Malicious Webpage Detection Using Machine Learning

Prajwal H[1], Chirag K[2], Deepak Reddy[3], Lochan S[4]
Computer Science Department, REVA University, Bengaluru, India

***Abstract*—As the Internet experiences exponential growth, the risks associated with cybercrime have escalated, particularly through phishing attacks and deceptive online tactics. Cybercriminals create seemingly legitimate websites to manipulate users to reveal sensitive information such as login credentials, financial details, and personal data. This study focuses on developing a machine learning-based system for detecting and classifying malicious websites with the goal of preventing data from being phished. By evaluating multiple machine-learning models and leveraging feature engineering, we identified the most effective approach for accurate classification. A hugging-face model, which demonstrated the highest accuracy, was selected and integrated into a browser extension for real-time web protection. The proposed methodology enhances web security by preventing users from accessing harmful websites, thereby offering a practical and scalable solution for mitigating cyber threats.**

***Index Terms*—Internet, cybercrime, phishing attacks, deceptive tactics, malicious websites, machine learning, feature engineering, classification, hugging-face model, browser extension, web security, cyber threats, real-time protection, data privacy.**

I. INTRODUCTION

The rapid expansion of the Internet has revolutionized communication, commerce, and information exchange, offering unparalleled opportunities for individuals and organizations. However, this digital transformation has introduced an alarming surge in cyber threats, with malicious webpages emerging as a major security concern. Cybercriminals continuously refine their tactics to create sophisticated and deceptive websites that closely resemble legitimate platforms. These fraudulent sites are designed to manipulate unsuspecting users to disclose sensitive information such as login credentials, financial details, and personal data. Consequently, phishing attacks, malware distribution, and other malicious cyber activities have escalated, leading to severe consequences, including identity theft, financial fraud, and large-scale data breaches.

Traditional cybersecurity mechanisms, such as blacklisting and heuristic analysis, have proven to be insufficient for addressing the dynamic and evolving nature of cyber threats. Blacklists, which rely on precompiled databases of known malicious websites, have quickly become obsolete as new threats emerge at an unprecedented rate. On the other hand, heuristic-based detection methods analyze behavioral patterns to identify suspicious activity but often struggle with high false-positive rates and an inability to detect novel attack vectors. These limitations highlight the need for more adaptive and intelligent cybersecurity solutions capable of identifying and mitigating emerging threats in real-time.

To overcome these challenges, this study focuses on developing a machine learning-based model for detecting and classifying harmful webpages. By utilizing advanced learning algorithms and feature-engineering techniques, the proposed system aims to enhance the accuracy and efficiency of malicious website detection. Machine learning offers the advantage of continuous learning from new threats, allowing for the real-time identification of fraudulent websites with minimal reliance on manually updated databases. This approach significantly improves cybersecurity defenses by proactively identifying malicious sites before they can cause harm.

This paper outlines the methodologies used to construct the model, including data collection, feature extraction, model training, and performance evaluation. It further discusses the objectives of the project and the anticipated outcomes, emphasizing the potential of machine learning to strengthen web security. Ultimately, this research contributes to a more adaptive and robust cybersecurity framework that equips users with an intelligent, scalable, and proactive defense mechanism against evolving online threats.

II. LITERATURE SURVEY

Xuan and Nguyen (2022) [1] investigated the application of machine learning techniques for malicious URL detection. This study utilized various classifiers, including decision trees and support vector machines, to classify URLs as benign or malicious. The experimental results demonstrated that feature engineering plays a crucial role in improving the model accuracy.

Rahman et al. (2021) [2] studied neural network-based approaches for URL detection, focusing on deep feedforward networks and long short-term memory (LSTM) models. Their research highlighted that deep learning models outperform traditional classifiers, especially when trained on large datasets.

Sharma et al. (2020) [3] explored deep learning techniques for detecting malicious URLs. The authors utilized convolutional neural networks (CNNs) and recurrent neural networks (RNNs) to analyze the structure and behavior of URLs, achieving significant improvements over traditional machine-learning methods.

Mahajan and Kumar (2021) [4] proposed a hybrid feature-based approach that integrates lexical, host-based, and content-based features to improve detection accuracy. The study demonstrated that combining multiple feature types enhances the robustness of machine-learning models against adversarial attacks.

Naidu et al. (2022) [5] analyzed domain attributes to enhance malicious URL detection. Their study emphasized the significance of domain registration patterns, WHOIS information, and DNS records in identifying suspicious domains. The findings suggest that domain-attribute-based models can complement machine learning techniques for better detection performance.

