International Scientific Journal of Engineering and Management
Volume: 04 Issue: 01 | Jan – 2025
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata
ISSN: 2583-6129
DOI: 10.55041/ISJEM02214

Real-Time Scam URL Detection System Using Decision Tree Classifier and TF-IDF Vectorization

1st Dr. T. Amalraj Victorie ,2rd M. Vasuki ,3rd D.Hemasujithaa and 4thR.Keerthana,

1 Professor, Department of computer Applications, Sri Manakula Vinayagar Engineering College (Autonomous), Puducherry
605008, India amalraj@smvec.ac.in

2Associate Professor, Department of computer Applications, Sri Manakula Vinayagar Engineering College (Autonomous), Puducherry 605008, India
dheshna@gmail.com

3Post Graduate student, Department of computer Applications, Sri Manakula Vinayagar Engineering College (Autonomous), Puducherry 605008, India
dhemasujithaa@gmail.com

4Post Graduate student, Department of computer Applications, Sri Manakula Vinayagar Engineering College (Autonomous), Puducherry 605008, India
keerthanaram13@gmail.com

ABSTRACT

The Scam URL Alert System is a web-based application designed to help users determine the legitimacy of URLs by analyzing them for potential threats. The system leverages a machine learning model, specifically a Decision Tree Classifier, to classify URLs as either safe ("legit") or unsafe ("fake").The process begins with the user inputting a URL into the application. Using advanced techniques like tokenization and TF-IDF vectorization, the model extracts features from the URL and predicts its legitimacy. The backend, built with Python's Flask framework, facilitates seamless communication between the user interface and the machine learning model. The system provides an intuitive and dynamic user experience with a visually appealing design, including a custom background image and responsive updates.

Keyword: URL classification, Phishing prevention , Decision TreeClassifier , TF-IDF vectorization, Cybersecurity, Legitimate URL verification, Sound alert.

1. INTRODUCTION

The Scam URL Alert System is a cybersecurity solution designed to detect and alert users about malicious or phishing URLs in real time. As internet usage grows, phishing attacks pose a significant threat by tricking individuals into revealing sensitive information such as login credentials, credit card details, and personal data. This system leverages machine learning to classify URLs as either safe (legitimate) or unsafe (fake), providing users with a simple yet powerful defense mechanism.

Using a decision tree classifier trained on a robust dataset of legitimate and malicious URLs, the system ensures high accuracy. It processes URLs through custom tokenization and TF-IDF vectorization to extract meaningful patterns, enhancing detection capabilities. To prioritize user safety, the system offers immediate feedback: safe URLs are displayed with a green message, while unsafe URLs trigger a red warning message and an audio alert to ensure prompt notification.

ISSN: 2583-6129
DOI: 10.55041/ISJEM02214

Built with Python and Flask, the Scam URL Alert System provides a user-friendly, responsive web interface where users can easily input URLs for verification. Additional features, such as automatic browser launch and responsive design, improve accessibility. The system is designed for scalability, enabling continuous model updates with new datasets to detect emerging phishing techniques, including obfuscated URLs. Its modular architecture supports seamless integration with other applications like chat platforms or web browsers, offering proactive phishing prevention. By combining accuracy, usability, and adaptability, this project addresses the pressing need for real-time protection against online threats.

2. PROBLEM STATEMENT:

The widespread adoption of the internet has led to a surge in phishing attacks, where attackers lure users into sharing private information through deceptive websites. These fraudulent sites often replicate the appearance of genuine ones, making it difficult for users to identify the threat. Relying on manual methods to detect such malicious links is inefficient, error-prone, and time-consuming, leaving individuals and organizations vulnerable to risks like data theft, financial fraud, and privacy breaches.

There is a pressing need for an automated system that can swiftly and accurately identify harmful URLs. Such a solution must not only utilize advanced techniques like machine learning to analyze and classify URLs but also ensure a seamless and user-friendly experience. By providing real-time alerts and actionable insights, this system can significantly reduce the impact of phishing attempts, offering a proactive approach to online security.

3. LITERATURE SURVEY:

Phishing and scam websites have become a growing concern in the realm of cybersecurity, prompting extensive research on techniques to detect and mitigate them. Several researchers have made significant contributions to this field. [1] Fette, I., Sadeh, N., & Tomasic, A. (2007) introduced one of the earliest detection methods using rule-based systems. Their approach relied on predefined patterns, such as abnormal URL lengths, multiple subdomains, and the presence of special characters like @, -, and //. These features helped identify phishing URLs, but the method suffered from limited adaptability to new threats, leading to higher false-positive rates. [2] Sheng, S., Wardman, B., Warner, G., Cranor, L., Hong, J., & Zhang, C. (2009) proposed blacklist-based systems to maintain a database of known phishing URLs. Their work highlighted that incoming URLs could be checked against this list to block access to malicious sites. While effective for previously known phishing URLs, blacklist-based systems were found to be ineffective against zero-day phishing attacks, where attackers altered URL structures to evade detection. [3] Ma, J., Saul, L., Savage, S., & Voelker, G. (2009) pioneered the use of machine learning-based phishing detection systems. Their work introduced supervised learning methods to classify URLs using features like URL length, subdomains, and lexical patterns. Their research compared the performance of Decision Trees, Logistic Regression, and Support Vector Machines (SVMs). Among these, the Decision Tree Classifier stood out for its simplicity, interpretability, and competitive accuracy. This contribution is crucial to the Scam URL Alert System, which also employs a Decision Tree for its classification process.[4]Marchal, S., Armano, G., Frisoni, S., Giacinto, G., & Kounelis, I. (2014) advanced the field by introducing the use of lexical and host-based features for phishing detection. Their research focused on real-time classification, where URL features such as the number of dots (.), domain length, and the presence of suspicious words (like "login" or "verify") were extracted and analyzed. Their real-time classification system is significant for modern phishing detection tools, including the Scam URL Alert System, which incorporates these real-time feedback capabilities. [5] Basnet, R., Sung, A., & Liu, Q. (2012) introduced the concept of using Natural Language Processing (NLP) for URL analysis. Their work emphasized the use of tokenization to break URLs into meaningful components, such as domain names, subdomains, and query strings. They also proposed using TF-IDF vectorization to transform these tokenized features into numerical vectors, which were then fed into machine learning models. This approach allowed classifiers to recognize hidden patterns in URL structures and enhanced the model's ability to detect unseen phishing URLs. The Scam URL Alert System leverages this idea by incorporating tokenization and TF-IDF as core

ISSN: 2583-6129
DOI: 10.55041/ISJEM02214

components of its feature extraction process, allowing for more robust classification. [6] Verma, R. & Das, A. (2017) focused on real-time phishing detection systems. Their research proposed fast, lightweight feature extraction methods that utilized regex patterns and simplified NLP-based processing. Their system enabled users to receive instant feedback on URL legitimacy, which is a key feature of the Scam URL Alert System. By integrating these real-time detection capabilities, the system ensures that users are alerted before accessing phishing websites, enhancing security and user experience.

4. ARCHITECTURE DIAGRAM

The process begins when the user opens the index.html page in the browser and enters a URL in the input field, then clicks the "Find" button. The frontend (in index.html) sends the URL to the Flask web application through a POST request. The Flask web server (backend) receives this URL, processes it, and forwards it to the Site Prediction Model for classification. The Site Prediction Model (in site_prediction.py) loads the dataset (urls.csv), preprocesses it, and uses the trained Decision Tree Classifier to analyze and classify the URL as either "legit" or "fake." Based on the model's prediction, the Flask application returns the result to the frontend. If the URL is deemed legitimate, it is displayed in green; if the URL is identified as fake, it is displayed in red and accompanied by an alarm sound (audio.mp3) to alert the user of the potential threat.

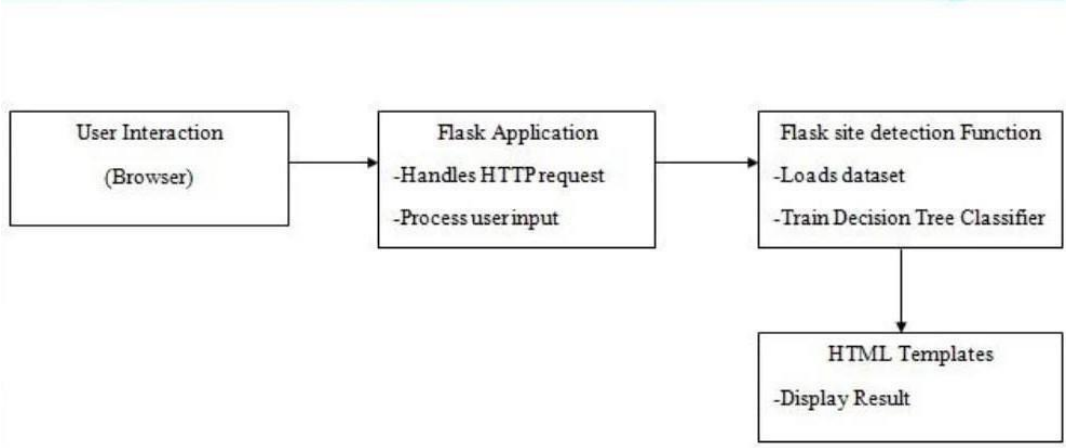

Fig 1: Scam URL Alert System Architecture

5. PROPOSED TECHNIQUE

Step 1: Data Preparation:

Tokenization: The URL is split into meaningful components, such as protocol, domain, subdomain, and path. This allows for a granular analysis of URL characteristics.

TF-IDF Vectorization: Each token from the URL is converted into a numerical feature using Term Frequency-Inverse Document Frequency (TF-IDF). This method assigns importance to unique tokens, enabling the system to distinguish between benign and malicious URLs.

Step 2: Model Architecture:

Input Layer (User Interface): Users input a URL via a web-based interface built with HTML, CSS, and JavaScript. The input URL is sent to the backend via AJAX/HTTP POST requests.

Backend Layer (Flask Server): The Flask server processes input and facilitates communication between the UI and

ISSN: 2583-6129
DOI: 10.55041/ISJEM02214

the machine learning model. It sends the URL to the feature extraction module and returns the prediction result.

Feature Extraction Layer: Tokenization splits the URL into components (protocol, domain, path, etc.). TF- IDF Vectorization converts tokens into numerical feature vectors, emphasizing important URL tokens.

Classification Layer (Decision Tree Classifier): Uses a Decision Tree Classifier to classify URLs as "legit" or "fake". The model is trained on labeled URL datasets, using learned decision rules to classify new URLs.

Prediction Layer (Result Display): The classification result is displayed on the web interface with real- time feedback. Users are notified if the URL is safe (legit) or unsafe (fake).

Step 3: Training the model:

The Decision Tree Classifier is a machine learning algorithm that splits data into subsets based on the feature values, aiming to achieve maximum information gain or reduce uncertainty. The key concept behind its working is the splitting criterion used to divide the data at each node. The most commonly used criteria are Gini Impurity and Information Gain (based on entropy).

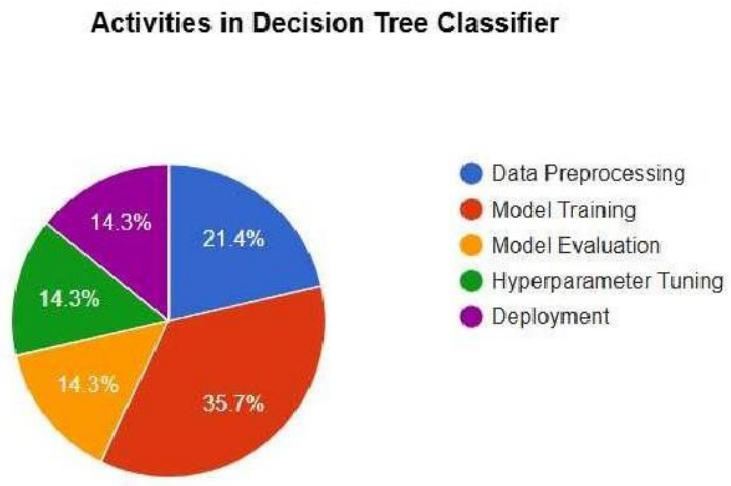

Fig 2: Pie Chart for Decision tree classifier

1. Gini Impurity Formula:

The Gini Impurity is used to measure the "impurity" or disorder of a node. The formula for Gini Impurity at a node t is:

International Scientific Journal of Engineering and Management
Volume: 04 Issue: 01 | Jan – 2025
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata
ISSN: 2583-6129
DOI: 10.55041/ISJEM02214

$$\text{Gini}(t) = 1 - \sum_{i=1}^{\mathbb{C}} P_i^2 \quad [1]$$

Where:

- C is the number of classes.
- pi is the probability of an element being classified into class ii.

The lower the Gini Impurity, the better the classification at that node.

Example: Gini Impurity

URL	CLASS
Abc.com	Fake
xyz.com	Legit
123.com	Legit
Fakeurl.com	Fake

At one step in the decision tree, we have a node with the following distribution:

- 2 Fake URLs
- 2 Legit URLs

Step 1: Calculate Probabilities
For this node:
Probability of "Fake" ($pfake$) : 2/4=0.5
Probability of "Legit" ($plegit$) : 2/4=0.5 Step 2: Apply the Gini Formula Gini(t)=1- ($p^2 fake + p^2 legit$)
Substitute the probabilities:
Gini(t)=1-($0.5^2 + 0.5^2$)

Gini(t) = 1 - (0.25 + 0.25)

Gini(t) = 1 - 0.5

Gini(t) = 0.5

6. DISCUSSION AND RESULT

International Scientific Journal of Engineering and Management
Volume: 04 Issue: 01 | Jan – 2025
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata
ISSN: 2583-6129
DOI: 10.55041/ISJEM02214

Fig 2: Fake URL Alert message

The output illustrates the Scam URL Alert System's detection of a suspicious URL ("httpps.gooogle.com"). After the user enters the URL and clicks the "Find" button, the system analyzes it and identifies it as unsafe. The result is displayed as "THE LINK IS FAKE IT IS UNSAFE TO USE" in red text, clearly warning the user about the malicious nature of the URL. This output emphasizes the system's effectiveness in protecting users by flagging and alerting them to potentially harmful links.

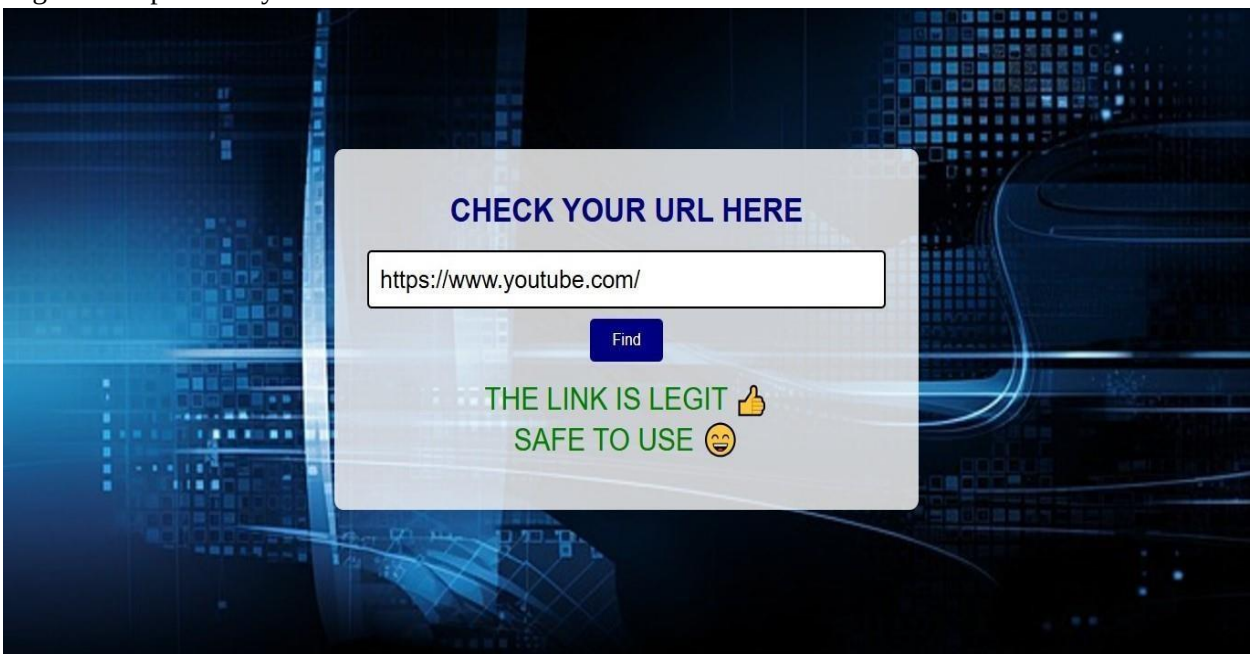

Fig 3: Safe URL confirmation

The output demonstrates the functionality of the Scam URL Alert System, where the user enters a URL (e.g., "https://www.youtube.com/") in the input box and clicks the "Find" button. The system processes the URL using a trained machine learning model to analyze its legitimacy. In this case, the system confirms the URL is safe and displays the result as "THE LINK IS LEGIT SAFE TO USE " in green text, providing a clear and reassuring

ISSN: 2583-6129
DOI: 10.55041/ISJEM02214

message. This output highlights the system's ability to identify legitimate URLs effectively, ensuring user safety through an intuitive and visually engaging interface.

7. CONCLUSION AND FUTURE ENHANCEMENT

The Scam URL Alert System addresses the growing challenge of detecting malicious URLs in real time, providing users with an efficient and user-friendly solution. By leveraging Decision Tree Classifiers and TF-IDF Vectorization, the system accurately classifies URLs as "fake" or "legit". Its interactive design, featuring color-coded feedback and audio alerts, enhances the user experience while raising awareness of online security threats. This project demonstrates the practical application of machine learning in cybersecurity, offering a scalable and efficient approach to phishing detection. While the system shows promising results, further improvements could focus on expanding the URL dataset and incorporating more advanced machine learning models to detect sophisticated phishing tactics.

Future enhancements include integrating advanced algorithms like Random Forests or Neural Networks for better adaptability and detection of complex URL patterns. Real-time URL monitoring could be implemented by integrating with browsers or mobile apps, allowing users to receive immediate alerts while browsing. Additionally, developing a browser extension or mobile app would enable instant URL checks, enhancing user convenience and accessibility. Finally, dataset expansion and continuous learning are crucial for adapting to new phishing techniques, ensuring that the system stays up-to-date with the evolving threat landscape.

REFERENCES

1. Chakraborty, P., & Shankar, S. (2020). Exploring Machine Learning Algorithms for Phishing Website Detection: A Comprehensive Survey. Published in the International Conference on Computing, Communication, and Networking Technologies (ICCCNT), this work evaluates different machine learning approaches used for identifying phishing websites, offering insights into effective solutions.

2. Jangra, A., & Soni, P. (2019). Utilizing Machine Learning for Phishing URL Detection. This study, featured in the International Journal of Computer Applications (IJCA), investigates machine learning models, comparing their accuracy and efficiency in detecting harmful URLs.

3. Mishra, S., & Prasad, K. (2020). Hybrid Approaches to Phishing Detection Using Machine Learning. Published in the Journal of Computer Networks and Communications, this paper discusses how combining machine learning models enhances phishing detection accuracy, with practical implications for cybersecurity.

4. Ming, F., & Zhang, T. (2020). Challenges and Advancements in Phishing Detection Systems. This IEEE Access article reviews the state of phishing detection technologies, addressing current limitations and suggesting future improvements.

5. Noble, J., & Solovyev, D. (2018). Comparative Analysis of Machine Learning Techniques in Phishing Website Identification. Published in the International Journal of Computer Science and Information Security (IJCSIS), the study highlights the strengths and challenges of different approaches in detecting fraudulent sites.

6. Dhanasekaran, R., & Vasudevan, S. (2019). Analyzing Machine Learning Algorithms for Detecting Malicious URLs. This paper in the Journal of Cyber Security and Privacy evaluates multiple algorithms,

ISSN: 2583-6129
DOI: 10.55041/ISJEM02214

providing insights into their comparative performance and practical usability.

7. Sharma, S., & Chaurasia, V. (2020). Improving Phishing Detection Through Ensemble Learning. Featured in the Proceedings of the 2nd International Conference on Computational Intelligence and Data Science (ICCIDS), this research demonstrates how ensemble methods increase detection accuracy.

8. Bhardwaj, A., & Kumar, A. (2020). Effective Detection of Phishing Websites Using Advanced Algorithms. Published in the International Journal of Computer Science and Information Security (IJCSIS), this paper explores innovative techniques to classify phishing websites and discusses their application.

9. Meena, S., & Bhuvanesh, S. (2020). Designing a Hybrid Model for Malicious URL Detection. Published in the Journal of King Saud University - Computer and Information Sciences, this study introduces a hybrid approach to enhance the accuracy and reliability of phishing URL detection systems.