Analyzing URL Legitimacy via Random Forest Classification

A N V K Swarupa[1], Kiran Kumar Pulamolu[2], I Usha[3], A Durga Bhavani[4]

[1,3] Asst.Prof, Department of CSE, Sasi Institute of Technology and Engineering, Tadepalligudem, West Godavari,Andhra Pradesh, India.

[2] Professor, Department of CST, Sasi Institute of Technology and Engineering, Tadepalligudem, West Godavari,Andhra Pradesh, India.

[4] Assoc.Prof, Department of CSE, Sasi Institute of Technology and Engineering, Tadepalligudem, West Godavari,Andhra Pradesh, India.

ABSTRACT

Bad URLs are growing more and more common, which poses a severe threat to internet security as they serve as a primary entrance point for identity theft, malware distribution, and phishing. Traditional URL detection systems primarily use signature based techniques, but these methods often prove in effective again fast evolving threats. This study explores the application of machine learning techniques to enhance dangerous URL identification and classification. The system combines advanced feature extraction techniques with cutting-edge machine learning algorithms to efficiently and accurately detect malicious URLs. In order to capture crucial characteristics that distinguish dangerous from benign URLs, the research entails collecting and preprocessing a range of URL datasets, training and evaluating numerous machine learning models, and feature engineering. Specifically, we use Random Forest, XG Boost, and Light GBMs algorithms in sentiment analysis to find potentially harmful URLs. These algorithms were chosen because they can detect patterns that indicate fraudulent behavior and perform well with big data sets. With an astounding accuracy percentage of 99.2%, Random Forest outperforms the other algorithms when algorithm performance is measured by accuracy. However, with respective accuracy rates of 94.5%, the XG Boost and Light GBM algorithms also exhibit excellent performance.

Keywords: Light GBM Classifier, Random Forest, XGBoost, Blacklists, Accuracy, and Performance evaluation.

INTRODUCTION

The internet has become an essential part of our daily lives, it also presents significant risks most notably, the threat of malicious URLs. These harmful links can deceive users into visiting phishing sites designed to steal sensitive information, distribute malware, or cause other forms of damage. Traditional methods like blacklisting, which rely on maintaining databases of known harmful URLs, have limited effectiveness. Cybercriminals can easily create new URLs to bypass these static defenses.

Machine learning (ML) offers a powerful alternative for tackling these evolving threats. By integrating ML algorithms with tools like Streamlit a Python based framework for building interactive web applications it's possible to develop efficient malicious URL detection systems. Streamlit allows for the rapid creation of user-friendly interfaces, enabling users to input URLs for analysis. Features like text boxes, buttons, and progress bars can be added to enhance

the user experience.

Recent advancements in cyber security highlight the growing importance of accurately detecting and neutralizing threats from malicious URLs. ML techniques play a critical role in this effort by enabling the analysis of large datasets and uncovering patterns that indicate malicious intent. Foundational research by Ma et al.
Building on this, studies like those by Gowtham & Tripathy (2017) evaluated the effectiveness of algorithms such as Random Forest and Gradient Boosting, stressing the importance of feature selection and model performance. Sahoo et al. (2017) offered a comprehensive survey of ML techniques for URL detection, classifying approaches by feature types and dataset properties, and pointing to future research directions. Mohammad et al. (2014) explored innovative strategies using self-structuring neural networks, while Marchal et al. (2012) introduced real-time systems like PhishStorm that focus on scalability and proactive threat response.

If a URL matches an entry, it is flagged as malicious; otherwise, it is assumed safe. However, due to the rapid creation of new URLs by attackers, maintaining a complete and current blacklist is nearly impossible.

Collectively, these studies underscore the transformative impact of machine learning in malicious URL detection. As cyber threats continue to evolve, ML-based systems hold the potential to significantly improve detection accuracy and strengthen defenses across the digital landscape.

RELATED WORKS

Numerous studies have explored the use of machine learning (ML) techniques for detecting and classifying malicious URLs, leveraging a range of features, models, and datasets to improve accuracy and real-time performance.

