MALICIOUS URL DETECTION

[1] Saleha Farha, [2] A.Riya Shree,[3] J.Sribhanu, [4] K.Yashasree.

[1]Assistant professor in Department of Information Technology, Bhoj Reddy Engineering College for Women

[2,3,4,]UG Scholars in Department of Information Technology, Bhoj Reddy Engineering College for Women

[2] riyaakhnoor09@gmail.com , [3] sribhanu931@gmail.com ,[4] yashasree4601@gmail.com

Abstract

With the increasing use of the internet and online services, phishing attacks have become a significant threat to users' personal data and security. Phishing URLs are designed to deceive users into entering sensitive information, which is then intercepted by attackers. This project aims to detect phishing URLs using machine learning algorithms, including Random Forest, Support Vector Machine (SVM), and Decision Tree. The approach involves training the model on a dataset consisting of both known phishing and legitimate URLs. Once trained, the model can be used to predict phishing attempts from new, unseen URLs. By leveraging the power of machine learning, which is increasingly being applied across various domains, this system aims to enhance online security by identifying phishing URLs and safeguarding user data against potential cyber-attacks.

I INTRODUCTION

Spam and phishing emails present significant challenges to both email service providers and users. The increasing volume of malicious emails not only consumes valuable time but also exposes users to potential security risks, with sensitive information often at stake. Traditional email filtering techniques, typically rule-based or statistical in nature, often fail to keep pace with the constantly evolving tactics used by spammers and phishers, leading to a high rate of false positives (legitimate emails marked as spam) and false negatives (malicious emails that pass undetected).To address these issues, deep learning techniques have emerged as a promising solution for enhancing email filtering systemsModels like Recurrent Neural Networks (RNNs) and Convolutional Neural Networks (CNNs) have shown remarkable abilities in processing and understanding complex patterns in textual data. By leveraging these advanced models, coupled with intelligent feature engineering and optimization strategies, we aim to create a robust and accurate spam and phishing email filtering system.

This study proposes a deep learning-based approach for efficient spam and phishing email detection. Our method focuses on improving classification accuracy while minimizing computational resources, overcoming the limitations of traditional approaches. The core strength of our proposed system lies in the ability of deep learning models to autonomously learn complex patterns and representations from the email content, making them more adept at identifying subtle distinctions between legitimate, spam, and phishing emails. The system is trained on a diverse dataset that includes legitimate emails, spam, and phishing examples, enabling the model to classify emails accurately. In addition, we employ advanced feature engineering techniques to capture both the structural and textual characteristics of emails, ensuring a comprehensive understanding of the content.

To optimize performance and ensure efficiency, we incorporate techniques like hyperparameter tuning, ensemble learning, and continuous adaptation. Hyperparameter tuning allows us to find the best configuration for the deep learning models, ensuring optimal classification results. Ensemble learning combines multiple models to improve decision-making, while continuous adaptation ensures that the filtering system remains effective over time by updating the model with new data and monitoring its performance. By integrating these techniques, our approach seeks to deliver a more efficient, scalable, and accurate spam and phishing email filtering system, capable of adapting to the ever-evolving landscape of email-based threats.

II LITERATURE SURVEY

Lee et al. proposed a machine learning-based URL filtering framework utilizing random forest and gradient boosting techniques. Their dataset consisted of 8,000 benign and 8,000 malicious URLs collected from various sources. The model achieved a detection accuracy of about 88%, with particular effectiveness in identifying phishing URLs. However, the authors noted a decrease in performance when dealing with dynamically generated domains. They stressed the importance of frequent model retraining to address emerging threats and highlighted the need for real-time analysis in practical applications. Despite offering a solid balance of precision and recall, the model's performance required constant updates to maintain effectiveness.

Ahmed and Gupta proposed a convolutional neural network (CNN)-based approach for malicious URL detection by treating each URL as a sequence of characters. Their deep learning model was trained on over 20,000 labeled URLs, achieving an accuracy of 92%. However, they pointed out that the computational overhead of using deep architectures was significant, and such models would require dedicated hardware or cloud-based resources, which could be impractical for smaller enterprises. Furthermore, the interpretability of the CNN model posed challenges, as the layers were not easily explained to users or administrators, raising concerns about transparency and trustworthiness.

