iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 12 **Issue:** V **Month of publication:** May 2024

DOI: https://doi.org/10.22214/ijraset.2024.61807

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 V May 2024- Available at www.ijraset.com

Phishing URL Detection Using XGBoost

Abin Jose[1], Alfred Shyjo I[2], Vivek A V[3], Harikrishna Jayakumar[4], Sachin K Tomy[5]
Department of Computer Science and Engineering, Mar Athanasius College of Engineering, Kothamangalam, Kerala

Abstract: ***Phishing attacks are a major threat to cybersecurity, affecting individuals and organizations around the world. In this project we are developing a phishing site detection system using XGBoost, a widely used machine learning algorithm that is well-known for its effectiveness and precision in classification tasks. Our approach involves extracting features from URLs and related domains, preprocessing that data, and training XGBoost's model. We test our system's performance by using a dataset of both phishing and normal websites to see how well our system detects phishing attempts. The approach involves the extraction of features from URLs and associated domains, followed by data preprocessing and the training of XGBoost's model. Performance evaluation is conducted using a dataset comprising both phishing and legitimate websites to assess the system's efficacy in detecting phishing attempts. The project aims to enhance cybersecurity measures by providing an efficient and accurate solution for identifying and mitigating phishing attacks, ultimately contributing to the protection of online users and organizations against malicious activities***

Index Terms: ml, url, phishing, machine learning, xgboost

I. INTRODUCTION

This project focuses on developing a phishing website detection system for common internet users. The rise of phishing websites on the internet necessitates the development of robust tools to identify and classify such websites. The objective is to create a detector that utilizes XG boost machine learning algorithm for phishing websites classification and detection. The system's front end will be implemented using Python. A comprehensive review of existing techniques and approaches for phishing URL detection will be conducted. This will en- compass an overview of phishing detection methods, including traditional machine learning algorithms and advanced techniques such as ensemble methods and deep learning models like XGBoost The literature review highlights the diverse range of techniques and approaches for phishing URL detection, ranging from traditional machine learning algorithms to advanced deep learning models. Future research directions may focus on hybrid approaches, leveraging the strengths of different techniques to develop more robust and effective phishing detection systems. Additionally, the integration of real-time threat intelligence and behavioral analysis could further enhance the capabilities of phishing URL detection systems in mitigating evolving cyber threats.

The methodology for phishing URL detection begins with data collection and preprocessing, where a diverse dataset of labeled phishing and legitimate URLs is gathered and cleaned. Feature engineering techniques are then applied to extract relevant features and enhance model performance. Next, a selection of machine learning algorithms, including traditional methods like logistic regression and advanced techniques like XGBoost, are evaluated and trained on the dataset. Model performance is assessed using various evaluation metrics, and hyperparameter tuning is conducted to optimize performance. The trained model is deployed into a production environment and integrated with existing security systems or web browsers for real- time detection. Continuous monitoring and improvement of the model are performed, incorporating feedback and updates to adapt to evolving phishing tactics. The framework details the structure and methodology used for evaluating various Phishing URL detection models ,built using Python with the Jupyter Notebook IDE.

The framework uses a dataset and compare different models based on precision and accuracy statistics. It systematically assesses both models discussed in the paper and additional models introduced, providing insights into their performance and effectiveness. The framework components and their interactions, along with data flow and processing steps are outlined to facilitate a comprehensive understanding of comparative analysis process. This section will delve into the implementation specifics of the project, focusing on the data collection process, annotation procedures, and preprocessing techniques applied to the collected data. It outlines the configuration and hyper parameter tuning of the selected machine learning models, such as XGBoost and ensemble methods, for phishing URL detection. The integration of feature engineering techniques, including URL length, domain age, and presence of special characters are incorporated. It is functionality implemented using Python with Jupyter Notebook IDE. The project's performance will be evaluated using appropriate metrics for phishing URL detection. Comparative analysis will be conducted to assess the performance of the developed models against existing approaches in the field.

The results will be analyzed, and the findings will be discussed, including the strengths and limitations of the system in terms of accu- racy, efficiency, and generalizability. The project's conclusions will summarize the achievements and contributions made in phishing URL detection. The potential applications and impact of the developed phishing URL detection system will be discussed, emphasizing its role in enhancing cybersecurity and safeguarding users against online threats. Suggestions for future enhancements and research directions will be provided, aiming to further improve the effectiveness and scalability of phishing detection systems. Ultimately, the project aims to contribute to creating a safer online environment by mitigating the risks associated with phishing attacks and enhancing user security while browsing the web.