Feature Extraction and Model Diversity-
Sophisticated systems begin with extracting lexical, content-based, and network-based features to classify URLs. Various models such as Random Forest (RF), Support Vector Machine (SVM), Deep Belief Networks (DBN), and Convolutional Neural Networks (CNN)—have been used to distinguish between benign and malicious URLs. A similarity reasoning stage has also been integrated to enhance classification accuracy. Performance metrics (precision, recall, F1-score) were used to evaluate effectiveness on datasets such as PhishTank, Malware Domain List, and Alexa. [1]

Multi-class Classification with Large Datasets -
Using a dataset of over 650,000 URLs, some systems classified URLs into four categories: Phishing, Benign, Defacement, and Malware. Algorithms like RF, LightGBM, and XGBoost showed high accuracy, with RF achieving 96.6%. The goal was to improve real-time detection and integrate warning mechanisms into search engines. [2]

Comparative Studies with Traditional Classifiers-
Studies using datasets of up to 450,000 URLs compared classifiers such as Logistic Regression, SGD, KNN, Naïve Bayes, and Decision Trees. RF consistently demonstrated the highest detection accuracy. Emphasis was placed on balanced datasets to reduce bias and improve reliability. [3]

Lightweight Classifiers and URL Structure Analysis-Some works used basic ML models (e.g., KNN, Logistic Regression) with a focus on structural features like URL length and presence of characters such as '@'. KNN achieved over 90% accuracy. Future improvements include incorporating HTML and JavaScript content analysis. [4]

Word Embeddings and Cost-Sensitive Models-Advanced approaches combined domain-engineered features with word embeddings to capture deeper semantic patterns. Cost-sensitive neural networks outperformed traditional classifiers, achieving over 99% precision, especially in imbalanced datasets. Techniques like SMOTE were used for balancing. [5]

Large-Scale Detection and Feature Selection-On datasets containing millions of URLs and features, classifiers such as RF, SVM, MLP, and Naïve Bayes were evaluated. Feature selection techniques (e.g., correlation-based filtering) enhanced performance without requiring domain expertise. RF and MLP were top performers. [6]

Novel Frameworks (e.g., Markov Detection Tree)-Some studies introduced unique models like Markov Detection Tree (MDT), combining decision trees with Markov decision processes. These approaches focused on webpage attributes and entropy-based classification, achieving high precision on labeled datasets. [7]

Host and Lexical Feature-Based Classification-Using UCI datasets, models based on host and lexical features employed RF and Gradient Boosting classifiers to achieve up to 98.6% accuracy. These works emphasized refining feature extraction and enhancing cyber security applications. [8]

Multi-modal Approaches-

Integrating both textual and visual webpage features using CNNs, some systems improved detection performance. Combining CNN outputs in a neural network reduced false positives and boosted MCC, illustrating the benefit of multi-modal analysis. [9]

Phishing-Specific Detection on Login URLs-A unique focus on login URLs (Phishing Index Login URL dataset) using TF-IDF and Logistic Regression achieved 96.5% accuracy. The study stressed the importance of updated datasets and suggested incorporating visual content for future improvements. [10]

PROPOSED METHODOLOGY

The workflow diagram illustrating the experimental design of the model developed to predict whether a URL is benign, malware, phishing, or defacement. The fig 1 outlines the process of training and testing various models to identify the one that yields the most accurate URL classification results.

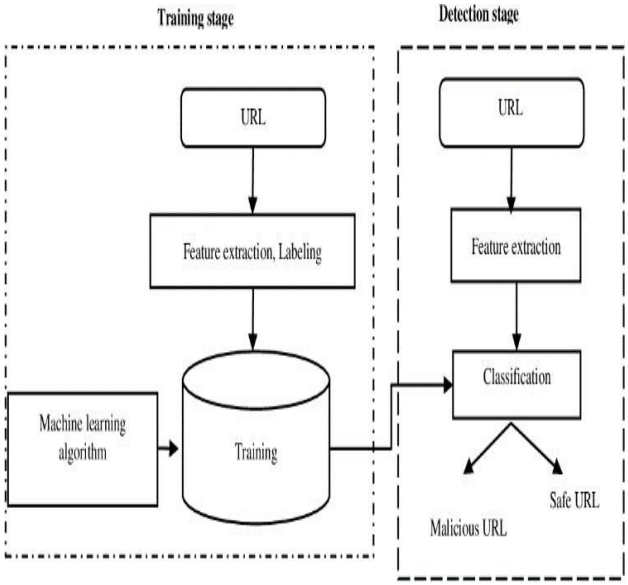

Fig 1 Proposed Methodology

i) Obtaining Data: A labeled dataset with a well organized structure is gathered from the Kaggle source. This dataset carefully classifies URLs as safe, dangerous, compromised, or infected with malware, making sure that no data is left out or is empty.

ii) Preprocessing and Data Cleaning: The data goes through a thorough purification procedure before being used to train the model. This entails applying normalization algorithms, carefully managing any possible missing information, and extracting useful supplementary features from the URLs. To guarantee consistency, numerical values are

standardized and categorical values are meticulously encoded.

iii) Training Models:

The model training process utilizes a range of powerful machine learning algorithms, including XGBoost, LightGBM, and Random Forest classifiers. Over 80% of the cleaned dataset is used for training, leveraging the Python scikit-learn library to implement these algorithms.

Model Validation and Optimization:

The remaining 20% of the dataset is reserved for validating the model's performance. This phase involves fine-tuning the model to achieve optimal performance metrics such as sensitivity, F1 score, recall, and overall accuracy through a rigorous evaluation and optimization process.

iv) Model Comparison:

Following the optimization phase, a comprehensive evaluation is performed to compare the performance of each machine learning classification technique on the validation set. This analysis allows for the identification of the model that delivers the highest accuracy and reliability in classifying URLs, ensuring the most effective and dependable detection system.

v) Dataset Analysis

The dataset used in this study was sourced from Kaggle and serves as a comprehensive resource for training and testing malicious URL detection systems. It comprises a total of 651,191 URLs, systematically categorized into four distinct classes:

Benign URLs (Safe):

This class contains over 428,000 URLs representing legitimate and trustworthy websites that pose no threat to users.

Defacement URLs:

With more than 96,000 entries, these URLs are associated with websites that have been compromised by attackers, often resulting in unauthorized content alterations.

Phishing URLs:

This category includes over 94,000 URLs designed to deceive users into revealing sensitive information such as login credentials or financial data.

Malware URLs:

Comprising over 32,000 instances, these URLs are specifically crafted to deliver malicious software onto a user’s device upon access.

Decoding URL Intent with Word Clouds

Word clouds are used as visual tools to represent the frequency of terms within each URL category. They provide intuitive insights into the nature and intent of URLs across different classes. The analysis of word clouds for each class reveals distinct patterns:

Benign URLs:

These word clouds prominently feature legitimate components such as "html", standard domain extensions like ".com" and ".org", and terms like "wiki", suggesting connections to reputable information sources.

Fig 2 Benign URL wordcloud

From Fig 2, the presence of the keyword "html" in the Benign URL word cloud indicates that many safe websites commonly include this term in their address or page

structure. This observation suggests that the presence of "html" could be a valuable feature for a Random Forest classifier, helping the model infer a higher probability of a URL being benign when this term appears in its textual representation.

For Phishing URLs, the word cloud prominently displays a mix of legitimate-looking terms such as "tools," "www," and "index", along with suspicious words like "battle" and "net". This combination reflects the deceptive strategy commonly used in phishing attacks—mimicking the appearance of trustworthy domains to mislead users, while masking malicious intent beneath superficially familiar elements.

Fig 3 Phishing URL Word Cloud Analysis

From Fig 3, the prominence of terms like "https" and ".exe" in the phishing URL word cloud suggests that malicious URLs often exploit secure connection prefixes and executable file extensions to appear legitimate or to lure users into downloading harmful files. Incorporating these features into the Random Forest model can help it better identify URLs containing such elements as potentially dangerous, thereby increasing the likelihood of correctly classifying phishing attempts.

For Malware URLs, the word clouds highlight a strong emphasis on ".exe" files, which signals the frequent use of executable files as a vector for malware distribution. Additionally, the presence of obfuscated or encoded strings such as "E7," "BB," and "MOZI" may indicate attempts to conceal malicious payloads, raising further red flags during detection.

Fig 4 Malware URL Word Cloud Analysis

From Fig 4, the noticeable presence of terms such as "info," "php," and "wp-content" in the Malware URL word cloud indicates that these URLs may be crafted to imitate legitimate websites or exploit known vulnerabilities in PHP-based platforms, such as WordPress. The Random Forest algorithm can leverage the frequency of these terms as distinguishing features, helping to flag URLs that exhibit potentially malicious behavior.

Defacement URL Word Cloud Analysis-
In the Defacement URL word cloud, commonly used web development terms like "index," "php," and "itemid" are prominently featured. This pattern suggests

that these URLs are designed to target website structures, often by injecting or manipulating code, which aligns with typical defacement activities. These features can serve as strong indicators for machine learning models when identifying attempts to alter or compromise website content.

Fig 5 Defacement URL Word Cloud Analysis

From Fig 5, the frequent occurrence of terms such as "https," ".exe," and "php" in the Defacement URL word cloud suggests a recurring pattern in how these malicious URLs are constructed. The use of secure protocols ("https"), executable file references (".exe"), and server-side scripting extensions ("php") indicates an attempt to mimic legitimate web structures while potentially delivering malicious payloads or enabling unauthorized modifications.

These elements can serve as strong discriminative features for the Random Forest model, aiding in the accurate classification of defacement URLs by identifying structural and content-based clues commonly associated with harmful intent.

vi) Feature Analysis

The performance of machine learning models in URL classification relies heavily on selecting meaningful features. This study groups the most effective features into five key categories that help distinguish between benign and malicious URLs:

- **URL Structure Features**
  Features like URL_Length, Hostname_Length, and TLD_Length evaluate the overall complexity of a URL. Longer or unusually structured URLs may indicate attempts to hide malicious content.
- **Character and Symbol Features**
  Attributes such as Digit_Count, Percent_Encoding_Count, and Dash_Count detect excessive use of numbers or symbols, which are often used to obfuscate harmful links.
- **Protocol and Redirection**
  Elements like HTTPS_Count, HTTP_Count, and Shortened_URL reveal how URLs use protocols and whether they are masked using URL shorteners—common in phishing attacks.
- **Suspicious Content Indicators**
  The presence of certain words (e.g., “login”, “secure”, “verify”) may signal attempts to mislead users and are strong indicators of phishing.

- **Domain and Lexical Features**
  Features such as IP_Presence, Subdomain_Count, At_Symbol_Count, and WHOIS_Anomaly help identify URLs that deviate from standard domain usage or attempt to impersonate trusted sources.

PROPOSED MACHINE LEARNING ALGORITHMS

To classify URLs as benign or malicious, three widely-used machine learning models were applied to a dataset split into 80% for training and 20% for testing:

Random Forest

Builds multiple independent decision trees using random feature subsets. The final classification is determined by majority voting. Known for high interpretability and resistance to over fitting.

LightGBM (Light Gradient Boosting Machine)

A fast and efficient gradient boosting model that builds trees sequentially, correcting previous errors. Offers better performance and speed than traditional methods like Random Forest.

XGBoost (eXtreme Gradient Boosting)

A powerful gradient boosting algorithm optimized for speed and scalability. Incorporates regularization techniques to prevent overfitting, making it suitable for complex classification tasks such as malicious URL detection.

Experimental Settings

The implementation was carried out on Google Colab, focusing on real-time URL threat detection. Key highlights include:

Feature Extraction: Included URL length, character patterns, HTTPS presence, and redirection behaviors.

Data Preparation: The dataset was thoroughly cleaned, balanced, and preprocessed to manage edge cases like homoglyph attacks and hidden redirects.

Model Optimization: Emphasis was placed on speed and accuracy, ensuring reliable classification.

User Interface: A Python-based tool was developed, enabling users to input URLs, receive instant classification, and view confidence scores.

Outcome: A robust, scalable, and efficient system capable of protecting against phishing and other online threats.

RESULTS AND ANALYSIS

The Random Forest classifier achieved the highest prediction accuracy of 99.2%, outperforming both:

•LightGBM: 95% accuracy

•XGBoost: 96.2% accuracy

This high performance is attributed to the effective integration of:

•Lexical features, capturing structural and semantic URL characteristics

•NLP techniques, particularly word cloud-based feature extraction, which enhanced pattern recognition

While lexical features posed challenges for LightGBM and XGBoost, leading to slight accuracy drops, Random Forest effectively handled these variations, demonstrating its robustness and superior capability in malicious URL classification.

Our experiments show that the Random Forest classifier outperforms other models, achieving a prediction accuracy of 99.2%. This is attributed to the effective integration of lexical features and NLP techniques like word cloud-based feature extraction.

Table 1 Model Comparison

Model	Accuracy	F1-Score
Random Forest	99.2%	0.95
LightGBM	95.0%	0.94
XGBoost	96.2%	0.94

While LightGBM and XGBoost also

demonstrated strong performance, certain lexical features led to reduced accuracy in these models. In contrast, Random Forest effectively managed feature variability, highlighting its robustness and reliability in malicious URL detection.

Visual Performance Evaluation

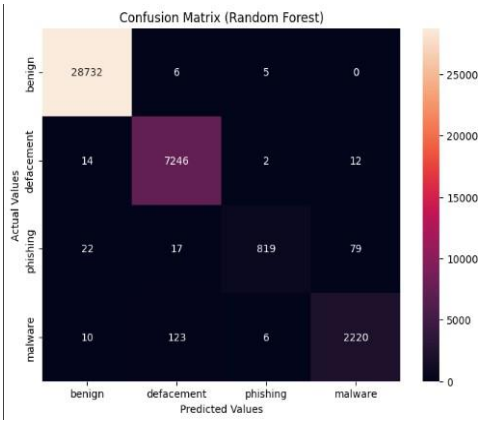

Fig 6 Confusion Matrix (Random Forest)
The confusion matrix indicates that the model accurately identifies most benign, defacement, phishing, and malware URLs. High values along the diagonal demonstrate strong classification performance, with only minor misclassifications.

	precision	recall	f1-score	support
benign	1.00	1.00	1.00	28743
defacement	0.98	1.00	0.99	7274
phishing	0.98	0.87	0.93	937
malware	0.96	0.94	0.95	2359
accuracy			0.99	39313
macro avg	0.98	0.95	0.97	39313
weighted avg	0.99	0.99	0.99	39313

accuracy: 0.992
recall: 0.953
precision: 0.981
f1: 0.966

Fig 7 Random Forest Accuracy Score
The model exhibits high precision and recall across all URL categories. With an overall accuracy of 99.2% and F1-score of 0.95, it proves to be reliable for distinguishing between safe and malicious URLs.

	precision	recall	f1-score	support
benign	0.97	0.99	0.98	85621
malware	0.97	0.99	0.98	19292
phishing	0.98	0.91	0.94	6504
defacement	0.91	0.83	0.87	18822
accuracy			0.96	130239
macro avg	0.96	0.93	0.94	130239
weighted avg	0.96	0.96	0.96	130239

accuracy: 0.962
recall: 0.932
precision: 0.956
f1: 0.943

Figure 8: LightGBM Accuracy Score
LightGBM shows strong results for benign,

	precision	recall	f1-score	support
benign	0.97	0.99	0.98	85621
malware	0.96	0.99	0.98	19292
phishing	0.97	0.91	0.94	6504
defacement	0.90	0.83	0.86	18822
accuracy			0.96	130239
macro avg	0.95	0.93	0.94	130239
weighted avg	0.96	0.96	0.96	130239

accuracy: 0.959
recall: 0.927
precision: 0.952
f1: 0.938

Fig 9 XGBoost Accuracy Score
XGBoost achieves comparable accuracy to LightGBM with 96.2% overall accuracy and an F1-score of 0.94, performing well across most categories but slightly trailing in defacement detection.

User Interface Evaluation

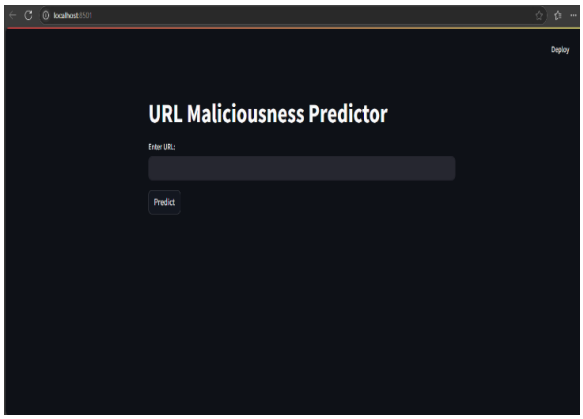

Fig 10 User Interface

The initial screen of the user interface clearly guides users, making the system intuitive and accessible.

Figures 11–14: URL Type Predictions

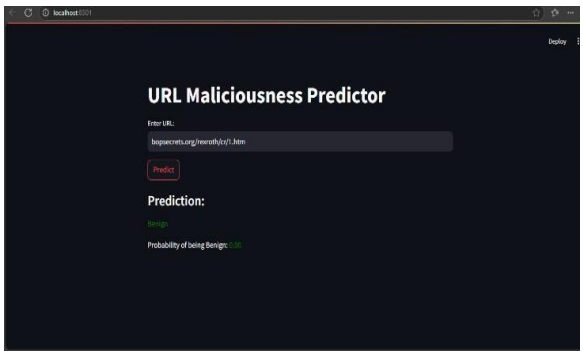

Fig11 Correctly predicts a Benign URL

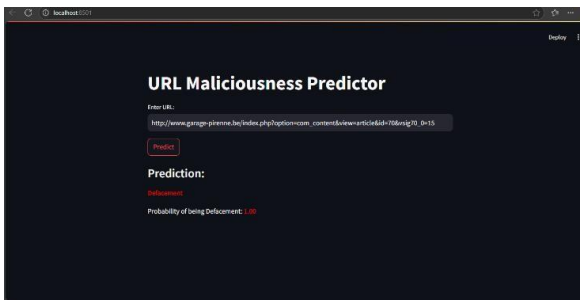

Fig 12 Identifies a Defacement URL

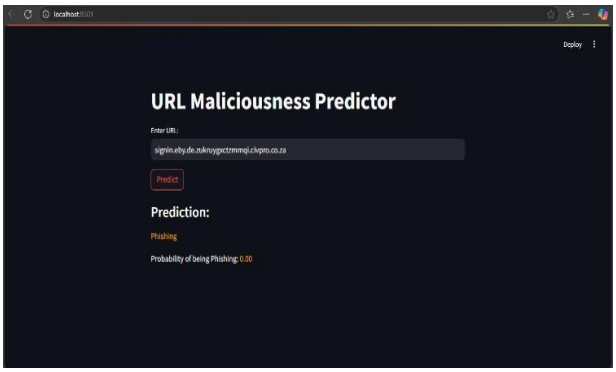

Fig 13 Classifies a Phishing URL

Fig 14 Malware URL

These results confirm the system's capability to distinguish URL types based on learned features and present outcomes clearly to the user.

Conclusion

In this study on malicious URL detection, we implemented and evaluated three advanced machine learning algorithms—Random Forest, XGBoost, and LightGBM—to determine the most effective model for accurately classifying URLs as malicious or benign.

After extensive experimentation and testing, the Random Forest algorithm emerged as the top performer, achieving an impressive accuracy of 99.2%. Its superior performance is largely due to its robustness, ability to handle complex and high-dimensional feature sets, and effectiveness in capturing subtle patterns that distinguish malicious URLs from legitimate ones.

While XGBoost and LightGBM also delivered strong results with accuracies of 96.2% and 95% respectively, they did not match the overall precision and reliability of the Random Forest model—particularly when dealing with intricate lexical features.

To make the system practical and user-friendly, we developed a web-based interface using Streamlit. This interface enables users to input URLs and receive real-time classification results with corresponding confidence scores, significantly enhancing the usability of the detection system.

Key Takeaways:

Random Forest is an ideal choice for malicious URL detection due to its accuracy, resilience to over fitting, and interpretability. The integration of lexical features and NLP techniques (e.g., word clouds) improved feature representation and detection performance.

User accessibility is enhanced through a simple and intuitive Streamlit-based web interface.

FUTURE SCOPE

Future research can focus on:

Optimizing model performance further through feature selection and tuning. Exploring hybrid models to combine the strengths of different algorithms. Enhancing detection of evasive threats such as zero-day phishing attempts or homoglyph attacks.

REFERENCES

[1] M. Aljabri *et al.*, "Detecting Malicious URLs Using Machine Learning Techniques: Review and Research Directions," *IEEE Access*, 2022.

[2] U. Shetty D. R., A. Patil, and M. Mohana, "Malicious URL Detection and Classification Analysis Using Machine Learning Models," *IEEE IDCIoT*, 2023.

[3] S. Shantanu, J. B., and J. A. Kumar R., "Malicious URL Detection: A Comparative Study," *IEEE ICAIS*, 2021.

[4] M. Mehndiratta *et al.*, "Malicious URL: Analysis and Detection using Machine Learning," *IEEE INDIACom*, 2023.

[5] A. Crișan *et al.*, "Detecting Malicious URLs Based on Machine Learning Algorithms and Word Embeddings," *IEEE*, 2020.

[6] F. Vanhoenshoven *et al.*, "Detecting Malicious URLs using Machine Learning Techniques," *IEEE*, 2016.

[7] J. Liu *et al.*, "A Markov Detection Tree-Based Centralized Scheme to Automatically Identify Malicious Webpages on Cloud Platforms," *IEEE Access*, 2018.

[8] F. O. Catak, K. Şahinbaş, and V. Dörtkardeş, "Malicious URL Detection Using Machine Learning," in Artificial Intelligence Paradigms for Smart Cyber-Physical Systems, IGI Global, 2020.

[9] M. Alsaedi *et al.*, "Multi-Modal Features Representation-Based CNN Model for Malicious Website Detection," *IEEE Access*, 2024.

[10] M. Sánchez-Paniagua *et al.*, "Phishing URL Detection: A Real-Case Scenario Through Login URLs," *IEEE Access*, 2022.