www.ijbar.org
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86

Malicious URL Detection Using Machine Learning

[1] Mrs.J.Santoshi, [2] Neelam Manasa, [3]Papakanti Tulasi, [4] Afroz fatima

1 Assistant Professor in the Department of C.S.E, Matrusri Engineering College, Saidabad, Hyderabad, Telangana, India

[2,3,4] UG Scholar in the Department of C.S.E, Matrusri Engineering College, Saidabad, Hyderabad, Telangana, India

Abstract

The increasing frequency of cyber attacks through malicious URLs—used for phishing, malware distribution, and online fraud—has exposed the limitations of traditional blacklist-based detection systems. This project presents an end-to-end machine learning-based approach for detecting malicious URLs by analyzing both lexical and host-based features.Developed using Python and Jupyter Notebook, the system processes a dataset of URLs by extracting features such as URL length, frequency of special characters, presence of IP addresses, and suspicious keywords. Multiple machine learning models, including Random Forest, Logistic Regression, Decision Tree, and XGBoost, are trained and evaluated to determine the most accurate classifier. A user-friendly graphical interface, built with Tkinter, allows real-time URL input and displays prediction outcomes, making the tool accessible for general users.The system demonstrates high precision and effectiveness, providing a lightweight yet robust solution for early detection and prevention of web threats. This project highlights the practical application of machine learning in cybersecurity and underscores the importance of intelligent URL analysis in today's digital landscape.

I INTRODUCTION

The widespread integration of internet services into daily life—spanning communication, finance, commerce, and education—has been paralleled by a significant rise in cybersecurity threats. Among these, **malicious URLs** have emerged as a primary vector for phishing attacks, malware distribution, and online fraud. Cyber adversaries increasingly exploit unsuspecting users by embedding harmful links in emails, messages, advertisements, and websites. These URLs are deliberately designed to mimic legitimate web addresses or redirect users to deceptive platforms, often resulting in compromised credentials, financial loss, or system infection.

Conventional defense mechanisms, such as blacklist-based URL filtering and rule-based detection systems, have proven insufficient in the face of rapidly evolving threat landscapes. These methods rely heavily on known attack signatures or predefined heuristics, limiting their ability to detect newly generated or obfuscated malicious URLs. Furthermore, the scalability and adaptability of such systems are challenged by the volume and variability of modern web traffic.

www.ijbar.org
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86

To overcome these limitations, recent research has shifted toward data-driven techniques—particularly those leveraging **machine learning (ML)**. ML models can learn patterns from historical data and generalize to detect previously unseen threats. In this paper, we present an end-to-end machine learning framework for **malicious URL detection**, integrating both lexical (e.g., URL length, special characters, keyword presence) and host-based features (e.g., use of IP addresses, domain registration attributes). A range of supervised learning models—including Random Forest, Logistic Regression, Decision Tree, and XGBoost—are trained and evaluated for their effectiveness in binary classification of URLs.

To enhance practical usability, we incorporate a real-time prediction interface via a Graphical User Interface (GUI) built with Tkinter. This allows even non-technical users to input URLs and receive instant feedback regarding potential threats.

This work contributes a lightweight, effective, and interpretable solution for early-stage web threat detection. The proposed model demonstrates high classification accuracy and highlights the viability of machine learning-based systems as a proactive layer of defense in modern cybersecurity infrastructures.

II LITERATURE SURVEY

The detection of malicious URLs has become a critical focus in cybersecurity due to the increasing inadequacy of traditional blacklist-based systems in identifying zero-day threats and dynamically generated URLs. Recent research has applied machine learning techniques for improved detection using lexical and host-based feature sets.

A hybrid detection approach was proposed using lexical features such as URL length, token count, and the presence of special characters, in combination with content-based attributes. Classification was performed using Decision Tree and Support Vector Machine models, which showed improved performance over rule-based systems, though the exclusion of host-based features limited its overall detection capability [1].

A scalable system combining lexical analysis with host-based attributes such as domain age, IP usage, and WHOIS registration data demonstrated high detection accuracy when trained with Random Forest and Gradient Boosting classifiers. However, the model suffered from latency issues in real-time scenarios due to delays in retrieving host-based metadata [2].

Character-level Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) have also been utilized to automatically extract features from URLs, achieving strong performance on large datasets. Despite the accuracy gains, these deep learning approaches demand significant computational resources and lack transparency, making them less suitable for lightweight or interpretable systems [3].

www.ijbar.org
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86

Ensemble learning approaches have been applied by combining models such as Logistic Regression, Naive Bayes, and XGBoost to improve generalization and reduce false positives. This strategy, paired with effective feature selection and dimensionality reduction, produced reliable classification results across diverse datasets [4].

An implementation that integrated both lexical and host-based features within a Random Forest classifier, accompanied by a user-friendly GUI using Tkinter, provided real-time malicious URL detection. The system emphasized usability and practical deployment, especially for non-technical end users [5].

Lightweight frameworks optimized for edge devices have also been explored, where tree-based models like Decision Tree and Random Forest were shown to strike a balance between computational efficiency and detection accuracy. These models were found to be highly suitable for real-time and resource-constrained environments [6].