II. RELATED WORKS

A. Detecting Phishing Websites using Machine Learning Al- gorithm

In general, malicious websites aid the expansion of on- line criminal activity and stifle the growth of web service infrastructure. Therefore, there is a pressing need for a com- prehensive strategy to discourage users from going to these sites online. They advocate for a method that uses machine learning to categories websites as either safe, spammy, or malicious. The proposed system is limited to examining the URL itself, rather than the contents of websites. As a result, it does away with both browser-based vulnerabilities and run- time delays. The proposed method outperforms blacklisting services in terms of generality and coverage since it makes use of learning techniques. There are three distinct categories for website addresses. Neutral Web sites provide average, risk- free functionality. For a website, "spam" refers to any attempt to overwhelm the user with advertisements or sites (such as false surveys and online dating sites). Malware is defined as a website designed by hackers to cause harm to computers and steal private data. The experimental data demonstrates a dramatic improvement in performance with the new model compared to the baseline.

B. An Efficient Phishing Attack Detection using Machine Learning Algorithms

Phishing is an illegal method which involves user's personal information at high risk. Phishing websites prey individu- als, the cloud storage hosting companies and government agencies. Though there are various anti-phishing approaches like hardware as they are not cost effective and they don't choose these approaches. To overcome this, many software- based techniques are used. Zero-day phishing problem cannot be omitted with the existing models. To prevail over these issues and detect phishing attack an approach using heuristic methodology has been proposed. They classify whether a link is phishing or non-phishing based on the input features we take like Web Traffic and Uniform Resource Locator (URL). The proposed methodology is executed by retrieving datasets from phishing cases and Machine Learning model using algorithms like Random Forest, SVM, Genetic.

C. Detecting Phishing Websites Using Machine Learning

Phishing, a cybercriminal's attempted attack, is a social web-engineering attack in which valuable data or personal information might be stolen from either email addresses or websites. There are many methods available to detect phish- ing, but new ones are being introduced in an attempt to increase detection accuracy and decrease phishing websites ' success to steal information. Phishing is generally detected using Machine Learning methods with different kinds of algorithms.This study aim is to use Machine Learning to detect phishing websites. We used the data from Kaggle consisting of 86 features and 11,430 total URLs, half of them are phishing and half of them are legitimate. We trained our data using Decision Tree (DT), Random Forest (RF), XGBoost, Multilayer Perceptrons, K-Nearest Neighbors, Naive Bayes, AdaBoost, and Gradient Boosting and reached the highest accuracy of 96.6using X G Boost.

D. Phishing Website Detection Using Random Forest and Support Vector Machine

As the Internet usage is growing rapidly, people are chang- ing their choice from traditional shopping to electronic com- merce. However, instead of robbery at banks and stores, criminals now locate their victims with some tricks in cy- ber world by applying the anonymous internet framework. Hackers are using new tactics, including phishing, to mis- lead the victims by using fake websites to gather sensitive information, including account ids, usernames, and password. Understanding whether a website is legitimate or phishing is an incredibly difficult problem because of its phishing attack structure, which primarily targets the vulnerabilities of web users. This work was proposed to solve these issues by using machine learning technology to detect the phishing websites. The system analyzed the HTML code structure that include in the hyperlink of the websites. Two machine learning tech- niques, namely Random Forest and Support Vector Machine are tested to identify the best machine learning algorithm in detecting phishing website. The performance metric for each algorithm was used as measured. The result average of

accuracy for Random Forest is 99.98 percent and Support Vector Machine is 84.73 percent. This study aims to detect phishing websites hyperlink and produces the best algorithm which is Random Forest that achieved the highest accuracy to be used for the system.

III. PROPOSED MODEL

The current solution only helps to detect if the url is effected. The proposed system will help for detecting ,url if infected then provide the original url for safe browsing Extract features from the URLs that can be used as input for machine learning models, Train a machine learning model using the preprocessed data. Using algorithms like decision trees, random forests, or neural networks for this purpose. After classification, provide the user with the original URL submitted for analysis. This can be done by storing the original URL along with the classification result and displaying it to the user when requested.