Al Kadhim et al. (2021) [6] conducted a comparative study on multiple machine learning algorithms for URL detection. This research evaluated models such as Random Forest, Naïve Bayes, and Gradient Boosting, highlighting their respective strengths and weaknesses in detecting malicious URLs. The findings emphasize the importance of selecting the correct features to optimize detection performance.

Kumar and Kumar (2019) [7] focused on detecting short malicious URLs. This study analyzed various characteristics of shortened URLs, such as redirection behavior and embedded content, to differentiate between benign and harmful links. The research underscored the challenges associated with detecting malicious intent in shortened URLs owing to their obfuscated nature.

Shibu et al. (2021) [8] introduced a lightweight malicious URL-detection system designed for real-time applications. The authors leveraged lightweight machine-learning models that require minimal computational resources while maintaining high accuracy, making them suitable for deployment in resource-constrained environments.

Mathur et al. (2024) [9] proposed a hybrid deep learning framework that combines CNNs with traditional machine learning techniques for enhanced malicious URL detection. Their study demonstrated the effectiveness of hybrid models in improving the classification accuracy and reducing false positives.

III. MACHINE LEARNING MODELS AND TECHNIQUES

Malicious webpage detection using machine learning primarily falls into two categories: supervised and unsupervised.

Supervised Learning

Supervised learning relies on labelled datasets, in which URLs or webpage features are explicitly categorized as malicious or benign. The model learns from these labelled examples to accurately classify the new URLs. Common algorithms used include the following.

Support Vector Machines (SVM) are effective for high-dimensional data classification.

Decision Trees & Random Forests: Rule-based models that identify malicious patterns.

Deep Learning (CNNs, RNNs): Extracts complex features from webpage content and behaviour.

Although this approach provides high accuracy, it requires a continuously updated labelled dataset, which can be challenging to maintain.

Unsupervised Learning

Unsupervised learning detects malicious activities without labelled data by identifying patterns and anomalies. Common methods include:
Clustering (K-Means, DBSCAN): Groups similar URLs based on their characteristics.
Anomaly Detection (Autoencoders, Isolation Forests): Flags deviate from normal behavior.
This approach is useful for detecting zero-day attacks but may produce more false positives compared to supervised methods.

IV. DATASET USED

It is imperative that the growth of machine learning models depends heavily on the quality and range of the dataset that is fed to the model. Commonly used datasets for malicious URL detection include the following:

PhishTank: Fake websites that are usually developed by malicious people can be found here.

OpenPhish: Gives a list of live phishing sites that users can filter category-wise and check their success rate.

Kaggle: Contains diverse datasets that include both realistic and fake URLs vital for research or competition.

V. CHALLENGES FACED

Despite significant advancements in machine learning, malicious webpage detection remains a complex and evolving challenge. Several key factors hinder the development of efficient and accurate detection models.

1. Feature Engineering Complexity

The effectiveness of a machine-learning model depends on selecting relevant and diverse features, such as the URL structure, domain attributes, webpage content, and network behavior. An inadequate feature selection process can lead to poor model generalization, whereas excessive features increase the computational complexity without necessarily improving the detection accuracy.

2. Data Imbalance

Malicious webpages constitute a small fraction of all web traffic, leading to a class imbalance in the training datasets. Because benign webpages significantly outnumber malicious webpages, models trained on such data tend to be biased toward benign classifications, thereby reducing their ability to detect rare but critical threats.

3. Adaptive and Evasive Threats

Attackers frequently modify phishing sites and malicious domains using techniques such as domain-generation algorithms (DGAs), obfuscation, and encoding to evade detection. This dynamic nature of threats makes it challenging for static models to maintain high detection accuracy without frequent retraining.

By utilizing a pre-trained hugging-face model, we optimize the detection efficiency while minimizing resource constraints, making this approach a practical and scalable solution for malicious webpage detection.

VI. PERFORMANCE METRICS:

To evaluate the effectiveness of malicious URL detection, we implemented and compared multiple machine-learning models. Each model was trained using labelled datasets containing both benign and malicious URLs to ensure a balanced and representative dataset for classification. The models were assessed on the basis of their predictive accuracy, generalization capability, computational efficiency, and robustness against evolving cyber threats. The primary goal of this evaluation was to identify the most effective model for real-time malicious URL.

The models used in our research were as follows:

1. *Decision Tree Classifier:*

A rule-based, tree-structured model recursively splits the dataset based on feature values to create a hierarchical classification structure. It is highly interpretable and computationally efficient, making it suitable for real-time application. However, decision trees are prone to overfitting, especially when dealing with complex datasets, as they tend to learn specific patterns rather than generalize well to unseen data.

2. *Random Forest Classifier:*

Ensemble learning techniques enhance the performance of decision trees by constructing multiple trees and aggregating their predictions. This approach significantly improves generalization by reducing overfitting and increasing the model stability. By averaging the outputs of numerous independently trained decision trees, random forests provide robust

performance, even when handling noisy or imbalanced datasets.

*3. **Extra Trees Classifier:***

A variant of the Random Forest algorithm introduces additional randomness during the tree construction. Unlike traditional decision trees, Extra Trees randomly select split points instead of choosing the best split based on the information gain or Gini impurity. This additional randomness helps to reduce variance, making the model more resilient to overfitting and improving robustness against unseen malicious patterns.

4. AdaBoost Classifier:

An ensemble boosting method that iteratively combines multiple weak classifiers creates a strong predictive model. AdaBoost assigns higher weights to misclassified instances, forcing the subsequent classifiers to focus on difficult cases. This approach enhances accuracy and reduces bias, but can be sensitive to noisy data and outliers, which may negatively impact the overall performance.

5. Gaussian Naïve Bayes (Gaussian NB)

A probabilistic classification model based on Bayes' theorem assumes independence between features.

The GaussianNB is computationally efficient and works well with small datasets, making it suitable for real-time detection. However, its performance may be limited when dealing with correlated features because the independence assumption does not always hold in real-world scenarios.

6. Stochastic Gradient Descent (SGD) Classifier:

A linear classification model was optimized using gradient descent, making it highly scalable and suitable for large datasets. SGD is particularly effective when handling high-dimensional feature spaces but requires careful hyperparameter tuning to achieve optimal performance. In addition, it is sensitive to the learning rate selection, which can affect the convergence speed and model stability.

VII. OBSERVATIONS

Random Forest and Extra Trees performed well owing to their ensemble nature, reduced overfitting, and improved generalization.

AdaBoost showed competitive performance, but was sensitive to noisy features.

The decision Tree provided good interpretability, but had a higher variance.

Gaussian NB is computationally efficient but struggles with feature dependencies.

The SGD Classifier was suitable for large-scale datasets but required careful tuning for optimal performance.

VIII. INITIATIVE

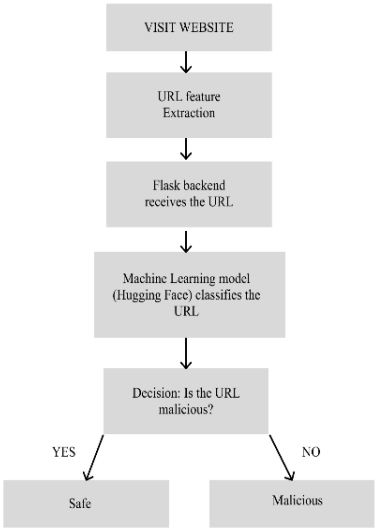

Fig 1: Malicious URL Detection and Classification Workflow

The flowchart outlines a machine learning-based system for detecting malicious URLs, enhancing web security by preventing users from accessing harmful websites. The process begins when a user visits a website, triggering URL feature extraction, where relevant characteristics, such as domain attributes, length, special characters, and patterns, are analyzed. The extracted URL data are then sent to a Flask backend, which processes the information and forwards it to a pretrained Hugging-Face machine learning model. This model classifies URL based on learned patterns and determines whether they are safe or malicious. The system then makes a final decision: If the URL is classified as safe, the user can proceed without concern; however, if it is deemed malicious, a warning is issued to prevent potential cybersecurity threats. This real-time detection mechanism provides an effective and scalable solution to mitigate phishing attacks and other online security risks.

IX. METHODOLOGY

Model Comparison and Evaluation

To identify the most effective approach for malicious URL detection, six machine learning models—Decision Tree, Random Forest, Extra Trees, AdaBoost, Gaussian Naïve Bayes (GaussianNB), and Stochastic Gradient Descent (SGD) classifiers —were trained and evaluated using a labeled dataset containing both benign and malicious URLs. The models were assessed based on multiple performance metrics, including accuracy, precision, recall, and F1-score, to determine their classification effectiveness. While traditional models provide varying levels of performance, pre-trained models from Hugging Face demonstrated superior precision and recall, making them the optimal choice for the accurate and real-time detection of malicious websites.