Zhang et al. introduced a hybrid ensemble model that combined decision trees, logistic regression, and an anomaly detection module. By incorporating features like textual URL patterns and DNS query behaviors, the ensemble model achieved a detection accuracy of 89%. They emphasized the importance of contextual data, such as whois records and IP reputation, to enhance detection capabilities. However, the use of third-party services for whois queries sometimes introduced delays in real-time results. The authors recommended implementing an offline caching mechanism for repeated lookups to reduce latency, making the system more responsive.

Montero et al. compared k-Nearest Neighbors (k-NN) and Support Vector Machines (SVM) for malicious URL filtering. Their dataset, comprising nearly 15,000 URLs from six months of real-world traffic logs, revealed that SVM outperformed k-NN by approximately 4% in terms of accuracy. This demonstrated that SVM was better suited for high-dimensional URL feature spaces. However, they also identified a higher computational cost associated with training SVM, which could be a drawback in certain deployment scenarios. The study concluded that the choice of classifier should depend on factors such as dataset size, available computational resources, and real-time responsiveness requirements.

Choi et al. explored a big data-driven approach to malicious URL detection, leveraging distributed computing frameworks like Apache Spark. They collected over 1 million URLs from multiple threat intelligence platforms and employed an ensemble method that combined gradient boosting and Naive Bayes, achieving an accuracy of 91%. While the system scaled well with large datasets, it required a robust cluster computing environment, which could be prohibitive for smaller organizations. The authors suggested the use of incremental model updates for real-time threat mitigation, provided the infrastructure could support such operations. Despite the scalability benefits, the resource-intensive nature of big data solutions made them less suitable for smaller enterprises.

III EXISTING SYSTEM

Current antivirus solutions and URL filtering systems often rely on methods like blacklists, signature-based detection, crowdsourced feedback, and heuristic scanners to detect malicious URLs. However, these systems have notable drawbacks:

1. **Blacklists:** These require frequent updates to remain effective. Delays in updating can allow new threats to bypass detection.

2. **Signature-based Detection:** These solutions compare URL structures to known malicious patterns but are easily bypassed as attackers frequently modify link structures.

3. **Crowdsourced Feedback:** This method depends on user reports, which can be unreliable and delayed, leading to incomplete coverage and slow responses.

4. **Heuristic Scanners:** Heuristic systems scan incoming traffic for suspicious payloads. However, they can result in high false positives, flagging benign URLs as malicious due to overly broad matching rules.

IV PROBLEM STATEMENT

Malicious URLs, particularly phishing links, have become one of the most widespread cyber threats. The rise in digital connectivity, particularly with the increase in hybrid work environments and cloud services, has amplified this problem. Cybercriminals craft deceptive links that bypass traditional security measures, making it crucial to find more effective detection methods.

Statistically, organizations face millions of phishing attempts monthly, especially smaller enterprises with limited security infrastructure. Traditional detection methods such as blacklist-based approaches and heuristic scanners struggle with the adaptive tactics of cybercriminals. They either produce high false positives or fail to detect novel attacks, demonstrating the need for more refined, data-driven solutions.

Machine learning models, trained on large and diverse datasets, have shown potential in learning nuanced patterns from URLs. However, challenges like dataset bias and model interpretability remain. Therefore, there is an urgent need for a system that combines high detection accuracy with user engagement tools in real-time.

Objective:

1. **Develop an Intelligent Scanning System:** Create a secure QR code scanner capable of detecting malicious URLs embedded in QR codes.

2. **Leverage Machine Learning Algorithms:** Use machine learning to analyze URL patterns and classify potentially harmful links.

3. **Real-Time Threat Detection:** Provide immediate scanning and protection by detecting malicious URLs in real time.

4. **Minimize False Positives:** Ensure high classification accuracy while minimizing false positives, allowing legitimate URLs to pass without disruption.