IV. PHISHING URL DETECTION

The methodology for our URL phishing detection project encompasses various stages, each crucial for the successful development and implementation of the forecasting models. Here's a breakdown of the methodology:

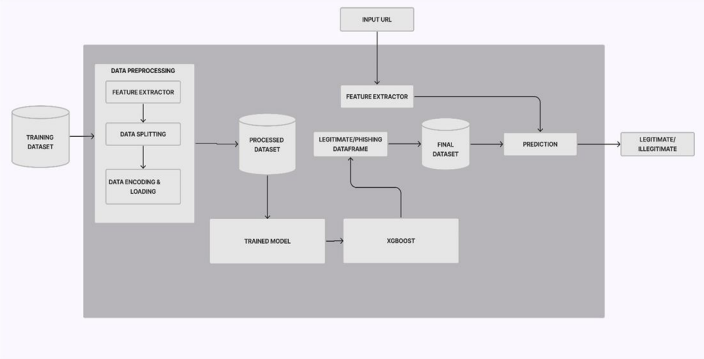


Fig. 1. Proposal Design

A. Data Collection and Preparation

Gather a diverse dataset of URLs, including both legitimate and phishing URLs. This dataset should accurately represent real-world URL patterns. Label each URL in the dataset as legitimate (0) or phishing (1). Preprocess the dataset by extracting relevant features from the URLs, such as Length of the URL, Presence of specific keywords or patterns associated with phishing , Domain reputation or age, Use of HTTPS, Presence of IP address in the URL, Number of subdomains, Domain entropy.
Encode categorical features and normalize numerical features as necessary. Split the dataset into training and testing sets.

B. Data Preprocessing:

Once the dataset is assembled, the next step is data pre- processing. This involves cleaning and organizing the data to make it suitable for machine learning algorithms. Tasks include handling missing values, scaling numerical features to a standardized range, and encoding categorical variables. The objective is to create a coherent and standardized dataset that can be effectively utilized for training and testing the machine learning model. Proper data preprocessing is essential for the model to learn meaningful patterns and relationships within the data.

C. Feature Engineering

Explore additional features that may be indicative of phish- ing URLs. Conduct feature selection or dimensionality re- duction techniques if necessary to reduce noise and improve model performance.

D. Model Training with XGBoost

Train an XGBoost model on the training dataset using the engineered features. Define the appropriate loss function and evaluation metric for the phishing detection task. Optimize hyperparameters using techniques like grid search, random search, or Bayesian optimization to maximize model performance. Use techniques like cross-validation to assess the model's generalization ability and reduce overfitting.

E. Model Evaluation

Evaluate the trained XGBoost model on the testing dataset using appropriate evaluation metrics. Analyze the model's performance using metrics such as accuracy, precision, recall, F1-score, ROC-AUC, and confusion matrix. Conduct error analysis to identify common misclassifications and areas for improvement.

F. Model Interpretation and Validation

Interpret the trained XGBoost model to understand which features contribute the most to phishing detection. Validate the model's predictions on a separate validation dataset or through real world testing to ensure its effectiveness in detecting phishing URLs.

G. Deployment and Monitoring

Deploy the trained XGBoost model for real-time phishing detection in applications such as web browsers, email filters, or network security systems. Implement monitoring mechanisms to continuously evaluate the model's performance and update it as needed to adapt to evolving phishing techniques and URL patterns.

H. Feature Extraction:

Feature extraction is a crucial phase where relevant charac- teristics are identified and extracted from the dataset. In the context of malware detection, these features could encompass file size, API calls, system calls, or opcode sequences. The selection of features significantly influences the model's ability to discriminate between malicious and benign files. Thoughtful consideration of which features to include and how to repre- sent them is paramount to the success of the detection system.

I. Splitting the Dataset:

The dataset is then split into two subsets: a training set and a testing set. Typically, around 80percent of the data is allocated for training and 20percent for testing. This division ensures that the model is exposed to a diverse range of examples during training while retaining unseen data for evaluation. Striking the right balance in the split is essential to building a model that generalizes well to new, unseen instances.

J. Model Selection:

Choosing the right machine learning algorithm is a critical decision in the development process. Commonly used algo- rithms for malware detection include Random Forests, Gradi- ent Boosting Machines (e.g., XGBoost), and Deep Learning (neural networks). The choice depends on factors such as the size and complexity of the dataset, as well as the specific re- quirements of the malware detection task. Different algorithms may excel in different scenarios, and experimentation may be necessary to determine the most suitable approach.