Transfer Learning with Hugging Face

To enhance the detection accuracy and reduce the computational overhead, transfer learning was applied using a pretrained hugging-face model. This approach allows the model to leverage knowledge from large-scale datasets, thereby improving its ability to recognize malicious patterns in URLs. The model was fine-tuned using domain-specific features, ensuring that it was adapted to evolving phishing tactics and other cyber threats. By utilizing transfer learning, the system maintained high accuracy while significantly reducing the time and resources required for training.

React-Based Browser Extension

A React-based browser extension is developed to automate the URL analysis process. The extension continuously monitors URLs entered by users in the browser, extracts key features, and sends them for classification. This seamless integration enables real-time security checks without disrupting the user browsing experience. The extension was designed with a user-friendly interface that provided instant alerts and warnings whenever a potentially harmful website was detected.

Flask Backend for Classification

A Flask-based backend API is implemented to act as the central processing unit of the detection system. When the browser extension captures a URL, it is sent to the Flask server, where it undergoes feature extraction and classification using the fine-tuned hugging-face model. The backend processes incoming requests efficiently, classifies URLs as safe or malicious, and returns results in real time. The lightweight nature of Flask ensures fast response times, enabling seamless integration with browser extension.

Deployment and Real-Time Detection

A fully integrated system was deployed to provide real-time malicious URL detection and protection. Upon detecting a malicious URL, the browser extension immediately alerts the user, thereby preventing access to potentially harmful websites. This proactive approach enhances cybersecurity by mitigating risks associated with phishing attacks, malware distribution, and fraudulent websites**.** The solution is scalable and adaptive, ensuring continued effectiveness against emerging cyber threats, while maintaining a low computational footprint for optimal performance.

X. CONCLUSION

Machine learning has proven to be a powerful tool in the detection of malicious webpages, offering a more dynamic and adaptive approach than traditional detection methods. Machine learning models can effectively differentiate between benign and harmful websites by analyzing complex URL structures, domain attributes, and webpage content. The use of advanced classification techniques, feature engineering, and transfer learning has significantly enhanced the detection accuracy, providing real-time protection against phishing attacks, malware distribution, and fraudulent websites.

Despite these advancements, several challenges remain to be overcome. Feature extraction plays a crucial role in model performance, and identifying the most relevant attributes for distinguishing malicious URLs remains a complex task. Data imbalance is another issue, as datasets often contain fewer malicious samples than benign ones, which can lead to biased model predictions. Additionally, adversarial attacks, in which cybercriminals manipulate website structures to bypass detection systems, pose an ongoing threat, requiring models to be more robust and adaptive.

Future research should focus on enhancing model resilience by integrating real-time data updates to more efficiently detect newly emerging threats. Exploring hybrid approaches that combine machine learning models with rule-based techniques, deep

learning architectures, and threat intelligence feeds could further improve detection accuracy and adaptability. Additionally, incorporating continuous learning mechanisms would enable the model to evolve alongside changing cyberthreat landscapes, ensuring long-term effectiveness in combating online security risks.

By addressing these challenges and refining existing methodologies, machine-learning-based malicious URL detection systems can serve as scalable, efficient, and proactive solutions for strengthening cybersecurity and protecting users from evolving digital threats.

REFERENCE

[1] Xuan, Q., Nguyen, D. (2022). Malicious URL Detection Based on Machine Learning. Semantic Scholar.

[2] Rahman, K., et al. (2021). Neural Network Approaches for URL Detection. IOP Conference Series: Journal of Physics: Conference Series, 2037(1), 012016.

[3] Sharma, A., Jain, M., Singh, H. (2020). Deep Learning for Malicious URL Detection. IEEE Xplore.

[4] Mahajan, S., & Kumar, M. (2021). Hybrid Feature-based Detection of Malicious URLs. IEEE Xplore.

[5] Naidu, P., Babu, G. S., and Vijayalakshmi, P. (2022). Domain Attribute Analysis in Malicious URL Detection. Journal of King Saud University, Computer and Information Sciences.

[6] Al Kadhim, A., et al. (2021). Comparative Study of Machine Learning Algorithms for URL Detection. UHD Journal of Science and Technology.

[7] Kumar, R., & Kumar, S. (2019). Malicious Short URL Detection. International Research Journal of Engineering and Technology (IRJET) 6(11): 152–156.

[8] Shibu, S., et al. (2021). Lightweight Malicious URL Detection Systems. International Research Journal of Engineering and Technology (IRJET).

[9] Mathur, A., et al. (2024). Hybrid Approach for Malicious URL Detection. In Lecture Notes in Computer Science. Springer, Cham.