5. **Improve User Security:** Protect users from phishing, malware, and other cyber threats by blocking harmful URLs before they are accessed.

6. **Enhance User Experience:** Provide a seamless and intuitive user experience for QR code scanning and URL validation.

V PROPOSED SYSTEM

The proposed system for malicious URL detection uses a curated dataset of 10,000 labeled URLs, with both benign and malicious examples. It employs four machine learning classifiers—Support Vector Machine (SVM), logistic regression, k-Nearest Neighbors (KNN), and random forest. The SVM classifier achieved 90% accuracy and is selected as the final model due to its superior performance. The system also integrates QR-based verification, allowing users to scan a QR code to confirm the safety of a URL before accessing it.

Advantages:

1. **High Accuracy:** The system effectively identifies malicious URLs, minimizing the risk of accidental clicks and reducing organizational risks.

2. **Multi-Model Approach:** By using multiple classifiers, the system ensures robust evaluation and cross-checking, improving accuracy.

3. **Web-Based Interface:** Users can easily input a URL to receive real-time automated feedback on its safety.

4. **QR Code Generation:** A QR code is generated containing the URL and its classification status, enabling quick validation and portability of results.

VI METHODOLOGY

1. **Feature Extraction:** Converts URLs into structured features, capturing both lexical and host-based attributes to enhance classification accuracy.

2. **Machine Learning Model Training and Evaluation:** The four classifiers (SVM, logistic regression, KNN, and random forest) are individually trained and evaluated. Performance metrics guide the selection of the final model.

3. **QR Code Generation:** After classifying the URL, a QR code is generated that contains the URL and its status, allowing users to quickly validate its safety.

Algorithms used

To detect malicious URLs, four machine learning models are employed:

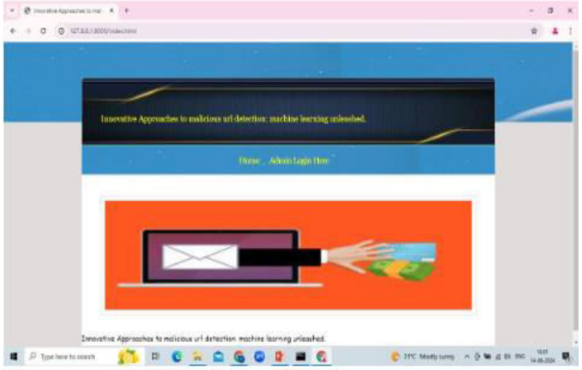
1. **Support Vector Machine (SVM):** The SVM finds an optimal margin that separates benign and malicious URLs by standardizing feature values and using kernel tricks (such as RBF). With hyperparameter tuning, it achieves 90% accuracy, making it the top-performing model.

2. **Logistic Regression:** A simpler approach that uses probabilities to classify URLs. While it's not as accurate as the SVM, logistic regression offers interpretability, making it useful for reference and understanding model decisions.

3. **k-Nearest Neighbors (KNN):** KNN compares new URLs to their nearest neighbors in the feature space, voting based on the majority classification of these neighbors. It's computationally less intensive but can struggle with performance in large datasets.

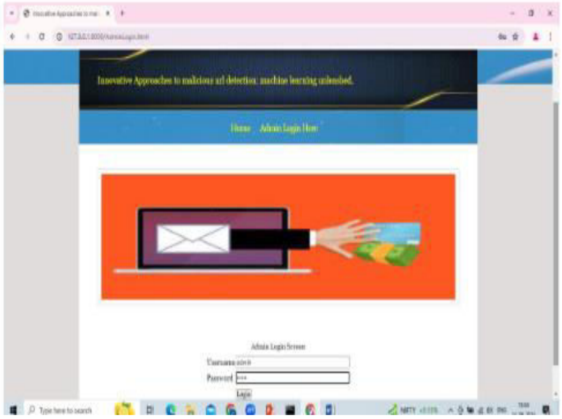
4. **Random Forest:** Random forest uses multiple decision trees to cast votes on URL classification, averaging the results to avoid overfitting. While its accuracy is lower than the SVM, it remains a robust classifier.