K. Training the Model:

With the algorithm selected, the model is trained using the training dataset. During this phase, the model learns to recognize patterns and relationships within the data. It involves adjusting the model's parameters and hyperparameters to opti- mize its performance. Techniques like cross-validation can be employed to ensure that the model generalizes well to new, unseen instances and doesn't overfit the training data.

L. Model Evaluation

Evaluate the trained XGBoost model on the testing dataset using appropriate evaluation metrics. Analyze the model's performance using metrics such as accuracy, precision, recall, F1-score, ROC-AUC, and confusion matrix. Conduct error analysis to identify common misclassifications and areas for improvement.

M. Deployment:

Once the model achieves satisfactory performance, it is ready for deployment in the target environment. Deployment involves integrating the model into the existing security in- frastructure, such as an endpoint protection system or net- work gateway. Ensuring a seamless integration process and compatibility with other systems is crucial for the successful deployment of the malware detection model. Rigorous testing in the deployment environment is necessary to verify its performance and reliability.

V. RESULT

It contains features relevant to phishing URL detection, such as URL structure, domain age, and presence of certain key- words. Extract features from your dataset that are relevant for phishing URL detection. This can include features like URL length, use of special characters, domain age, etc. Preprocess your data by handling missing values, encoding categorical variables, and scaling numerical features if necessary. Train an XGBoost model using your preprocessed dataset. Split your dataset into training and testing sets to evaluate the model's performance. Evaluate the trained XGBoost model us- ing metrics such as accuracy, precision, recall, F1 score, ROC curve, and AUC. Interpret the results to assess the performance of your phishing URL detection model. A higher accuracy, precision, recall, and AUC indicate better performance.

VI. FUTURE SCOPE

1) *Real-Time Protection in Web Browsers:* Integration of machine learning-based phishing URL detection directly into web browsers to provide users with real-time pro- tection while browsing the internet. Empower users to make informed decisions by alerting them to potential phishing attempts and malicious websites in real-time, thereby enhancing online security and privacy.
2) *Email Security and Filtering:* Integration of phishing URL detection models into email security systems and spam filters to identify and block phishing emails containing malicious links. Enhance email security by automatically scanning incoming emails for suspicious URLs and pre- venting users from accessing potentially harmful content.

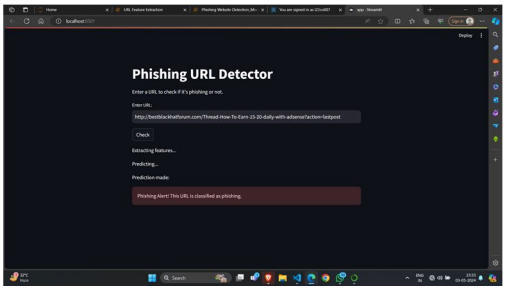


Fig. 2. negative prediction

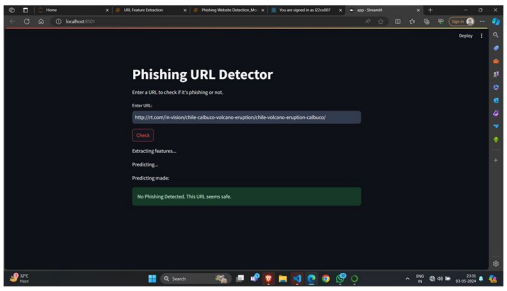


Fig. 3. positive prediction

3) *Network Security and Intrusion Detection:* Incorporation of machine learning-based phishing URL detection into network security appliances and intrusion detection systems to identify and block malicious URLs and phishing attempts at the network level. Strengthen network security by proactively detecting and mitigating phishing threats before they can infiltrate organizational networks and compromise sensitive data.
4) *Endpoint Protection and Security Solutions:* Integration of phishing URL detection capabilities into endpoint protection platforms and security solutions to provide comprehensive defense against phishing attacks across all endpoints and devices. Protect endpoints from accessing malicious URLs and prevent malware infections by detecting and blocking phishing attempts in real-time.
5) *Mobile Security and App Protection:* Integration of phish- ing URL detection models into mobile security solutions and app protection frameworks to safeguard mobile de- vices and applications from phishing attacks. Enhance mobile security by detecting and blocking phishing at- tempts targeting mobile users through malicious URLs embedded in text messages, social media apps, and other communication channels

