithms and multifaceted features in order to enhance the accuracy of phishing website detection. By incorporating machine learning techniques, the paper contributes to the ongoing discourse on bolstering cybersecurity measures, particularly in the identification and mitigation of phishing threats. The utilization of Random Forest and SVM algorithms, along with a focus on features such as URLs and website content, showcases a methodical and comprehensive strategy for detecting malicious online activities.

Kimpara et al. [8] serve as a valuable overview of diverse methodologies employed in the detection of phishing threats. While the paper does not specify a single accuracy figure, it provides a comprehensive survey of various phishing detection techniques, shedding light on their varying levels of efficacy. This survey encapsulates an extensive range of approaches, encompassing traditional blacklisting, heuristic methods, and modern machine learning-based strategies. By systematically reviewing these techniques, the authors offer insights into the strengths and weaknesses of each method, providing a nuanced understanding of the multifaceted landscape of phishing detection. The paper contributes to the knowledge base in cybersecurity by synthesizing information on a variety of approaches, empowering researchers and practitioners to make informed decisions regarding the implementation of phishing detection mechanisms based on their unique requirements and contextual considerations.

Emmanuel et al [9] present a focused exploration into the application of supervised machine learning algorithms for phishing detection, with a reported accuracy nearing 96%. The study employs algorithms such as Decision Trees and Naïve Bayes, leveraging URL and website content features as crucial inputs for the detection models. By concentrating on these specific features, the research provides a targeted and effective approach to identifying phishing websites. The utilization of supervised machine learning underscores the authors' commitment to precision in distinguishing between legitimate and malicious online entities. This paper contributes to the evolving landscape of cybersecurity by demonstrating the efficacy of supervised machine learning techniques and shedding light on the significance of features such as URLs and website content in the accurate detection of phishing threats.

Tummala et al. [10] showcase an effective system with a reported accuracy of 92% in distinguishing between genuine and fake websites. The research employs a combination of powerful machine learning algorithms, including Random Forest, Support Vector Machines (SVM), Decision Trees, and Gradient Boosting. This ensemble learning approach leverages the unique strengths of each algorithm to enhance the overall effectiveness of the fake website detection system. Random Forest excels in handling feature-rich data, SVM contributes to creating a robust decision boundary, Decision Trees provide interpretability, and Gradient Boosting further boosts the overall model performance. The integration of these algorithms demonstrates a holistic and synergistic strategy, underscoring the versatility and effectiveness of ensemble learning in the realm of fake website detection. This paper contributes to the ongoing discourse on leveraging machine learning for cybersecurity, particularly in the context of identifying and mitigating the risks associated with fraudulent online entities.

3. Proposed System: Phishing URLS Detection System

The proposed "Phishing URLs Detection System" is an innovative solution designed to combat the ever-growing threat of phishing websites and URLs. Phishing attacks, carried out through deceptive websites and URLs, continue to pose a significant risk to individuals, organizations, and online security. This proposal outlines the development of a comprehensive system that leverages advanced machine learning algorithms to detect phishing threats and empower users with a real-time, intelligent web interface.

3.1 Methodology:

3.1.1 Machine Learning-Based Detection: Employ a range of machine learning algorithms, including Gradient Boosting Classifier, CatBoost Classifier, Random Forest Classifier, Multi-layer Perceptron, XGBoost Classifier, K-Nearest Neighbors (KNN), Logistic Regression, Decision Trees, and Support Vector Machines (SVMs), to identify and classify phishing websites and URLs.

3.1.2 Model Evaluation: Train and test these machine learning models against a comprehensive dataset to assess their performance in terms of accuracy, F1 score, precision, and recall.

3.1.3 Identification of Most Efficient Model: Based on the evaluation results, identify the machine learning model with the highest accuracy score, indicating its efficiency in phishing detection.

3.1.4 Integration with Web Interface: Integrate the most efficient machine learning model with a user-friendly web interface. Users will input URLs or visit websites, and the system will provide a safety score to determine if the site is safe for use.

3.1.5 Identification of Common Phishing Features: Analyze the most common features used by phishing websites to better understand their characteristics and tactics.

3.2 Dataset:

The dataset used in this project was taken from kaggle. It has 11000+ URLs and 30 features for each record. The features are as follows:

Index, UsingIP, LongURL, ShortURL, Symbol@,Redirecting//, PrefixSuffix-, SubDomains, HTTPS, DomainRegLen, Favicon, NonStdPort, HTTPSDomainURL, RequestURL, AnchorURL, LinksInScriptTags, ServerFormHandler, InfoEmail, AbnormalURL, WebsiteForwarding, StatusBarCust, DisableRightClick, UsingPopupWindow, IframeRedirection, AgeofDomain, DNSRecording, WebsiteTraffic, PageRank, GoogleIndex, LinksPointingToPage, StatsReport, class

3.3 System Architecture:

When a user sends a URL request to check whether the URL entered if a phishing website or not the first step is that it extracts different features from the entered URL and enters 1 or –1 based on the conditions. This list with the scores of features extracted is now evaluated by the model stored based on highest accuracy and F1 score and the model classifies whether the website is a phishing one or a benign one. Fig-1 depicts the system architecture

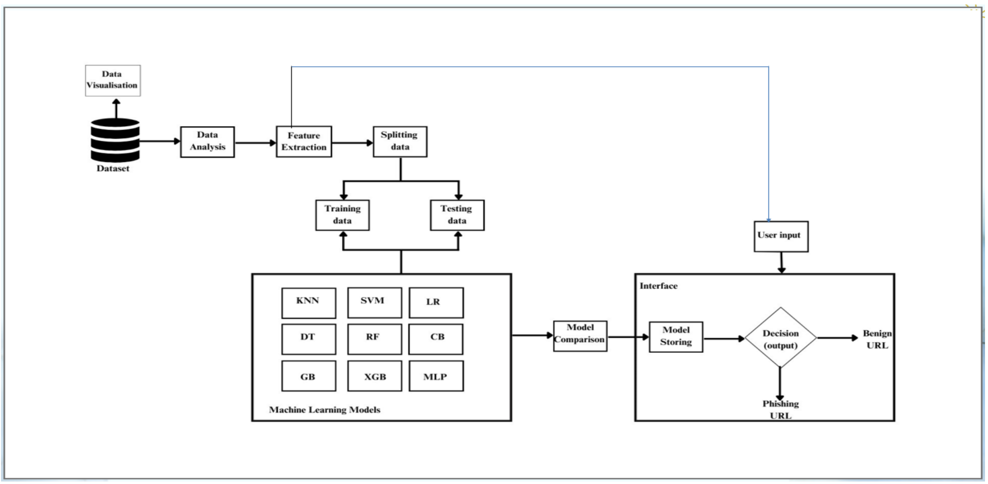


Fig 1: System Architecture

3.4 Components:

This project can be divided into 5 components. Each module has it's own set of functions and performs tasks based on it.

3.4.1 Data Collection:

Collect a dataset of URLs for training and testing the ML model. Data preprocessing to clean and format the collected data. Data storage for easy access during model training. This dataset contains a collection of website URLs for over 11,000 websites. Each sample in the dataset is described by 30 website parameters, and each sample is associated with a class label that identifies it as either a phishing website (labeled as 1) or a legitimate website (labeled as 0). EDA helps us understand the data's structure and distribution. Fig 2 depicts the pie graph of the phishing count.

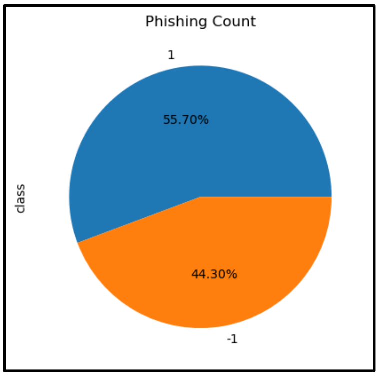
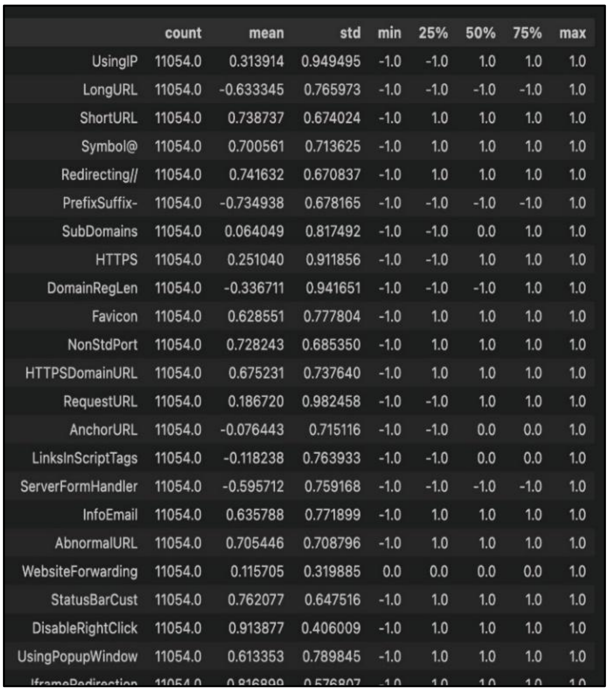


Fig 2 : phishing count

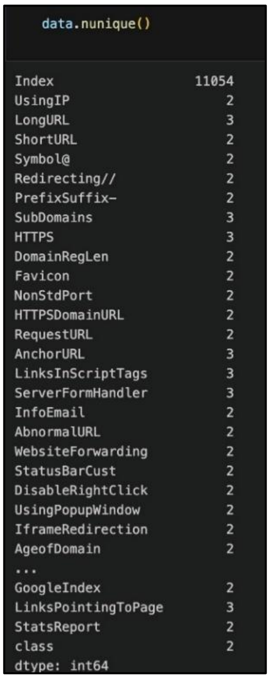
Here are some of the techniques used:

1. shape() to determine the number of rows and columns in the dataset.
2. columns() to list the columns in the dataset.
3. unique() to find the number of unique values in a particular column.

4. describe() to obtain descriptive statistics about the dataset, including count, mean, standard deviation, and minimum value.



	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



```
data.nunique()

Index                11054
UsingIP                  2
LongURL                  3
ShortURL                 2
Symbol@                  2
Redirecting//            2
PrefixSuffix-            2
SubDomains               3
HTTPS                    3
DomainRegLen             2
Favicon                  2
NonStdPort               2
HTTPSDomainURL           2
RequestURL               2
AnchorURL                3
LinksInScriptTags        3
ServerFormHandler        3
InfoEmail                2
AbnormalURL              2
WebsiteForwarding        2
StatusBarCust            2
DisableRightClick        2
UsingPopupWindow         2
IframeRedirection        2
AgeofDomain              2
...
GoogleIndex              2
LinksPointingToPage      3
StatsReport              2
class                    2
dtype: int64
```

Fig 3 and Fig 4: EDA

Fig 3 and fig 4 represent the results of the exploratory data analysis conducted on the dataset. Fig-3 provides a summary of descriptive statistics, offering insights into the central tendency, dispersion, and shape of the distribution of a dataset. Fig-4 counts the distinct values present in a column or Series.

The visualization in fig 5 is a correlation heatmap that shows how different features in the dataset are correlated. This heatmap provides insights into feature dependencies. By creating a heatmap with annotations that display the correlation values between different features. This visualization helps identify which features are closely related and may influence each other.

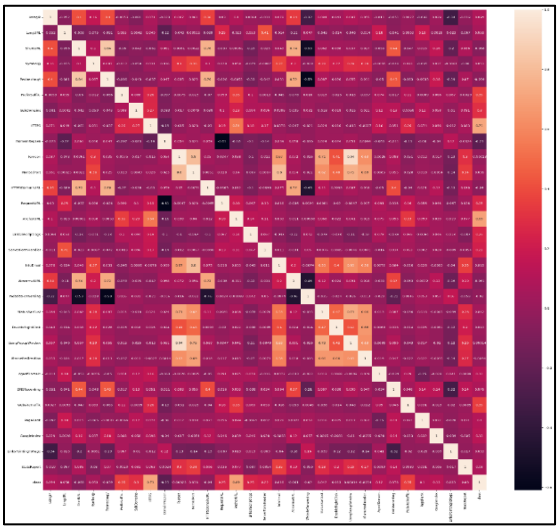


Fig 5: Correlation Matrix

3.4.2 Model Selection and Training:

Purpose: Choose an appropriate ML algorithm, train the model, and fine-tune hyper-parameters.
Algorithm selection (e.g., Decision Trees, Random Forest, Gradient Boosting, Logistic Regression).
Model training on the labeled dataset. The following supervised machine learning models are considered for on the dataset:

1. Logistic Regression
2. k-Nearest Neighbors
3. Support Vector Classifier
4. Naive Bayes
5. Decision Tree
6. Random Forest
7. Gradient Boosting
8. Catboost
9. XGBoost
10. Multilayer Perceptrons

3.4.3 Model Evaluation and storing:

Purpose: Assess the model's accuracy and effectiveness in identifying phishing URLs. Evaluation metrics (e.g., accuracy, precision, recall, F1-score). Visualization of results for easy interpretation.
Model Storing: To ensure that the trained model can be used later without retraining, it is serialized and saved to a file using the pickle library
The model.pkl file contains the serialized model, making it easy to reload and use in future applications without the need for retraining.

3.4.4 Feature Extraction:

Purpose: Extract relevant features from URLs that can be used as input for the ML model. URL parsing to separate components like domain, path, and query. Feature engineering is used to define the exact attributes for the model to work on. A feature importance analysis is also done to understand which feature are widely used to create confusion and scam under phishing.
In this part of the section, we explore the importance of different features in the trained model. Feature importance helps us understand which features contributed most to the model's predictions. We visualize this using a bar plot in fig 6.

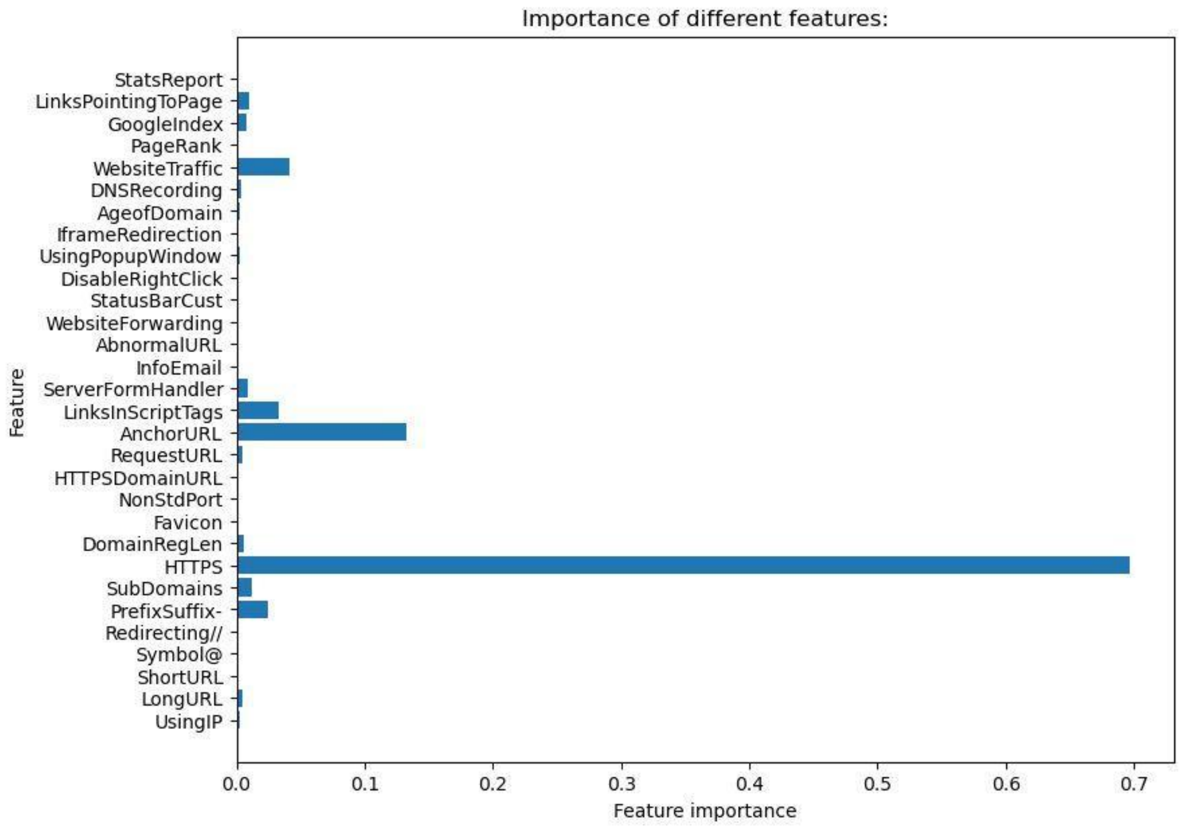
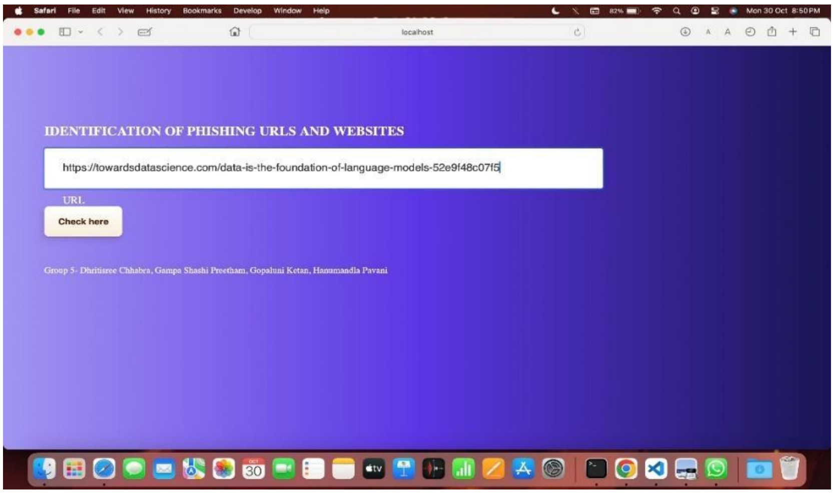


Fig 6: Feature importance

3.4.5 User Interface:

Create a user-friendly interface for users to interact with the phishing detection system. Designing a web-based dashboard or desktop application. Fig 7 represents the UI.



IDENTIFICATION OF PHISHING URLS AND WEBSITES

https://towardsdatascience.com/data-is-the-foundation-of-language-models-52e9f48c07f5

URL

Group 5- Dhritisree Chhabra, Gampa Shashi Preetham, Gopaluni Ketan, Hanumandla Pavani

Fig 7: User Interface

4. Results:

Model Evaluation: The models displayed the following metrics when tested against the dataset. The table-1 below depicts the accuracy, F1 score, recall and precision score against the dataset

ML model	Accuracy	F1 score	Recall	Precision
Gradient Boosting Classifier	0.974	0.977	0.994	0.986
CatBoost Classifier	0.972	0.975	0.994	0.989
XGBoost Classifier	0.969	0.973	0.993	0.984
Random Forest	0.967	0.971	0.991	0.991
Multi-layer Perceptron	0.965	0.969	0.996	0.979
Support Vector Machine	0.964	0.968	0.980	0.965
Decision Tree	0.960	0.964	0.991	0.993
K-Nearest Neighbors	0.956	0.961	0.991	0.989
Logistic Regression	0.934	0.941	0.943	0.927
Naive Bayes Classifier	0.605	0.454	0.292	0.997

Table-1: Metrics

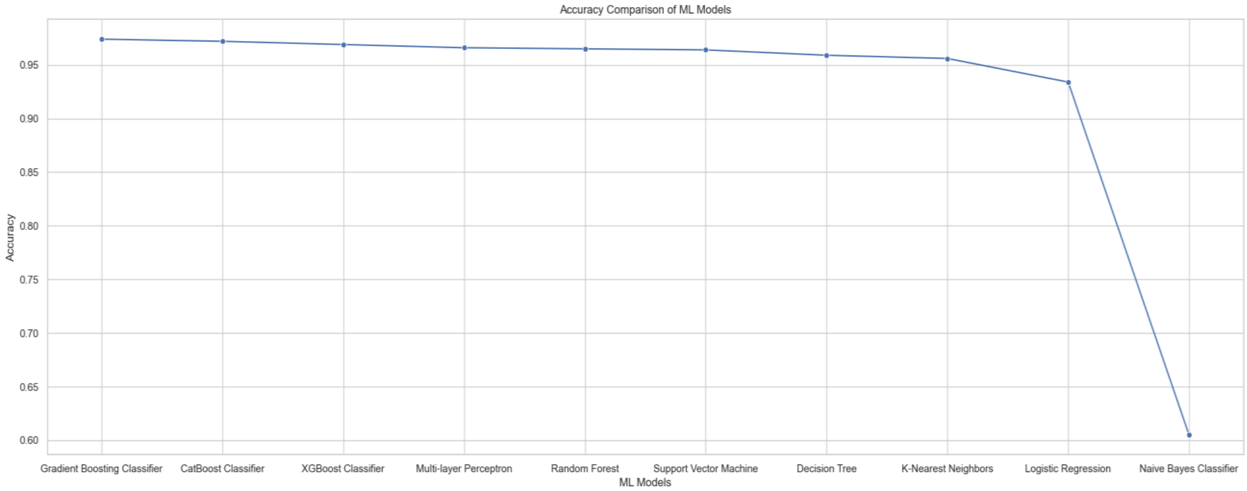


Fig 8: Accuracy Comparsion of ML modules

CatBoost is a recently open-sourced machine learning algorithm from Yandex. It can easily integrate with deep learning frameworks like Google's TensorFlow and Apple's Core ML. It can work with diverse data types to help solve a wide range of problems that businesses face today. The CatBoost Classifier demonstrated exceptional accuracy, F1 scores, and recall on the given classification task. With high precision and a strong ability to capture positive cases, CatBoost is an excellent choice for various classification problems. MLP Classifier stands for Multi-layer Perceptron classifier which in the name itself connects to a Neural Network. Unlike other classification algorithms such as Support Vectors or Naive Bayes Classifier, MLP Classifier relies on an underlying Neural Network to perform the task of classification. The Multi-layer Perceptron (MLP) Classifier demonstrated excellent accuracy and F1 scores on the given classification task. It achieved high recall and precision, making it a strong candidate for classification problems. Gradient boosting classifiers are a group of machine learning algorithms that combine many weak learning models together to create a strong predictive model. Decision trees are usually used when doing gradient boosting. Boosting algorithms play a crucial role in dealing with bias variance trade-off. Unlike bagging algorithms, which only controls for high variance in a model, boosting controls both the aspects (bias & variance), and is more effective. The gradient boosting algorithm is stored and used as it exhibits highest accuracy and F1 score. Fig 8 displays a comparison graph for all the algorithms employed in the comparative analysis. Fig 9 represents the accuracy of training and testing data of gradient boosting algorithm.

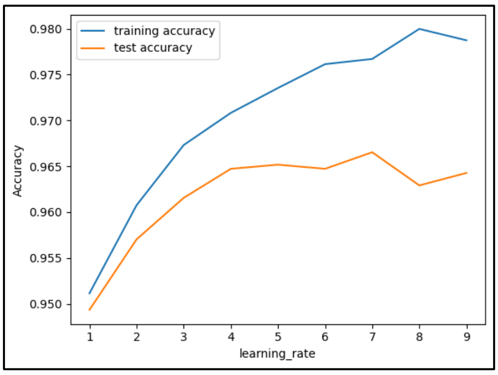


Fig 9: Accuracy of Training and Testing Data

5. Conclusion:

This project has been an exploration, a comparitive analysis of diverse machine learning models and an in-depth analysis of a phishing dataset to enhance our understanding of the features influencing the detection of begnin and phishing URLs. A wealth of knowledge has been gained, shedding light on the critical factors that contribute to a model's capability to discern the safety status of a given URL.

The process of performing Exploratory Data Analysis on the phishing dataset has not only provided insights into the dataset's characteristics but has also facilitated the identification of key features that significantly impact the models' ability to classify URLs as safe or potentially phishing/malicious. Among these features, "HTTPS," "AnchorURL," and "WebsiteTraffic" have emerged as crucial contributors to the classification process, underscoring their importance in distinguishing between legitimate and phishing URLs. Of particular note is the notable performance of the Gradient Boosting Classifier, which has exhibited an impressive accuracy rate of 97.4% in correctly classifying URLs into their respective safety categories. This high level of accuracy is a testament to the efficacy of the model in reducing the risk associated with malicious attachments and phishing threats. The Gradient Boosting Classifier's robust classification underscores its potential as a powerful tool in bolstering cybersecurity measures.

In conclusion, the journey through machine learning model exploration and EDA has enriched our understanding of phishing URL detection, emphasizing the pivotal role of specific features and showcasing the potential of advanced classifiers like Gradient Boosting in fortifying our defenses against malicious online activities. The insights gained will undoubtedly contribute to the ongoing discourse on cybersecurity and inspire further advancements in the field.

REFERENCES

[1]Abdelhamid, N., Ayesh, A., Thabtah, F.: Phishing detection based associative classification data mining. Expert Syst. Appl. 41(13), 5948–5959 (2014). https://doi.org/10.1016/j.eswa.2014.03.019

[2] Mallika Boyapati, Ramazan Aygun: "A comparison of machine learning techniques for phishing detection" (2017) https://ieeexplore.ieee.org/document/8117212

[3] Asadullah Saf, Satwinder Singh: "A systematic literature review on phishing website detection techniques"(2022) https://www.sciencedirect.com/science/article/pii/S1319157823000034

[4] Shouq Alnemari, Majid Alshammari: "Detecting Phishing Domains Using Machine Learning: A Literature Survey"(2023) https://www.mdpi.com/2076-3417/13/8/4649

[5] Adarsh Mandadi; Saikiran Boppana; Vishnu Ravella; R Kavitha: "Phishing Website Detection Using Machine Learning " (2022) https://ieeexplore.ieee.org/document/9824801

[6] Shuibin Lu, Hongyong Yuan, and Zhiyun Ren, "Phishing website detection using machine learning algorithms", International Journal of Computer Applications(0975-8887), vol. 181, no. 23 (2018)

[7] Md Nasim Adnan, Mohiuddin Ahmed, and Adamu Abubakar, "Phishing website classification and detection using machine learning", International Conference on Computer Communication and Informatics(ICCSIT) (2020)

[8] Dhamma Kimpara and Koji Nakao, "Detection of phishing websites by using machine learning-based URL analysis", 11nth International Conference on Computing, Communication and Networking Technologies(ICCCNT) (2020)

[9] Emmanuel O. Ogu and Abdullah Al Hasib, "Phishing attacks detection using machine learning approach", 3rd International Conference on Smart Systems and Inventive Technology(ICCSIT) (2020)

[10] W. B. Hu and S. K. Tummala, "Intelligent phishing website detection using Random Forest classifier", International Conference on Electrical and Computing Technologies and Applications(ICECTA) (2017)

[11] Fatima Salahdine, Zakaria El Mrabet, Naima Kaabouch, "Phishing Attacks Detection
A Machine Learning-Based Approach", University of North Dakota, (2022)

[12]Abu-Nimeh, S., Nappa, D., Wang, X., Nair, S.: "A comparison of machine learning techniques for phishing detection. In: Proceedings of the Anti-Phishing Working Group eCrime" Researchers Summit, pp. 60–69. ACM, New York (2007). https://doi.org/10.1145/1299015.1299021

[13]Aburrous, M., Hossain, M.A., Dahal, K.: "Experimental case studies for investigating e-banking phishing techniques and attack strategies". Cogn. Comput. 2(3), 242–253 (2010). https://doi.org/10.1007/s12559-010-9042-7

[14]Alkhozae, M.G., Batarfi, O.A.: "Phishing websites detection based on phishing characteristics in the webpage source code". Int. J. Inf. Commun. Technol. Res. 1(9), 238–291 (2011)

[15]Barraclough, P., Hossain, M., Tahir, M., Sexton, G., Aslam, N.:" Intelligent phishing detection and protection scheme for online transactions". Expert Syst. Appl. 40(11), 4697–4706 (2013). https://doi.org/10.1016/j.eswa.2013.02.009