These works collectively establish the effectiveness of machine learning in malicious URL detection, especially when a hybrid feature approach is used. However, most existing systems either fall short in real-time execution or lack accessibility. This motivates the present study, which aims to deliver an accurate, efficient, and user-friendly malicious URL detection system.

III PROBLEM STATEMENT

The increasing use of malicious URLs for cybercrimes such as phishing, malware delivery, and online fraud presents a significant threat to internet users. Traditional detection methods—such as blacklist-based filtering and rule-based systems—are proving to be ineffective against newly generated, obfuscated, or zero-day malicious URLs. These outdated approaches struggle to keep pace with the dynamic and evolving nature of online threats, leaving users vulnerable. There is a growing need for an intelligent, scalable, and real-time URL classification system that can accurately detect and block malicious links before they cause harm. A machine learning-based solution that analyzes static lexical and host-based features offers a promising alternative by learning patterns associated with malicious URLs and enabling proactive detection, even in offline environments.

A. Objectives

To design and implement an intelligent system that classifies URLs as either malicious or benign using machine learning techniques.

To extract and analyze relevant static features from URLs, such as length, presence of special characters, use of IP addresses, and suspicious keywords indicative of malicious behavior.

To train and evaluate multiple machine learning algorithms—including Random Forest, Logistic Regression, Decision Tree, and XGBoost—to

www.ijbar.org
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86

identify the most effective model for URL classification.

To enable real-time URL classification through a user-friendly graphical interface (GUI) that allows users to input URLs and receive instant results.

To contribute to cybersecurity by automating URL threat detection and minimizing reliance on outdated blacklist-based methods.

Scope

Static Feature Analysis: The system utilizes lexical and host-based features extracted from the URL string and domain information. It does not perform dynamic analysis or inspect webpage content.

Focus on Classical Machine Learning: Only traditional machine learning algorithms are employed; deep learning models and behavioral analysis techniques are outside the scope.

Offline Operability: The system relies on pre-trained models, allowing it to function without an internet connection once deployed.

Graphical User Interface: A lightweight GUI developed using Tkinter is provided to facilitate ease of use, enabling users to input URLs and receive threat assessments.

Extendable Security Tool: While currently designed as a standalone application, the system can be extended for integration with web browsers, email clients, or antivirus software to enhance real-time threat protection.

IV SYSTEM ARCHITECTURE

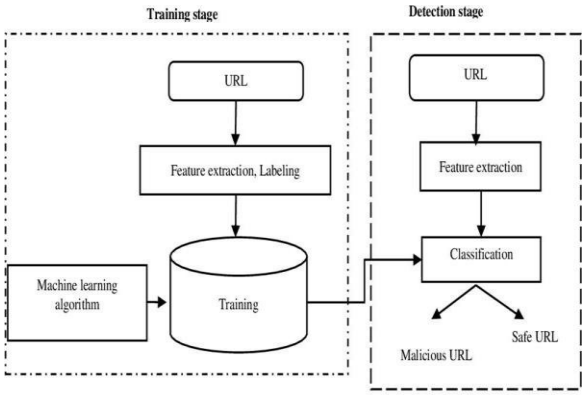


Fig 1 Malicious URL Detection Model using Machine Learning

V PROPOSED METHODOLOGY

The proposed machine learning-based approach for malicious URL detection involves a structured, multi-phase pipeline comprising problem analysis, data collection, pre-processing, model development, and evaluation. The goal is to create a robust and scalable system capable of accurately detecting and classifying URLs in real-time while adapting to evolving cyber threats.

1. Problem Analysis

Understanding the characteristics and behavior of malicious URLs and the strategies employed by cybercriminals to deceive users.

Analyzing existing detection techniques (e.g., blacklist-based methods) and identifying their limitations in terms of adaptability and accuracy.

www.ijbar.org
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86

2. **Data Collection**

Acquiring labeled datasets of URLs from reliable sources such as PhishTank, OpenPhish, Alexa Top 1M, and Kaggle.

Labeling the collected URLs into two categories: malicious and benign, based on their origin and nature.

3. **Pre-processing and Feature Extraction**

Lexical Features: Extracting characteristics from the URL string itself, such as length, the number of special characters, subdomain count, and the use of suspicious keywords.

Host-Based Features: Extracting data related to hosting information like WHOIS registration details, domain age, and IP address usage.

Network-Based Features (if available): Analyzing server responses, redirection patterns, and basic traffic behavior.

Addressing class imbalance in the dataset using oversampling techniques such as SMOTE (Synthetic Minority Over-sampling Technique) to enhance model fairness and generalization.

4. Model Development

Implementing and training various machine learning algorithms, including:

Random Forest: For analyzing feature importance and offering reliable classification.

XGBoost: For high-speed, high-accuracy classification, particularly on structured data.

Neural Networks: For advanced pattern recognition and extracting complex non-linear features, especially in large-scale datasets.

5. **Model Evaluation**

Assessing model performance using comprehensive metrics to ensure reliability and robustness:

Accuracy: Overall correctness of the model.

Precision: The ratio of correctly identified malicious URLs to all URLs predicted as malicious.

Recall: The model's ability to detect all actual malicious URLs.

F1-Score: Harmonic mean of precision and recall.

ROC-AUC: Evaluates the model's capability to distinguish between classes at various thresholds.

V IMPLEMENTATION

The development of the malicious URL detection system was carried out in a step-by-step manner, starting with the collection and preparation of data. Labeled URL datasets—indicating whether a URL is malicious or benign—were obtained from trusted open sources such as PhishTank, OpenPhish, and Kaggle. These datasets required

www.ijbar.org
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86

preprocessing to ensure the quality of input fed into the models. This stage involved cleaning the data by removing any duplicates or missing entries, standardizing the format of URLs through normalization, and converting the labels into numerical values using encoding techniques, making the data compatible with machine learning algorithms.

Once the data was clean and ready, the next step involved extracting meaningful features from each URL. These features were primarily divided into two categories: lexical and host-based. Lexical features refer to those directly derived from the structure of the URL, such as its total length, the number of dots it contains, presence of embedded IP addresses, and the use of specific special characters like '@', '-', or '='. Certain suspicious keywords like “login” or “verify,” commonly found in phishing links, were also considered. In addition, where available, host-based features were used. These included details like the age of the domain—retrieved from WHOIS data—and whether or not the URL made use of the secure HTTPS protocol.

With the features extracted, model development and training followed. The dataset was split into training and testing portions to ensure a balanced evaluation. Several machine learning algorithms were tested, including Random Forest, Logistic Regression, Decision Tree, and XGBoost. Each model was trained using the extracted features, and its performance was measured using well-established metrics like accuracy, precision, recall, and F1-score. This evaluation process helped identify which algorithm performed best in correctly identifying malicious URLs while minimizing false positives and false negatives.

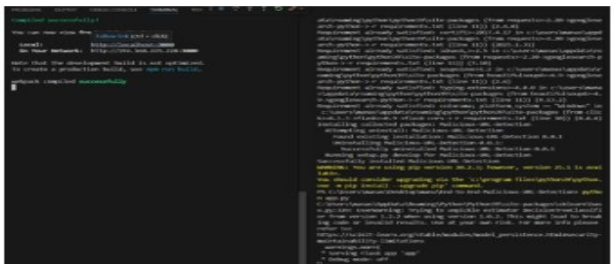
After identifying the most effective model, it was integrated into a real-time prediction engine. This engine takes in a new URL as input, extracts the relevant features using the same method as during training, and then uses the trained model to classify the URL as either malicious or benign. The result of this prediction is then passed to the front-end for user interaction.

To make the system accessible to all users, a simple graphical interface was built using Python’s Tkinter library. The GUI allows

www.ijbar.org
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86

users to input a URL, initiate the detection process with a single click, and receive the classification result almost instantly. Its clean layout and ease of use make it suitable even for users without technical expertise, turning the system into a practical and approachable tool for improving everyday web safety.

VI RESULTS



VII CONCLUSION

This paper successfully demonstrates the practical application of machine learning in the detection of malicious URLs by leveraging lexical features such as URL length, presence of special characters, embedded IP addresses, and suspicious keywords. The proposed system efficiently classifies URLs as either malicious or benign using a variety of machine learning models, including Random Forest, XGBoost, Logistic Regression, and Decision Tree. Among these, ensemble methods like XGBoost and Random Forest show superior performance in terms of accuracy and robustness.A key contribution of this work is the integration of a user-friendly graphical

interface developed using Tkinter, which enables real-time URL verification for end users. This feature enhances accessibility, allowing even non-technical individuals to benefit from the system's capabilities without requiring a deep understanding of machine learning or cybersecurity.

REFERENCES

1. Marchal, S., et al. (2017). *Detecting Malicious URLs using Machine Learning Techniques*. International Conference on Cyber Security. Retrieved from https://www.researchgate.net/
2. PhishTank. (n.d.). *PhishTank: An Open Dataset for Phishing URL Detection*. Retrieved from https://www.phishtank.com/
3. Ma, J., et al. (2018). *URL Classification Using Machine Learning*. *Journal of Cyber Security and Digital Forensics*, 6(3), 233–248.
4. Sahoo, K., et al. (2019). *A Comparative Study of Machine Learning Techniques for Malicious URL Detection*. *International Journal of Computer Applications*, 178(7), 10–20.
5. Chen, T., & Guestrin, C. (2016). *XGBoost: A Scalable Tree Boosting System*. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. DOI: 10.1145/2939672.2939785
6. Kaggle. (2021). *Malicious URL Detection Dataset*. Retrieved from https://www.kaggle.com/
7. Zhang, J., et al. (2019). *Malicious URL Detection Using Neural Networks*. *Journal of Cyber Security and Privacy*, 5(4), 120–131.
8. Tkinter Documentation. (n.d.). *Tkinter: Python GUI Programming*. Retrieved from https://docs.python.org/3/library/tkinter.html