iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 12 **Issue:** VI **Month of publication:** June 2024

DOI: https://doi.org/10.22214/ijraset.2024.63459

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue VI June 2024- Available at www.ijraset.com

SafeLink Analyzer Using Machine Learning

Prachi Tyagi[1], Sharad Pratap Singh[2], Kajal Kumari[3], Manali Patidar[4]
[1, 2, 3, 4] *Student B.Tech (CSE), Quantum University, Roorkee, India*

Abstract: Phishing remains a significant cyber threat that deceives individuals into accessing fraudulent websites, aiming to steal personal information. This study employs the Multinomial Naive Bayes and Logistic Regression machine learning algorithms to detect phishing URLs. We developed a web application using FastAPI and Python, providing users with a tool to verify the authenticity of URLs. Our findings indicate that these algorithms can effectively distinguish between phishing and legitimate URLs, thereby enhancing cybersecurity.
Keywords: Phishing, Cybersecurity, Multinomial Naive Bayes, Logistic Regression

I. INTRODUCTION

The digital landscape is increasingly fraught with sophisticated phishing attacks that pose severe risks to internet users, compromising personal data such as passwords, usernames, and financial information. Traditional security measures often fall short in combating these ever-evolving threats, necessitating more advanced and adaptive solutions. This research introduces the Safe Link Analyzer, a machine learning-based system designed to identify phishing URLs with high accuracy. By leveraging machine learning algorithms such as Multinomial Naive Bayes and Logistic Regression, the system can analyse URL patterns and features to discern legitimate URLs from phishing attempts.

This study presents the Safe Link Analyzer, which employs machine learning algorithms to analyse patterns and features of URLs, enabling accurate and timely identification of phishing attempts. Python and FastAPI were chosen for developing the application due to their flexibility and efficiency in creating scalable web services. The primary goal of this research is to demonstrate the effectiveness of machine learning algorithms in detecting phishing URLs and to showcase the practical application of these methods in a real-world setting.

The Safe Link Analyzer aims to significantly enhance overall cybersecurity by providing a reliable tool for detecting and preventing phishing attacks. Through comprehensive testing and evaluation, this study seeks to validate the efficacy of machine learning models in improving digital security and protecting users from phishing threats.

II. LITERATURE REVIEW

Numerous studies have explored the application of machine learning techniques in detecting phishing websites. This section provides an overview of the most relevant research in this field.

Mahajan and Siddavatam (2018) developed a phishing website detection system using various machine learning algorithms, highlighting the effectiveness of these methods in identifying phishing URLs. Their research underscores the importance of feature extraction and model selection in achieving high detection accuracy[1] .

Tyagi et al. (2023) introduced a next-generation phishing detection and prevention system utilising machine learning. Their approach combined multiple algorithms to enhance detection rates and reduce false positives, demonstrating the potential of hybrid models in cybersecurity[2] .

Chawla and Kohli (2022) focused on using artificial intelligence for phishing site detection. Their study emphasised the role of deep learning techniques and advanced feature extraction methods in improving the precision of phishing detection systems[3] .

Dutta (2021) conducted a comparative analysis of various machine learning techniques for detecting phishing websites. The study found that while traditional algorithms like logistic regression and naive Bayes are effective, ensemble methods such as random forests offer superior performance in terms of accuracy and false positive rates[4] .

Mandadi et al. (2022) explored the application of machine learning in phishing website detection, comparing the effectiveness of different algorithms. Their findings support the use of algorithms that handle text data efficiently, such as Multinomial Naive Bayes, for URL-based phishing detection .

Alrefaai et al. (2022) discussed the challenges of deploying machine learning models for phishing detection in real-world scenarios. They highlighted the importance of scalability, real-time processing, and user interface design in developing practical cybersecurity tools [5].

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue VI June 2024- Available at www.ijraset.com

The reviewed literature indicates that machine learning techniques are highly effective in detecting phishing websites. The choice of algorithm, feature extraction methods, and the combination of different models play crucial roles in the overall performance of these systems. This project builds on these insights by employing Multinomial Naive Bayes and Logistic Regression algorithms, leveraging their strengths in handling text data and binary classification tasks.

III. METHODOLOGY

A. *Data Collection*

A critical component of this project is the dataset, which comprises approximately 549,346 unique entries. Each entry includes a URL and a corresponding label indicating whether the URL is 'Good' (secure) or 'Bad' (phishing). This dataset is essential for training and evaluating the machine learning models.

B. *Feature Extraction*

To represent the URLs effectively, several key features were extracted:

1) *Term Frequency:* The frequency of specific words or characters within the URL.
2) *Existence of Keywords:* The presence of keywords commonly associated with phishing.
3) *URL Length:* The overall length of the URL
4) *Subdomain Count:* The number of subdomains within the URL.

IV. IMPLEMENTATION

The implementation process involves several steps and uses a variety of machine learning libraries and Python tools:

1) *Data Collection:* Compiling an extensive collection of URLs with labels
2) *Preprocessing:* Vectorizing text, stemming words, and tokenizing URLs.
3) *Model Building:* Training machine learning models using the preprocessed data
4) *Model Evaluation:* Assessing model performance using classification reports and confusion matrices.
5) *Deployment:* Implementing the model for real-time phishing detection using Uvicorn and FastAPI.

V. ALGORITHMS

A. *Logistic Regression*

Logistic regression is used for classification tasks where the dependent variable is categorical. This algorithm models the relationship between a binary dependent variable and one or more independent variables by estimating probabilities using a logistic function. It is particularly suited for binary classification problems like distinguishing between phishing and safe URLs. Logistic regression was chosen for its simplicity and effectiveness in binary classification tasks, providing a robust foundation for our model.

B. *Multinomial Naive Bayes*

Multinomial Naive Bayes (MNB) is suitable for discrete feature classification tasks, such as text categorization, where features represent counts or frequencies. This algorithm uses Bayes' theorem, assuming feature independence, and focuses on feature frequency within each class. MNB is well-suited for our project, where the primary features are derived from the textual content of URLs.

VI. MODEL TRAINING METHODOLOGY

The dataset is divided into training and testing sets using the train_test_split function to evaluate the model on unseen data. The training process involves building two models:

1) *Logistic Regression:* Used to classify URLs as phishing or authentic based on the extracted features
2) *Multinomial Naive Bayes:* Utilised for its efficiency with text data and ability to handle features derived from URLs.
3) *Libraries Used*

a) *Uvicorn and FastAPI:* For deploying the application and building the front-end GUI.
b) *Numpy and Pandas:* For numerical operations and data manipulation.
c) *Scikit-learn:* For training and evaluating the machine learning models.

VII. MODEL EVALUATION METRICS

The models' performance is evaluated using a range of metrics.

1) *Accuracy:* The percentage of correctly classified instances out of all examples.
2) *Precision, Recall, and F1-score:* Performance metrics derived from the classification reports, indicating true positives, false positives, and false negatives.
3) *Confusion Matrix:* A visual representation showing true positive, true negative, false positive, and false negative values.