VII RESULTS



Innovative Approaches to malicious url detection: machine learning unleashed.

Innovative Approaches to malicious url detection machine learning unleashed.

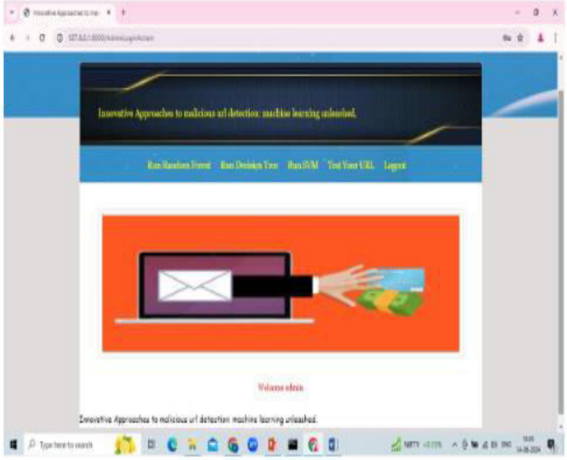


Innovative Approaches to malicious url detection: machine learning unleashed.

Admin Login Screen

Username admin

Password ••••

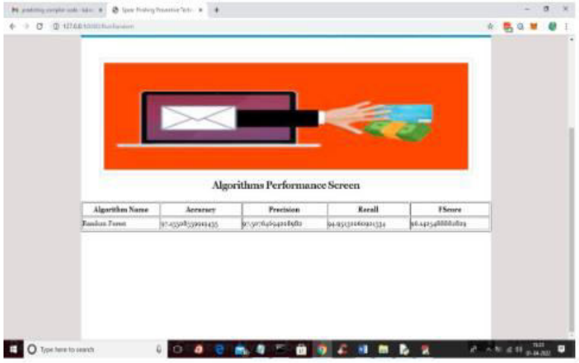


Innovative Approaches to malicious url detection: machine learning unleashed.

Run Random Forest Run Decision Tree Run SVM Test Your URL Logout

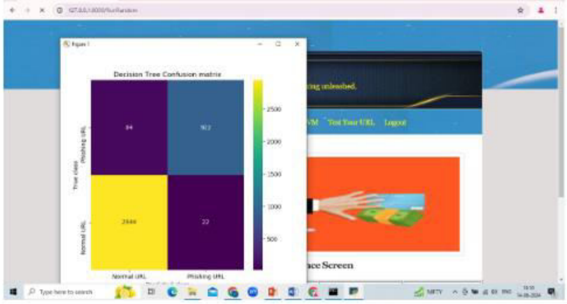
Welcome admin

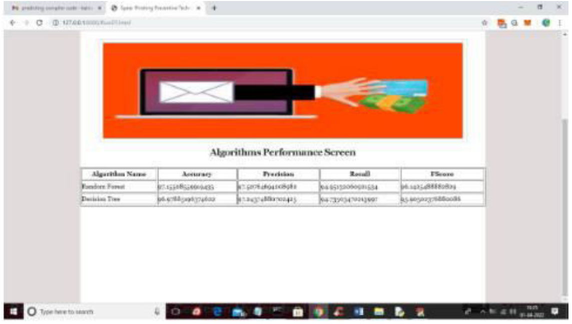
Innovative Approaches to malicious url detection machine learning unleashed.



Algorithms Performance Screen

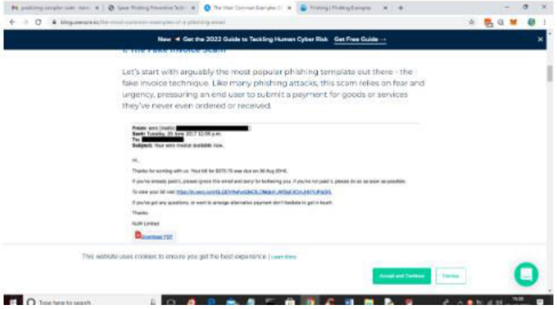
Algorithm Name	Accuracy	Precision	Recall	FScore
Random Forest	97.15508552660435	97.50784694108982	94.95152060921534	96.14254888880829





Algorithms Performance Screen

Algorithm Name	Accuracy	Precision	Recall	FScore
Random Forest	97.15508552660435	97.50784694108982	94.95152060921534	96.14254888880829
Decision Tree	96.47883296374602	97.24374880702425	94.73503470213997	95.86502376880086



New ◀ Get the 2022 Guide to Tackling Human Cyber Risk Get Free Guide →

Let's start with arguably the most popular phishing template out there - the fake invoice technique. Like many phishing attacks, this scam relies on fear and urgency, pressuring an end user to submit a payment for goods or services they've never even ordered or received.

From: [redacted]
Sent: Tuesday, 20 June 2017 12:05 p.m.
To: [redacted]
Subject: Your xero invoice available now.

Hi,

Thanks for working with us. Your bill for $370.75 was due on 30 Aug 2016.

If you've already paid it, please ignore this email and sorry for bothering you. If you've not paid it, please do so as soon as possible.

To view your bill visit [illegible]

If you've got any questions, or want to arrange alternative payment don't hesitate to get in touch.

Thanks

[illegible] Limited

Download PDF

This website uses cookies to ensure you get the best experience | Learn More

VIII CONCLUSION

In this paper, we explored the use of various machine learning algorithms—including Random Forest, Support Vector Machine (SVM), and Decision Tree—to detect phishing URLs. With the growing reliance on internet services, attackers are increasingly deploying phishing URLs that mimic legitimate websites to steal sensitive user information. When users click on these deceptive links, their input data may be transmitted to attackers, leading to significant security risks.

To counter this threat, we proposed a machine learning-based approach trained on a labeled dataset containing both phishing and legitimate URLs. The trained models effectively learn distinguishing patterns and can classify new, unseen URLs with high accuracy. Among the algorithms tested, SVM demonstrated the highest performance in terms of accuracy and robustness.

This paper highlights the potential of machine learning in strengthening cybersecurity, particularly for identifying phishing attempts in network traffic. As machine learning and deep learning technologies continue to evolve, their application in real-time threat detection systems will play a crucial role in enhancing user safety and combating cybercrime.

REFERENCES

1. Zhang, Z., Liu, Y., Zhang, X., & Zhang, H. (2020). An Efficient Email Spam Filtering Method Based on Deep Learning. IEEE Access, **8**, 182776–182785.

2. Huang, K., Huang, M., Guo, L., & Gao, J. (2020). Deep Learning for Efficient Spam Detection: A Comparative Study. In Proceedings of the 2020 IEEE International Conference on Big Data (pp. 2588–2593).

3. Ramachandran, G., & Pimple, S. (2020). Efficient Phishing Detection Using Deep Learning Techniques. In Proceedings of the 2020 International Conference on Electronics and Sustainable Communication Systems (pp. 1671–1675).

4. Mamun, M. A., Zeadally, S., & Doss, R. (2019). Deep Learning-Based Phishing Detection Techniques: A Comprehensive Survey. IEEE Access, **7**, 73050–73071.

5. Yadav, S., Bansal, R., & Saini, A. K. (2018). Deep Learning Techniques for Phishing Detection and Classification. In Proceedings of the 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA) (pp. 1–6).

6. Sinha, R., & Mohan, V. (2018). Efficient Email Spam Detection Using Deep Learning Techniques. In Proceedings of the 2018 International Conference on Communication and Signal Processing (ICCSP) (pp. 1516–1520).

7. Ayyadevara, V. S. S., & Kumar, A. A. (2017). Efficient Email Spam Classification Using Deep Learning Techniques. In Proceedings of the 2017 International Conference on Computer Communication and Informatics (ICCCI) (pp. 1–6).

8. Marinho, T., & Santos, R. (2017). Detecting Phishing Websites Using Deep Learning. In Proceedings of the 2017 IEEE/ACM 25th International Conference on Program Comprehension (pp. 313–314).