6) *Cloud Security and Data Protection:* Integration of phish- ing URL detection capabilities into cloud security plat- forms and data protection solutions to prevent phishing attacks targeting cloud-based services and sensitive data stored in the cloud. Enhance cloud security by detecting and blocking phishing attempts aimed at stealing credentials, compromising cloud accounts, or launching ransomware attacks.
7) *IoT Security and Device Authentication:* Integration of phishing URL detection mechanisms into IoT security solutions and device authentication frameworks to protect IoT devices from accessing malicious URLs and prevent unauthorized access to IoT networks. Strengthen IoT security by detecting and blocking phishing attempts targeting vulnerable IoT devices, such as smart cameras, thermostats, and home automation systems.
8) *Cyber Threat Intelligence and Incident Response:* Uti- lization of machine learning-based phishing URL detec- tion for cyber threat intelligence gathering and incident response activities, such as identifying trends, patterns, and indicators of compromise associated with phishing attacks. Empower cybersecurity analysts and incident responders with actionable intelligence to proactively defend against phishing threats and mitigate security incidents effectively.

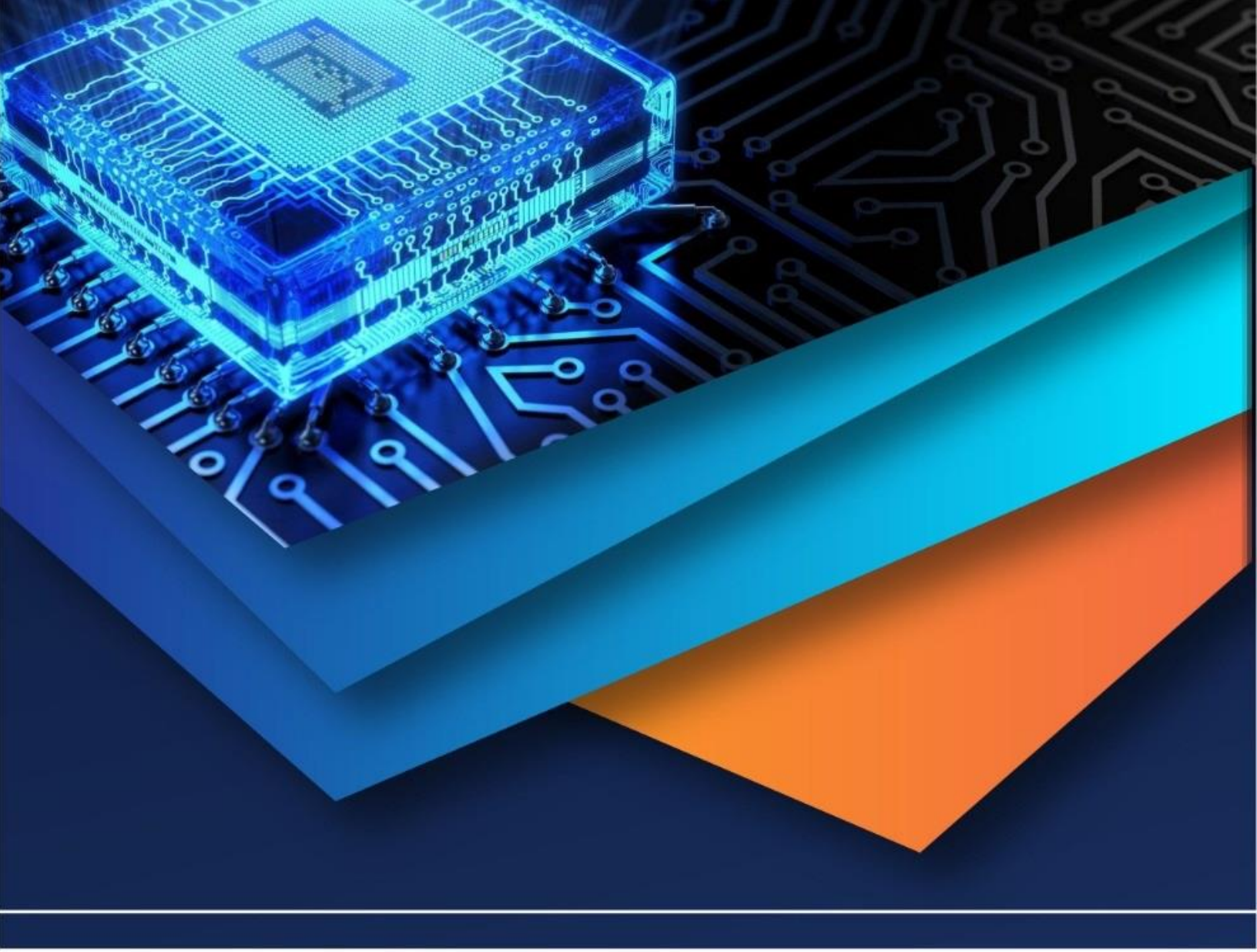
In summary, the future scope of phishing URL detection using machine learning is extensive. Advancements in model accu- racy, real-time monitoring, privacy and ethics, integration of dynamic external data sources, adaptive detection mechanisms, and predictive analytics for identifying safer browsing routes will contribute to more effective and robust phishing detection systems. These advancements will play a pivotal role in com- bating evolving phishing threats, safeguarding user privacy, and enhancing cybersecurity practices, ultimately fostering a safer and more secure online environment for individuals and organizations alike.