VIII. RESULTS AND DISCUSSION

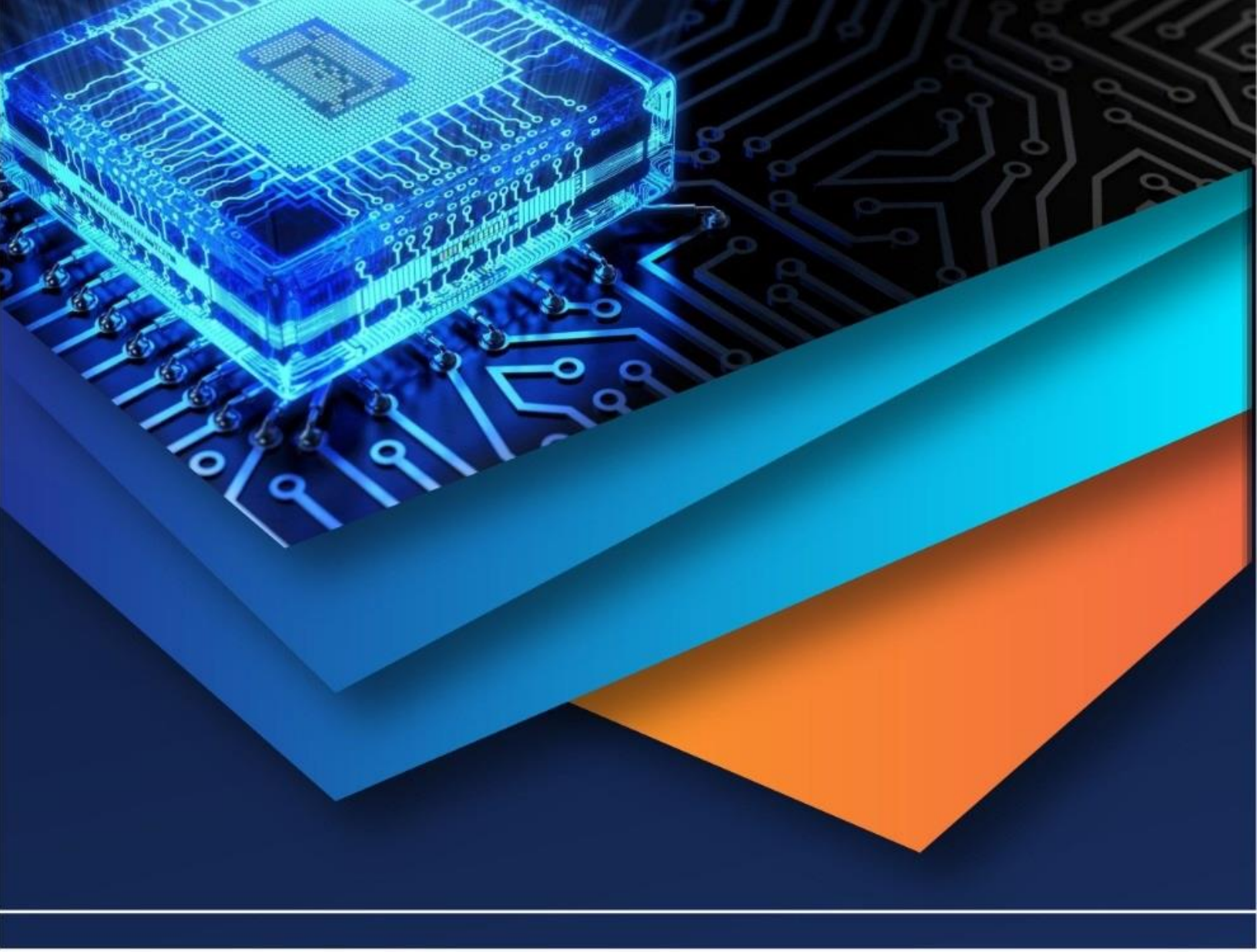
The research indicates that both Multinomial Naive Bayes (MNB) and Logistic Regression (LR) perform well in identifying phishing URLs. MNB, noted for its simplicity and efficiency with large datasets, achieved competitive precision, recall, and F1-score values. LR, leveraging its robust probabilistic foundation, also demonstrated strong performance in classification accuracy and probability estimation. Visualisation tools such as internal link analysis and word clouds provided additional insights into phishing patterns, complementing the quantitative evaluations. Overall, the study suggests that MNB and LR are effective choices for phishing URL detection, each offering distinct advantages in terms of computational efficiency and probabilistic modelling, while visualisation tools enhance the interpretability of findings and aid in understanding phishing behaviours.

IX. CONCLUSION

The system for detecting phishing websites has been successfully implemented and meets the project's objectives. This tool is especially beneficial for frequent internet users, allowing them to verify the safety of links before clicking. The project achieved a detection accuracy of 97.14% using the Random Forest algorithm, highlighting that more training data improves classifier performance. Future research will focus on enhancing detection precision through a hybrid approach combining machine learning and blacklist techniques.

REFERENCES

[1] Mahajan, R., & Siddavatam, I. (2018). Phishing Website Detection using Machine Learning Algorithms. International Journal of Computer Applications. https://doi.org/10.5120/ijca2018918026
[2] Tyagi, S., Tyagi, D., Dutta, P., & Dubey, D. (2023). Machine Learning is used to create the next generation of phishing detection and prevention systems. https://doi.org/10.1109/ICAISC56366.2023.10085529
[3] Chawla, A., & Kohli, S. (2022). Phishing Site Detection Using Artificial Intelligence. https://doi.org/10.1007/978-981-19-5037-7_48
[4] Dutta, A. K. (2021). Detecting Phishing Websites using Machine Learning Technique. PLoS One. https://doi.org/10.1371/journal.pone.0258361
[5] Alrefaai, S., Özdemir, G., & Mohamed, A. (2022). Detecting Phishing Websites Using Machine Learning. 2022 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA), Ankara, Turkey.https://doi.org/10.1109/HORA55278.2022.9799917
[6] Linear Regression in Machine Learning. Javatpoint. https://www.javatpoint.com/linear-regression-in-machine-learning
[7] Multinomial Naive Bayes in Machine Learning. The Clever Programmer. https://thecleverprogrammer.com/2021/08/06/multinomial-naive-bayes-in-machine-learning/
[8] Deploying ML Models as API Using FastAPI. GeeksforGeeks. https://www.geeksforgeeks.org/deploying-ml-models-as-api-using-fastapi/
[9] How to Create a Machine Learning App Using FastAPI and Deploying it to the Kubernetes Cluster. Section.io. https://www.section.io/engineering-education/how-to-create-a-machine-learning-app-using-the-fastapi-and-deploying-it-to-the-kubernetes-cluster/



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY