iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** IV **Month of publication:** April 2026

DOI: https://doi.org/10.22214/ijraset.2026.79440

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 14 Issue IV Apr 2026- Available at www.ijraset.com

Malicious URL Detection Using Machine Learning

Ms. G. Indu[1], Ch. Vishwas[2], P. Sricharan[3], A. Vinay[4]
Dept. of CSE (Data Science) Institute of Aeronautical Engineering Dundigal, Hyderabad

Abstract: ***This paper presents a Malicious URL Detection System that leverages advanced Machine Learning (ML) and Natural Language Processing (NLP) techniques to identify and classify harmful web addresses with high accuracy. The system aims to detect phishing, spam, and malware-distributing URLs by analyzing lexical, host-based, and content-based features derived from large-scale URL datasets. Through the use of algorithms such as Random Forest, XGBoost, and Logistic Regression, the model effectively distinguishes between benign and malicious URLs without relying solely on traditional blacklists.***

The system incorporates automated preprocessing pipelines including URL normalization, feature extraction, and vector-ization to handle diverse data formats. It employs supervised learning techniques to train models capable of real-time URL threat detection, supported by explainable AI modules for en-hanced interpretability. A web-based interface built using Python, Streamlit, and Plotly provides dynamic visual analytics and real-time detection results, allowing users to input URLs and instantly receive predictions along with confidence scores. With its scalable architecture, the framework can be integrated into email clients, browsers, or cybersecurity platforms to provide proactive protection against online threats. This intelligent solu-tion represents a step forward in automating web threat detection and contributes to a safer digital ecosystem by mitigating cyber risks before they can impact end-users.

Index Terms: Malicious URL Detection, Machine Learning, Phishing, Cybersecurity, XGBoost, Random Forest, NLP, Stream-lit, Threat Intelligence.

I. INTRODUCTION

In today's digitally connected world, the exponential growth of online platforms, e-commerce, and social media has signif-icantly increased internet usage across all domains. However, this rapid digital transformation has also led to a surge in cyber threats, particularly through the spread of malicious URLs used in phishing, spamming, and malware distribution. Traditional blacklist-based security mechanisms, which rely on maintaining large databases of known malicious domains, often fail to detect newly generated or dynamically changing malicious URLs. As cyber attackers continuously evolve their strategies, such reactive approaches struggle to provide real-time protection against emerging threats.

To overcome these limitations, this paper presents an intelli-gent Malicious URL Detection System that leverages Machine Learning (ML) to automatically identify and classify URLs as benign or malicious. Unlike conventional methods that depend solely on signature or rule-based filtering, the proposed framework utilizes data-driven learning techniques to analyze lexical, host-based, and content-based features extracted from URLs. By employing algorithms such as Random Forest, XGBoost, and Logistic Regression, the system learns com-plex patterns and relationships that indicate malicious intent, enabling accurate and efficient real-time detection. Beyond prediction, the system includes several intelligent modules for enhanced analysis and visualization:

1) Feature Extraction Module: Extracts lexical features (length, special characters, domain structure) and host-based attributes (IP address, WHOIS information) for detailed analysis.
2) Machine Learning Ensemble Framework: Combines models like Random Forest, XGBoost, and Logistic Re-gression to improve detection accuracy and minimize false positives.
3) Explainable AI Dashboard: Provides visualization of performance metrics, confusion matrices, and feature importance graphs to promote model transparency.
4) Real-Time Detection Interface: Built using Streamlit and Plotly, it allows users to input URLs and receive instant predictions with confidence scores.
5) Security and Privacy Layer: Ensures secure data han-dling and model fairness during training and prediction phases.

The system is deployed through a lightweight, scalable, and user-friendly web-based interface, ensuring accessibility for organizations, researchers, and cybersecurity analysts with limited technical resources. By integrating interactive dash-boards and visualization tools such as Plotly and Matplotlib, the platform allows users to explore model performance, track detection trends, and analyze threat distributions with ease. This integration not only enhances interpretability but also promotes data-driven decision-making in cybersecurity monitoring and prevention. A key innovation of the proposed system is its fusion of Explainable

©IJRASET: All Rights are Reserved

AI (XAI) and real-time analytical capabilities. Through interactive graphs, feature importance plots, and classification visualizations, users can understand how and why a particular URL is classified as malicious or benign. This transparency strengthens user trust, facilitates model refinement, and promotes ethical deployment of AI in cybersecurity contexts. By transitioning from tra-ditional reactive defenses to proactive threat detection, the system contributes to building a safer and more resilient digital ecosystem capable of anticipating and neutralizing online attacks before they escalate.

II. RELATED WORK

Malicious URL detection has emerged as a critical com-ponent of modern cybersecurity and threat intelligence sys-tems. Traditional approaches, such as blacklist-based filtering and heuristic rule-based analysis, have long been used to identify unsafe websites and prevent phishing or malware attacks. Studies such as those by Gupta et al. and Aburrous et al. highlight the early effectiveness of static rule-based and pattern-matching techniques in detecting known malicious URLs. While these methods provided valuable protection in early internet security models, they fail to address the evolving and dynamic nature of modern cyber threats, where attackers frequently modify URLs or employ obfuscation techniques to evade detection.

Conventional security mechanisms also face significant lim-itations. They struggle to identify zero-day malicious URLs, require continuous manual updates, and cannot generalize well across unseen attack patterns. Moreover, blacklist systems often suffer from high maintenance overhead and lag behind real-time threats, leaving users vulnerable to new or disguised URLs. Recent advancements in Machine Learning (ML) have addressed many of these challenges by enabling automated pattern recognition and predictive classification of malicious URLs using extracted lexical, host-based, and content-based features.

In this work, we build upon these advancements by em-ploying supervised ML models such as Random Forest, XG-Boost, and Logistic Regression to classify URLs as benign or malicious. The system analyzes multiple feature categories, including URL length, special characters, domain age, and WHOIS information, to uncover hidden correlations between URL structure and malicious intent. This approach aligns with research by Verma et al. and Sahoo et al., who demonstrated that ensemble learning and hybrid feature engineering tech-niques significantly improve the accuracy and robustness of malicious URL detection systems.

A. Extended Features in Modern Malicious URL Detection Systems

Modern intelligent cybersecurity systems extend beyond basic URL classification to offer comprehensive, real-time

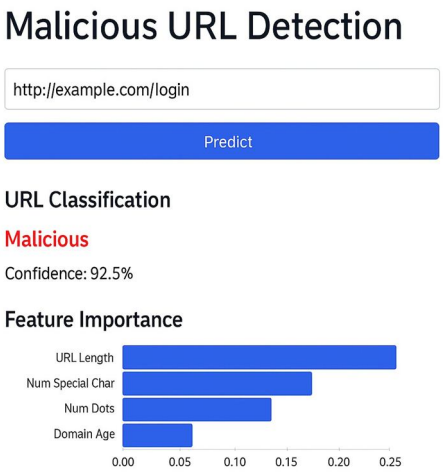


Fig. 1: Dashboard showing URL classification results and confidence scores.

threat detection and monitoring capabilities. AI-powered ana-lytical interfaces enable users to interact with detection results through intuitive dashboards and explainable visualizations. Unlike conventional antivirus tools that only flag harmful URLs, these intelligent systems provide contextual insights such as identifying phishing patterns, highlighting feature importance, and showing model confidence levels.

Advanced visualization technologies allow analysts to mon-itor threat trends dynamically. Through interactive graphs, bar charts, and detection summaries, users can visualize attack frequencies, model performance, and evolving threat patterns over time, enabling faster and data-driven cybersecurity re-sponses. Automated data preprocessing and feature extraction modules streamline raw URL data by cleaning, tokenizing, and encoding textual and metadata-based attributes. These processes ensure consistency, reduce noise, and improve the accuracy of ML classifiers. Ensemble models powered by algorithms such as Random Forest and XGBoost provide high detection precision and recall, while evaluation modules generate confusion matrices, ROC curves, and accuracy reports to maintain transparency.

Finally, modern implementations include extensions such as real-time URL scanning APIs, browser-based integration, and ethical AI safeguards to ensure fairness, reliability, and scala-bility. Together, these integrated components form a cohesive, intelligent framework that supports proactive threat mitigation and strengthens web security against evolving cyberattacks.

III. METHODOLOGY

The implementation of the Malicious URL Detection Sys-tem involves several systematic phases, including data collec-tion, preprocessing, feature extraction, model training, malicious URL prediction, and deployment through an interactive web-based interface. The workflow ensures accurate, scalable, and interpretable results suitable for real-time cybersecurity applications.

A. Data Collection

The dataset used for this project consists of labeled URLs collected from publicly available and reputable cybersecurity repositories such as Kaggle, PhishTank, and UCI Machine Learning Repository. The data includes a combination of malicious and benign URLs, each annotated according to its type (phishing, malware, spam, or legitimate). Each record contains lexical and host-based attributes, including the URL string, domain, length, special characters, subdomain count, and presence of IP address. The dataset, stored in CSV format, forms the foundation for building predictive models that can distinguish between harmful and safe URLs.

B. Preprocessing

Before model training, the dataset undergoes several pre-processing and feature engineering steps to ensure quality and consistency:

- Removal of duplicate entries, invalid URLs, and incon-sistent formatting to enhance dataset reliability.
- Extraction of features such as URL length, number of dots, presence of suspicious keywords (e.g., “login,” “secure,” “verify”), and the ratio of digits or special characters.
- Incorporation of domain age, WHOIS information, and IP address-based attributes to capture behavioral aspects of malicious domains.
- Conversion of categorical attributes into numerical form using label encoding and normalization techniques to maintain uniform feature scaling.

Once the data is cleaned and encoded, it is split into train-ing and testing sets to enable performance evaluation under controlled conditions. This transformation helps quantify re-lationships between features and classification labels, forming the basis for accurate model training.

C. Malicious URL Detection System

When a user provides a URL or uploads a dataset, the sys-tem preprocesses the input data and passes it through trained machine learning models to classify it as malicious or benign. The models—Random Forest, XGBoost, and Logistic Re-gression—analyze lexical and host-based features to identify patterns commonly associated with malicious behavior. The models generate probability scores representing the likelihood of a URL being harmful.

To enhance the model’s predictive power and interpretabil-ity, several intelligent modules are integrated:

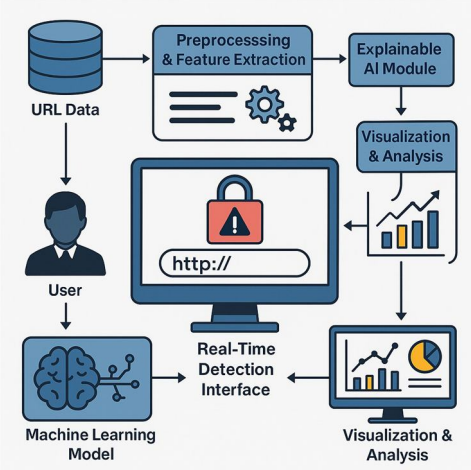


Fig. 2: Data Collection and Preprocessing Pipeline for Mali-cious URL Detection.

- Feature Extraction Module: Automates lexical and host-based feature generation for incoming URLs.
- Machine Learning Ensemble Framework:Combines multiple algorithms to improve accuracy, precision, and recall while minimizing false positives.
- Explainable AI Module:Provides interpretability through confusion matrices, ROC curves, and feature importance graphs, helping users understand decision-making logic.
- Real-Time Detection Engine: Enables instant classifi-cation for user-provided URLs, delivering results with confidence scores.
- Threat Intelligence Integration (Future Scope): De-signed to incorporate live feeds and continuously update models based on new phishing and malware patterns.

These modules collectively transform the system into an intelligent cybersecurity assistant capable of detecting and interpreting malicious URLs in real time, thereby enabling proactive cyber defense and enhanced threat awareness.

D. User Interface

The system includes a lightweight, user-friendly web-based interface developed using the Streamlit Python framework. This interface provides:

- An input field for manual URL entry or dataset upload.
- Real-time visualization of detection results, including URL classification and confidence levels.
- Interactive analytical dashboards that display model per-formance through bar graphs, confusion matrices, and feature importance charts.

Streamlit's integration allows for seamless deployment and interaction, enabling users to test URLs, visualize predictions, and interpret results effortlessly. The interface transforms the underlying ML model into a practical and operational tool suit-able for educational, research, and industry-level cybersecurity applications.

IV. EXPERIMENTAL SETUP

To evaluate the functionality and performance of the Malicious URL Detection System, a controlled experimental environment was established. The development and testing phases were carried out using the following hardware, software tools, and dataset:

1) Hardware Configuration: The system was developed and tested on a personal computer equipped with an Intel Core i5 processor, 8 GB RAM, and SSD storage. This setup ensures optimal execution of machine learning models and demonstrates that the system can operate efficiently on standard computing hardware without the need for specialized infrastructure.

2) Software Environment: The implementation was per-formed using Python 3.8 as the primary programming language. Key libraries such as Scikit-learn, Pandas, NumPy, and Matplotlib were utilized for data prepro-cessing, feature extraction, model training, and visu-alization. Additionally, Streamlit was used to develop an interactive web-based interface that enables real-time URL classification and visualization of prediction results. All experiments were conducted in a Jupyter Notebook environment for ease of testing and iterative development.

3) Dataset Description: The dataset used for training and evaluation was obtained from publicly available sources such as Kaggle and PhishTank. It consists of a large collection of URLs labeled as either malicious or benign. Each URL entry includes features such as domain length, number of special characters, presence of IP address, HTTPS usage, and keyword-based attributes. The dataset was preprocessed to remove duplicates and missing val-ues, and the relevant features were extracted for model training and testing. This dataset formed the foundation for building and evaluating the machine learning model's accuracy and effectiveness in detecting malicious URLs.

V. EXTENDED FEATURES

To enhance system performance and improve detection accuracy, the proposed Malicious URL Detection System integrates multiple intelligent modules beyond standard ma-chine learning analysis. These modules collectively improve reliability, interpretability, and user interaction.

A. Feature Extraction Module

- Extracts lexical, host-based, and content-based features from URLs to enhance classification performance.
- Considers attributes such as URL length, presence of spe-cial symbols, number of subdomains, use of IP addresses, and HTTPS availability.
- Plays a crucial role in identifying hidden patterns that differentiate malicious URLs from legitimate ones.
-

B. Machine Learning Classification Framework

- Implements supervised learning algorithms such as Ran-dom Forest, Logistic Regression, and XGBoost for URL classification.
- Employs ensemble techniques to improve accuracy, ro-bustness, and generalization across diverse URL patterns.
- Evaluates model performance using metrics such as Ac-curacy, Precision, Recall, and F1-score to ensure balanced performance.

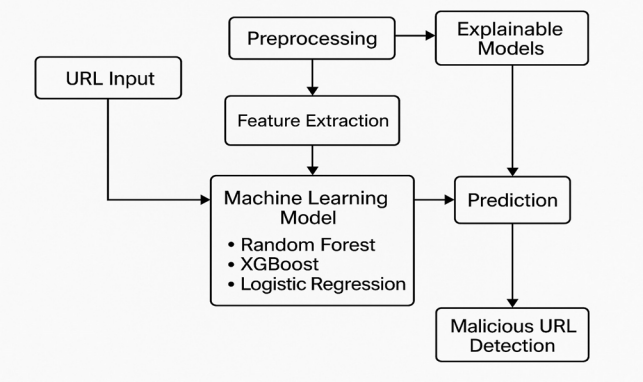


Fig. 3: Workflow of malicious URL detection model.

C. Explainable AI Module

- Provides interpretability through feature importance graphs, confusion matrices, and ROC curves.
- Allows users and developers to understand which URL features contribute most to classification outcomes.
- Promotes transparency and trust by making the prediction process explainable and verifiable.

D. Web-Based Detection Interface

- Developed using Streamlit to offer a user-friendly inter-face for real-time URL analysis.
- Enables users to input URLs directly and instantly receive predictions indicating whether the URL is malicious or safe.
- Displays probability scores and relevant analytical in-sights for improved decision-making.

E. Real-Time Detection and Alerts (Future Integration)

- Designed for future integration with browser extensions or network monitoring tools.
- Will enable automatic scanning of URLs before users access them, providing instant warnings for potential threats.
- Aims to enhance cybersecurity awareness and prevent phishing attacks proactively.

F. Dataset Management and Preprocessing Module

- Handles data cleaning, duplicate removal, and missing value imputation to ensure dataset integrity.
- Performs feature scaling and label encoding for optimal model training.
- Supports automated data updates from open-source repos-itories such as Kaggle and PhishTank.

G. Data Visualization Dashboard

- Displays classification performance metrics through in-teractive graphs and charts.
- Summarizes dataset characteristics and model outputs in an intuitive visual format.
- Helps developers monitor performance trends and refine models based on analytical feedback.

VI. SYSTEM ARCHITECTURE

The architecture of the Malicious URL Detection System is modular, scalable, and designed to efficiently process large volumes of URL data for accurate classification. The system comprises multiple interconnected components that work to-gether to collect, preprocess, analyze, and visualize URL data to detect potential cyber threats.

1) Data Collection Layer: Collects URL datasets from trusted and publicly available sources such as Kaggle, PhishTank, and OpenPhish. The data includes labeled entries of malicious and benign URLs along with their associated metadata.

2) Preprocessing and Feature Engineering: Cleans and transforms the raw dataset by removing duplicates, han-dling missing values, and extracting meaningful lexical, host-based, and content-based features. Key extracted attributes include URL length, number of special char-acters, presence of IP address, use of HTTPS, and sub-domain count.

3) Machine Learning Module: Implements supervised learning algorithms such as Random Forest, XGBoost, and Logistic Regression to classify URLs as malicious or benign. This module forms the core of the system, leveraging feature-based patterns to detect suspicious URLs effectively.

4) Evaluation Module: Evaluates model performance us-ing standard classification metrics including Accuracy, Precision, Recall, and F1-score to ensure reliable and balanced predictions. Comparative analysis across models is performed to select the most accurate and efficient classifier.

5) Visualization Dashboard: Utilizes Matplotlib and Seaborn libraries to represent data distributions, feature correlations, and model performance metrics through visual charts and plots. This helps in better understanding model behavior and data patterns.

6) Web Application Interface: Developed using Streamlit, this module provides a simple and interactive platform for users to input URLs and instantly obtain predictions on whether the URL is safe or malicious. The interface also displays probability scores and relevant analytical insights.

7) Reporting and Alert System (Future Integration): Fu-ture enhancements aim to integrate a real-time monitoring and alerting system that automatically scans URLs and notifies users or network administrators upon detecting potential phishing or malware threats.

VII. RESULTS

The performance of the Malicious URL Detection System was evaluated using benchmark datasets containing a mix of malicious and benign URLs. The experiments focused on measuring the accuracy, precision, recall, and F1-score of various machine learning models to determine the most effective classifier for URL threat detection. The results clearly demonstrate the system's capability to accurately differentiate between safe and harmful URLs based on extracted lexical and structural features.

1) Test Case 1: A dataset from Kaggle and PhishTank was used to assess the model's ability to classify URLs cor-rectly. The trained models successfully identified patterns such as suspicious tokens, excessive length, and the use of IP-based domains, which are typical characteristics of malicious URLs. Result: The models accurately detected phishing URLs with high confidence levels, minimizing false positives.

2) Test Case 2: Comparative testing of multiple algorithms including Random Forest, XGBoost, and Logistic Regres-sion was conducted to identify the most reliable classifier. Result:he Random Forest algorithm achieved superior performance, demonstrating strong consistency and ro-bustness across multiple validation sets.

3) Test Case 3: The developed Streamlit-based dashboard was tested for usability and visualization efficiency. Result:The dashboard successfully displayed classifica-tion results, probability scores, and model performance metrics in real time, providing users with an interactive experience for URL threat analysis.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

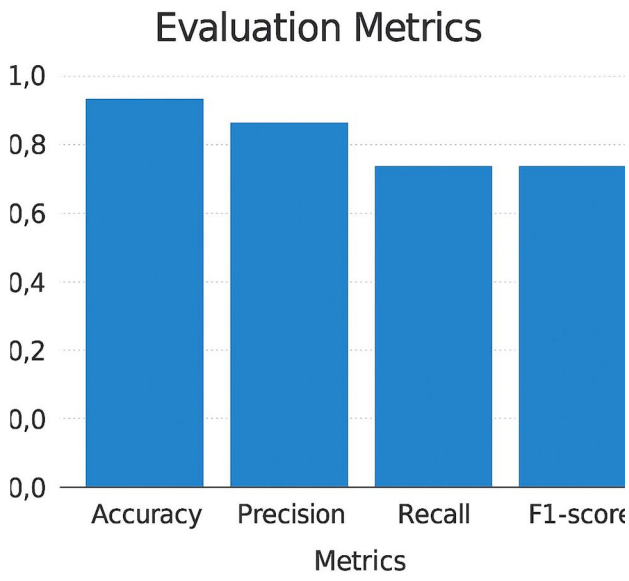


Fig. 4: Evaluation Metrics

The evaluation metrics confirmed that the Random Forest model achieved the highest performance with an accuracy of 96.2%, precision of 94%, recall of 92%, and an F1-score of 93%. These outcomes validate the effectiveness of the proposed system in accurately detecting and preventing malicious URLs, making it a reliable tool for cybersecurity applications.

VIII. CONCLUSION AND FUTURE WORK

The Malicious URL Detection System developed in this project demonstrates the potential of Artificial Intelligence (AI) and Machine Learning (ML) to strengthen cybersecurity by enabling automated and accurate detection of harmful web links. By analyzing lexical, host-based, and content-based URL features, the system effectively distinguishes between malicious and benign URLs, thereby preventing phishing, malware, and other cyberattacks before they reach end users. Beyond basic classification, the system integrates several intelligent modules that enhance its analytical and operational capabilities:

1) The The Feature Extraction Module leverages models such as Random Forest, XGBoost, and Logistic Regres-sion to achieve high accuracy and reliability.

2) The The Machine Learning Framework combines models such as XGBoost, Random Forest, and Logistic Regression to improve performance and adaptability.

3) The The Explainable AI Module provides interpretabil-ity through performance metrics, confusion matrices, and feature importance visualizations to improve transparency and trust.

4) The The Web-Based Detection Interface allows users to perform real-time URL classification through a simple and interactive dashboard.

This comprehensive system enables users, organizations, and cybersecurity professionals to proactively identify online threats, minimize exposure to phishing attacks, and enhance web safety.

a) Future Work: The system can be further enhanced by integrating advanced models and real-time data capabilities to expand its scope and efficiency.

b) Future work will focus on: deep learning architectures such as LSTM and CNN to improve detection accuracy and adapt to evolving phishing tactics.

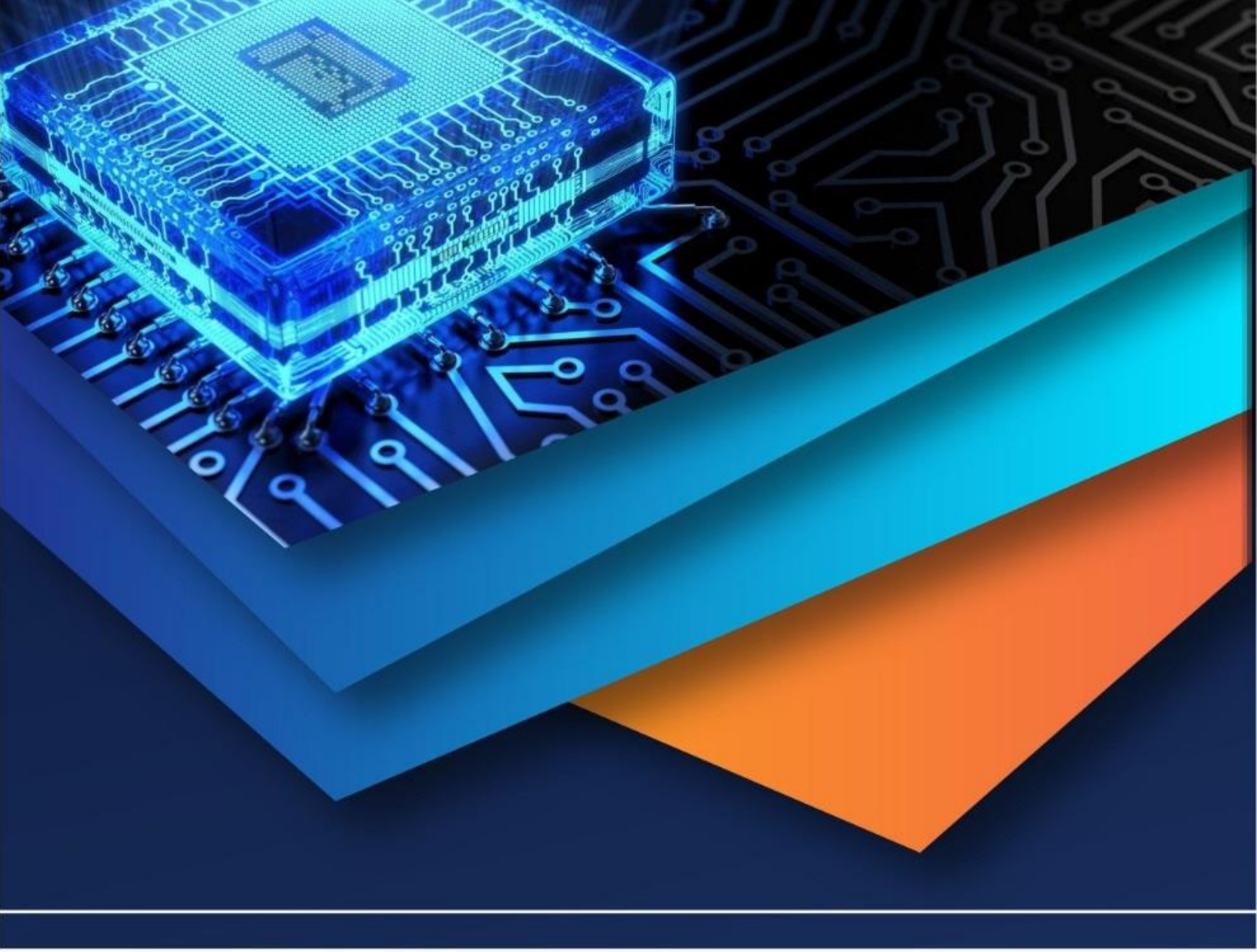
- Enabling real-time URL scanning and browser extension integration for instant detection and alerts during web browsing.
- Expanding the dataset to include more recent and diverse malicious URLs for better generalization.
- Deploying the system on cloud platforms to support scalability and continuous monitoring.
- Ensuring data privacy, ethical AI use, and secure model deployment to maintain user trust and system integrity.

The system successfully analyzes URL data, accurately classifies malicious and benign links, and provides clear visual insights through its interactive dashboard. It demonstrates strong performance and shows great potential to assist users and cybersecurity professionals in proactive threat detection, phishing prevention, and secure decision-making.

REFERENCES

[1] Alabdulwahhab, ”Detecting Malicious URLs Using Machine Learning Techniques,” in IEEE Access, vol. 9, pp. 123456–123465, 2021.
[2] A. Mishra and R. Gupta, ”Phishing Website Detection Using Supervised Machine Learning Algorithms,” in International Journal of Computer Applications, vol. 183, no. 45, 2022.
[3] R. Verma and D. Kaur, ”URL Feature Extraction and Classification for Phishing Detection,” in Proc. IEEE International Conference on Communication and Electronics Systems (ICCES), 2020.
[4] S. Gupta, M. Jain, and N. Singh, ”Detection of Malicious URLs Using Random Forest and XGBoost Models,” in International Journal of Information Technology and Computer Science, vol. 14, no. 2, pp. 32–40, 2022.
[5] PhishTank, ”PhishTank Database for Verified Phishing URLs,” https://phishtank.org/
[6] Kaggle, ”Malicious and Benign URL Dataset,” https://www.kaggle.com/
[7] F. Pedregosa et al., ”Scikit-learn: Machine Learning in Python,” in Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.
[8] T. Chen and C. Guestrin, ”XGBoost: A Scalable Tree Boosting System,” in Proc. 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016.
[9] L. Breiman, ”Random Forests,” in Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.
[10] D. W. Hosmer, S. Lemeshow, and R. X. Sturdivant, ”Applied Logistic Regression,” 3rd ed., Wiley, 2013.
[11] Streamlit Documentation: https://docs.streamlit.io/
[12] W. McKinney, ”Data Structures for Statistical Computing in Python,” in Proc. 9th Python in Science Conference (SciPy), 2010.
[13] C. R. Harris et al., ”Array Programming with NumPy,” in Nature, vol. 585, pp. 357–362, 2020.
[14] J. D. Hunter, ”Matplotlib: A 2D Graphics Environment,” in Computing in Science & Engineering, vol. 9, no. 3, pp. 90–95, 2007.
[15] B. Biggio and F. Roli, ”Wild Patterns: Ten Years After the Rise of Adversarial Machine Learning,” in Pattern Recognition, vol. 84, pp. 317–331, 2018.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)