VII. CONCLUSION

In conclusion, the project endeavors to tackle the pervasive threat of phishing attacks through the innovative application of machine learning techniques, particularly XGBoost algorithm, to URL phishing detection. By harnessing the power of advanced algorithms, the project aims to develop a robust and adaptive system capable of effectively identifying and mitigating phishing attempts in real-time. Throughout the course of the project, we have explored various methodologies, including feature engineering, model training, and perfor- mance evaluation, to refine our detection system and enhance its accuracy and reliability. Looking ahead, the future scope of phishing URL detection using machine learning is promising, with opportunities for further advancements in model accu- racy, real-time monitoring, and integration with external data sources. As cyber threats continue to evolve, it is imperative to remain vigilant and proactive in our efforts to safeguard users and organizations from phishing attacks. By leveraging machine learning technologies, we can revolutionize cyber security practices and create a safer online environment for all.

REFERENCES

[1] Kathiravan, V. Rajasekar, S. J. Parvez, V. S. Durga, M. Meenakshi and S. Gowsalya, ”Detecting Phishing Websites using Machine Learning Algorithm,” 2023 7th International Conference on Computing Method- ologies and Communication (ICCMC), Erode, India, 2023, pp. 270-275, doi: 10.1109/ICCMC56507.2023.10083999
[2] P. Chinnasamy, N. Kumaresan, R. Selvaraj, S. Dhanasekaran, K. Ramprathap and S. Boddu, ”An Efficient Phishing Attack Detec- tion using Machine Learning Algorithms,” 2022 International Con- ference on Advancements in Smart, Secure and Intelligent Comput- ing (ASSIC), Bhubaneswar, India, 2022, pp. 1-6, doi: 10.1109/AS- SIC55218.2022.10088399.
[3] S. Alrefaai, G. O¨ zdemir and A. Mohamed, ”Detecting Phishing Websites Using Machine Learning,” 2022 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA), Ankara, Turkey, 2022, pp. 1-6, doi: 10.1109/HORA55278.2022.9799917.
[4] N. Binti Md Noh and M. N. Bin M. Basri, ”Phishing Website De- tection Using Random Forest and Support Vector Machine: A Com- parison,” 2021 2nd International Conference on Artificial Intelligence and Data Sciences (AiDAS), IPOH, Malaysia, 2021, pp. 1-5, doi: 10.1109/AiDAS53897.2021.9574282
[5] Bhavesh Borisaniya, Sridhar G, et al., ”Phishing Websites Detection using Machine Learning Techniques: A Comprehensive Review,” 2017
[6] Alaa Shublaq and Yousef Kilani, ”Feature Selection Methods for De- tecting Phishing Websites: A Survey,” 2019.
[7] Das, S., Misra, S. ”Detecting Phishing Websites Using Machine Learning and Automated Feature Selection,” IEEE Transactions on Information Forensics and Security, vol. 16, pp. 1887-1901, 2021.
[8] D. Chandrakala, A. Sait, J. Kiruthika and R. Nivetha, ”Detection and Classification of Malware,” 2021 International Conference on Ad- vancements in Electrical, Electronics, Communication, Computing and Automation (ICAECA), Coimbatore, India, 2021,IEEE.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY