Enhanced Malicious URL Detection Using Machine Learning: A Comparative Analysis of Naive Bayes, SVM, Logistic Regression, and SDG Decision Tree Classifier

1. Sivaiah Borra
Email: bsivaiah@cmrcet.ac.in

2. Siddhartha Nampally
Email: siddharthanampally141003@gmail.com

3. Raju Basham
Email: bashamraju874@gmail.com

4. Mohammad Abdul Sameer
Email: Sameermohammedabdul555@gmail.com

CMR College of Engineering and Technology

CONTRIBUTIONS

Sivaiah Borra: Conceptualization, Data Collection, Feature Engineering, Model Implementation.
Siddhartha Nampally: Literature Review, Experimentation, Data Preprocessing, Evaluation Metrics.
Raju Basham: Algorithm Selection, Performance Analysis, Visualizations, Report Writing.
Mohammad Abdul Sameer: Code Optimization, Documentation, Hyperparameter Tuning, Reference Compilation.

***Abstract*—With the growing number of cyber threats, identifying and mitigating malicious URLs has become a critical challenge in cybersecurity. This study introduces a machine learning-driven approach for classifying URLs as either benign or harmful. We assess the effectiveness of four different classification algorithms: Naive Bayes, Support Vector Machine (SVM), Logistic Regression, and Stochastic Gradient Descent (SGD) Decision Tree Classifier. The model is trained on a dataset incorporating lexical, host-based, and content-based attributes. The findings illustrate that leveraging a combination of these extracted features alongside machine learning techniques enhances detection accuracy. This comparative evaluation offers valuable insights into the most efficient algorithm for practical cybersecurity applications.**

***Index Terms*—Malicious URL Detection, Machine Learning, Cybersecurity, Feature Engineering, URL Classification.**

I. INTRODUCTION

A Uniform Resource Locator (URL) serves as a reference to online resources, enabling users to access websites and web-based services. However, cybercriminals often manipulate URLs to mislead users, leading to various security risks such as phishing attacks, malware distribution, and drive-by downloads [2], [3]. Malicious URLs are crafted to redirect users to harmful sites, execute unauthorized scripts, or spread malicious software, posing significant threats to individuals and organizations alike.

To mitigate these threats, two main detection strategies have been developed: rule-based detection and machine learning-based detection. Rule-based detection relies on predefined heuristics and blacklists, which, although effective for known threats, struggle to adapt to evolving attack patterns and newly emerging threats [6], [8]. In contrast, machine learning-based techniques analyze URLs based on extracted features, enabling a more dynamic and adaptive approach to classification. These methods utilize various attributes, including lexical features, host-based properties, and behavioral characteristics, to enhance detection accuracy [1], [9].

This paper explores the efficacy of machine learning models in classifying URLs as either benign or malicious. Additionally, we propose an improved feature extraction method that enhances classification performance. By leveraging a combination of advanced feature engineering and machine learning classifiers, this study aims to contribute to the development of robust cybersecurity solutions capable of identifying and mitigating malicious URLs in real-time.

A. Motivation

The rapid growth of the internet and the increasing reliance on web-based services have made URLs a primary vector for cyberattacks. Malicious URLs are frequently used in phishing campaigns, malware distribution, and other cybercrimes, causing significant financial and reputational damage to individuals and organizations. According to recent reports, phishing attacks alone account for over 90% of data breaches, with malicious URLs being the most common delivery mechanism [5].

Traditional rule-based detection systems, such as blacklists, are limited in their ability to detect new and evolving threats. These systems rely on predefined signatures and heuristics, making them ineffective against zero-day attacks and sophisticated phishing techniques. Moreover, the sheer volume of

URLs generated daily makes it impractical to maintain and update blacklists in real-time [6], [8].

Machine learning-based approaches offer a promising solution to these challenges. By analyzing patterns in URL features, machine learning models can adapt to new threats and provide real-time detection capabilities. However, the effectiveness of these models depends heavily on the quality of feature extraction and the choice of algorithms. This study aims to address these challenges by proposing an enhanced feature extraction method and conducting a comparative analysis of multiple machine learning algorithms to identify the most effective approach for malicious URL detection.

The motivation for this research stems from the need for robust, scalable, and adaptive solutions to combat the growing threat of malicious URLs. By leveraging advanced machine learning techniques, this study seeks to contribute to the development of cybersecurity tools that can protect users from evolving cyber threats in real-time.

II. Related Work

The detection of malicious URLs has been extensively studied, with early research focusing primarily on signature-based methods. While effective against known threats, these approaches fail to detect zero-day attacks and newly emerging threats [2], [6]. Recent advancements have shifted toward machine learning-based techniques, which leverage lexical, host-based, and behavioral features to improve detection accuracy and enhance generalization capabilities [1], [9].

A. Signature-based Malicious URL Detection

Traditional approaches to malicious URL detection have relied on maintaining extensive lists of blacklisted URLs, which are used to flag harmful websites. When a user attempts to access a URL, it is checked against the blacklist, and if a match is found, the URL is classified as malicious. While this method is effective for well-known threats, it struggles to detect novel or evolving attacks that have not yet been blacklisted. Moreover, maintaining and updating these lists require significant resources, making this approach less scalable for real-time threat detection [6], [8].

B. Machine Learning-based Malicious URL Detection

Machine learning approaches offer a more adaptive solution for identifying malicious URLs by analyzing extracted features rather than relying on predefined signatures. These techniques can be broadly classified into three categories: supervised learning, unsupervised learning, and semi-supervised learning [1], [9].

Supervised learning models, such as Decision Trees, Support Vector Machines (SVMs), and Neural Networks, require labeled datasets for training. They excel in classification tasks but rely on the availability of high-quality labeled data. Unsupervised learning methods, in contrast, identify patterns and anomalies without requiring labeled data, making them particularly useful when labeled datasets are limited or unavailable. Semi-supervised learning combines both approaches, leveraging a small amount of labeled data alongside larger sets of unlabeled data to improve classification performance [10].

Additionally, some studies have explored advanced grouping techniques based on semantic and host-based attributes. These methods cluster related URLs based on shared properties, such as domain registration information and network behavior, to enhance detection accuracy. Despite their potential, machine learning-based methods face challenges related to data imbalance, evolving attack strategies, and computational efficiency [3], [7].

C. Limitations of Existing Approaches

Despite advancements in malicious URL detection, current methods still exhibit several limitations:

- **Lack of Adaptive Algorithm Selection:** Many existing systems do not incorporate adaptive techniques for selecting the most effective machine learning algorithms, which can lead to suboptimal performance in different threat landscapes [1].
- **Insufficient Feature Engineering:** URL feature extraction often lacks depth, failing to capture critical patterns associated with evolving cyber threats. Incorporating advanced feature selection techniques can enhance detection effectiveness [4].
- **Scalability Issues:** Many detection frameworks struggle to handle large-scale real-time data, limiting their deployment in high-traffic environments [5].
- **Inability to Address Concept Drift:** Cyber threats continuously evolve, rendering static models ineffective over time. Periodic model retraining and dynamic learning strategies are essential for maintaining detection accuracy [12].

Addressing these challenges requires continuous refinement of machine learning models, integration of real-time adaptive detection mechanisms, and improved feature engineering techniques to enhance robustness against emerging threats.

III. Proposed Methodology

Our proposed approach aims to accurately classify URLs as either benign or malicious using machine learning techniques. The system follows a structured pipeline that includes data acquisition, feature extraction, and classification through multiple machine learning models. By integrating various classifiers, we enhance the system's robustness and improve detection performance [1], [9].

A. Architecture

The architecture of the proposed system consists of the following key components [10]:

- Data collection from multiple sources
- Feature extraction from URLs
- Classification using different machine learning algorithms

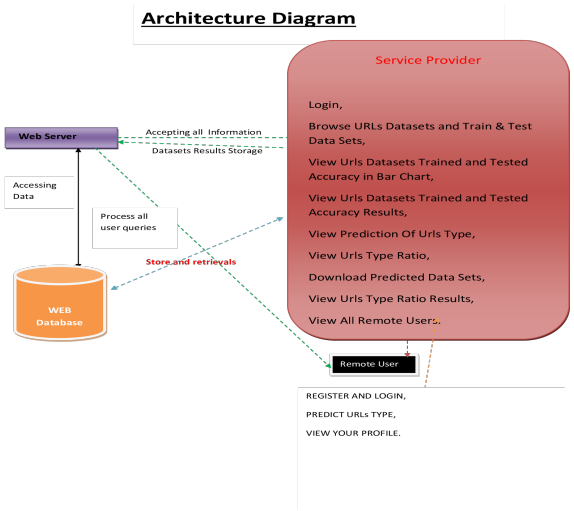

Fig. 1. Architecture of the Proposed System

B. Data Flow

The data flow diagram illustrates the stepwise progression of URLs through the system pipeline. Each stage processes the URL data sequentially, ensuring efficient classification into benign or malicious categories [7], [9].

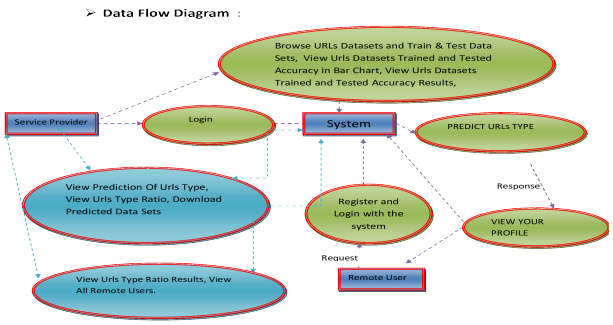

Fig. 2. Data Flow for Malicious URL Detection

C. Process Flow

The following steps outline the systematic process for malicious URL detection:

- **Data Acquisition:** Collect a dataset containing both benign and malicious URLs from publicly available sources such as PhishTank, URLHaus, and Alexa [16], [17].
- **Preprocessing and Feature Extraction:** Clean the collected data by eliminating duplicates and inconsistencies. Extract meaningful features from URLs, including lexical, host-based, and behavioral attributes [9], [10].
- **Model Training and Classification:** Utilize machine learning models, such as Naive Bayes, SVM, Logistic Regression, and SDG Decision Tree, to train on the extracted features. The trained models then classify new URLs based on learned patterns [1], [3].

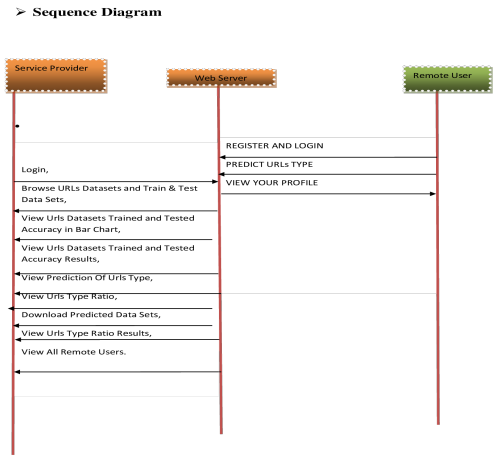

Fig. 3. Model Training and Classification Process

D. Flow Chart

The flowchart below illustrates the structured approach of the proposed system for malicious URL detection.

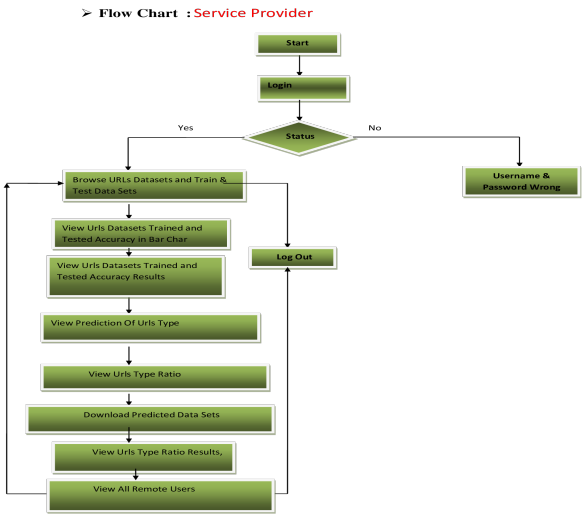

Fig. 4. Flow Chart of the Proposed System

E. Feature Extraction

Feature extraction is a critical aspect of URL classification. We categorize extracted features into three primary groups:

- **Lexical Features:** URL length, frequency of special characters, presence of numeric characters, and entropy measurements [9].
- **Host-based Features:** Domain information, including age, WHOIS registration, and DNS records [10], [12].
- **Behavioral Features:** Redirection patterns, presence of embedded links, and JavaScript execution behaviors [3], [13].

F. Machine Learning Models

The classification process involves training and evaluating multiple machine learning models:

- **Naive Bayes:** A probabilistic classifier based on Bayes' theorem [1].
- **Support Vector Machine (SVM):** A supervised learning model for binary classification [9].
- **Logistic Regression:** Predicts the probability of a URL being malicious [10].
- **Stochastic Gradient Descent (SDG) Decision Tree Classifier:** Combines SDG with decision tree learning for improved performance [14], [15].

To ensure reliable performance, we train and test these models using stratified cross-validation techniques. Hyperparameter tuning is applied to optimize model accuracy and efficiency [9].

IV. Experimental Results

The models were trained and evaluated on a dataset consisting of both benign and malicious URLs. Performance assessment was conducted using standard classification metrics, including accuracy, precision, recall, and F1-score, to provide a comprehensive analysis of model effectiveness [9], [10], [14].

A. Performance Metrics

To evaluate the performance of the classification models, we used the following metrics:

1) Accuracy: Accuracy measures the proportion of correctly classified instances (both true positives and true negatives) out of the total number of instances. It is calculated as:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TP} + \text{FP} + \text{FN}}$$

2) Precision: Precision measures the proportion of true positive predictions out of all positive predictions made by the model. It is calculated as:

$$\text{Precision} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Positives (FP)}}$$

3) Recall: Recall (also known as sensitivity) measures the proportion of true positives out of all actual positive instances. It is calculated as:

$$\text{Recall} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Negatives (FN)}}$$

4) F1 Score: The F1 score is the harmonic mean of precision and recall, providing a balanced measure of the model's performance. It is calculated as:

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

These metrics provide a comprehensive evaluation of the model's performance, ensuring that both false positives and false negatives are accounted for in the analysis.

B. Naive Bayes

The Naive Bayes classifier achieved an accuracy of 95.72%, demonstrating strong performance in classifying URLs based on probabilistic assumptions [1].

Category	Precision	Recall	F1-score	Support
0	0.96	0.99	0.97	8434
1	0.92	0.83	0.87	1832
accuracy	nan	nan	0.96	10266
macro avg	0.94	0.91	0.92	10266
weighted avg	0.96	0.96	0.96	10266

Fig. 5. Naive Bayes Model Confusion Matrix

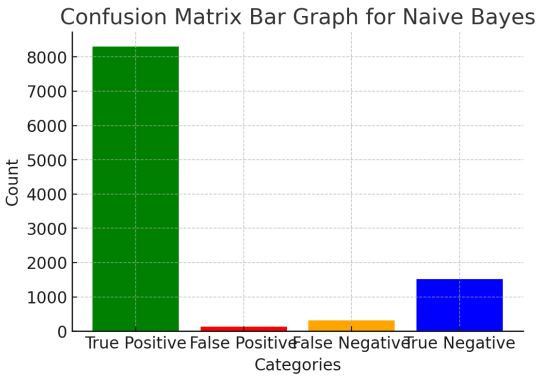

Fig. 6. Bar Graph of Confusion Matrix for Naive Bayes

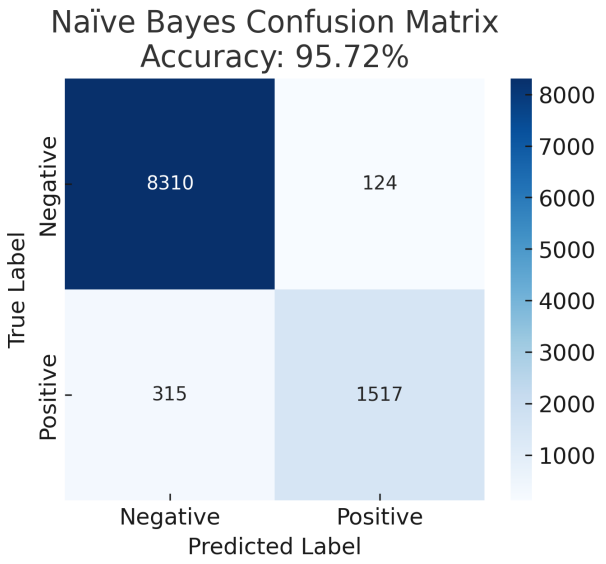

Fig. 7. Confusion Matrix for Naive Bayes

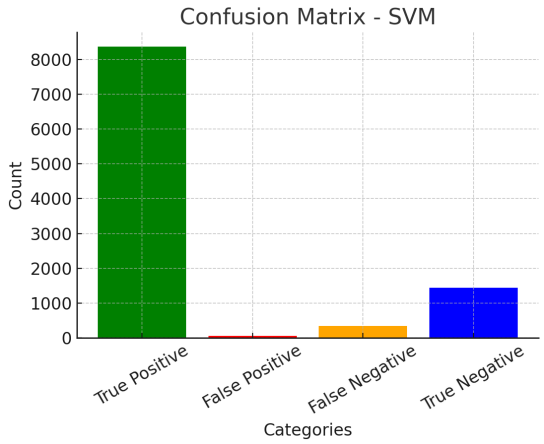

Fig. 9. Bar Graph of Confusion Matrix for SVM

C. SVM

The Support Vector Machine (SVM) model attained an accuracy of 96.12%, highlighting its capability to effectively separate malicious and benign URLs using hyperplanes [9].

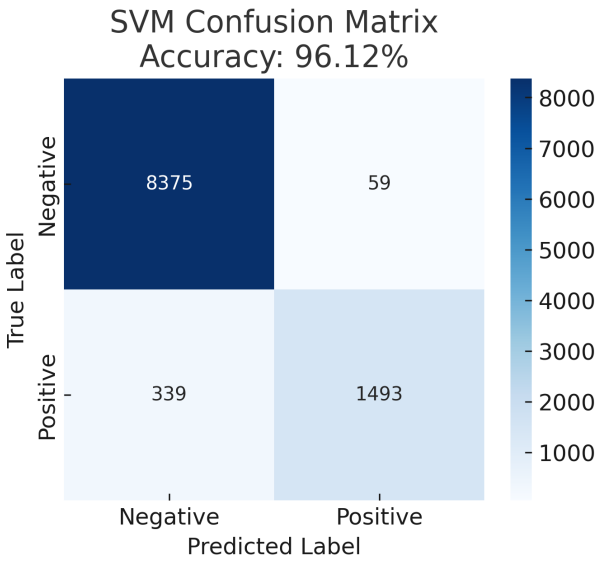

Fig. 10. Confusion Matrix for SVM

Class	Precision	Recall	F1-Score	Support
0	0.96	0.99	0.97	8434
1	0.92	0.83	0.87	1832
Accuracy			0.96	10266
Macro Avg	0.94	0.91	0.92	10266
Weighted Avg	0.96	0.96	0.96	10266

Fig. 8. SVM Model Visualization

D. Logistic Regression

Logistic Regression achieved an accuracy of 95.06%, making it a reliable model for URL classification due to its probabilistic nature and linear decision boundary [10].

Class	Precision	Recall	F1-score	Support
0	0.95	0.99	0.97	8434
1	0.96	0.75	0.85	1832
Accuracy	0.95	nan	nan	10266
Macro avg	0.95	0.87	0.91	10266
Weighted avg	0.95	0.95	0.95	10266

Fig. 11. Logistic Regression Model Visualization

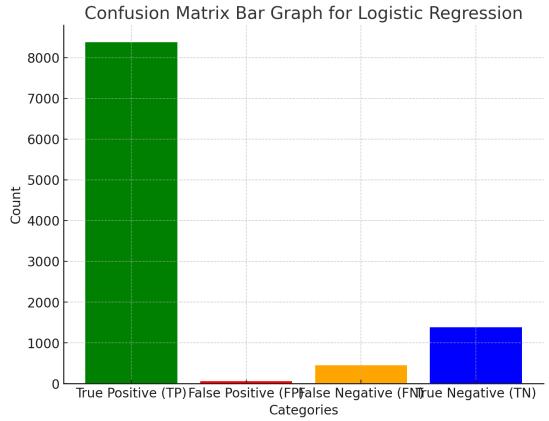

Fig. 12. Bar Graph of Confusion Matrix for Logistic Regression

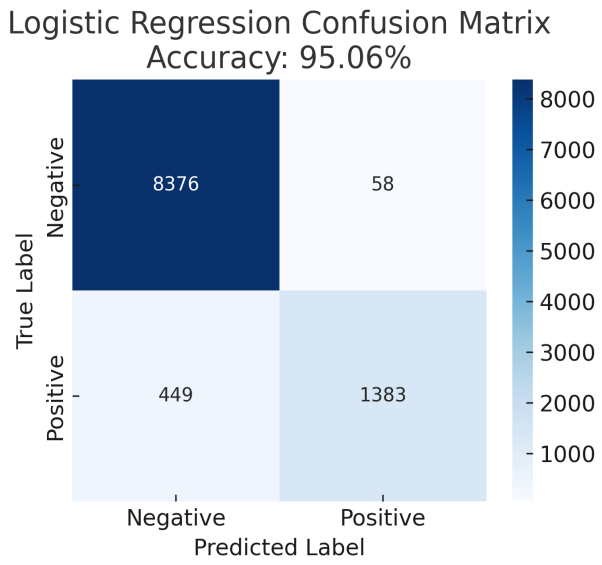

Fig. 13. Confusion Matrix for Logistic Regression

E. Decision Tree Classifier

The Decision Tree Classifier obtained an accuracy of 94.40%, showing its ability to make decisions based on feature splits, although it is slightly less robust than other models in handling complex URL patterns [14].

Category	Precision	Recall	F1-score	Support
0	0.96	0.99	0.98	8434
1	0.96	0.81	0.88	1832
Accuracy	nan	nan	0.96	10266
Macro avg	0.96	0.9	0.93	10266
Weighted avg	0.96	0.96	0.96	10266

Fig. 14. Decision Tree Classifier Model Visualization

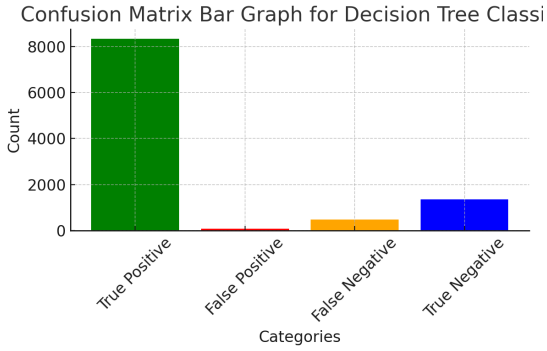

Fig. 15. Bar Graph of Confusion Matrix for SGD Decision Tree Classifier

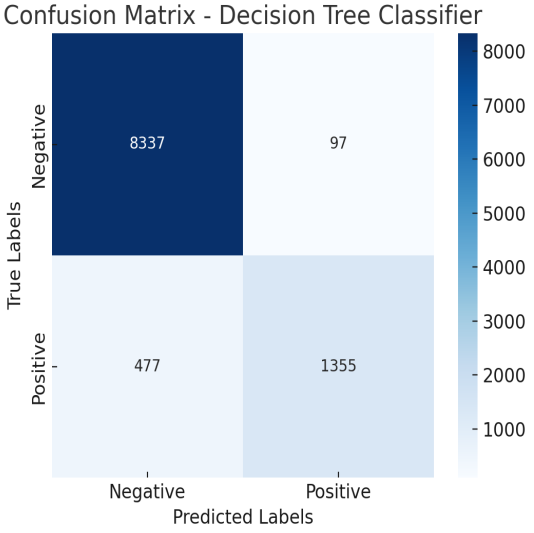

Fig. 16. Confusion Matrix for Decision Tree Classifier

The evaluation results indicate that SVM outperformed the other classifiers in terms of accuracy, followed by Naive Bayes,

Logistic Regression, and Decision Tree Classifier. The use of confusion matrices further supports the understanding of misclassification rates, helping to refine model selection for real-world malicious URL detection [9], [14].

V. Conclusion

This study highlights the effectiveness of machine learning techniques in identifying and classifying malicious URLs. By evaluating multiple classifiers, we observed that Support Vector Machine (SVM) exhibited the highest accuracy at 96.12%, making it the most suitable model for URL classification [9]. Naive Bayes, Logistic Regression, and Decision Tree Classifier followed closely, with varying performance based on precision and recall metrics [1], [10].

The experimental results underscore the importance of feature extraction in improving classification accuracy. Lexical, host-based, and behavioral features played a significant role in enhancing model predictions [9]. The confusion matrices further provided insights into misclassification rates, allowing us to refine detection mechanisms for better real-world applicability [14].

Future research can focus on integrating deep learning techniques such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) to improve classification accuracy further [15]. Additionally, implementing real-time detection frameworks and adaptive learning models will enhance the ability to identify emerging threats dynamically [10]. Expanding the dataset with more diverse URLs and improving feature engineering techniques can also contribute to increased robustness in malicious URL detection systems [16], [17].

References

[1] D. Sahoo, C. Liu, and S. C. Hoi, "A Survey on Malicious URL Detection Using Machine Learning," CoRR, vol. abs/1701.07179, 2017.

[2] M. Khonji, Y. Iraqi, and A. Jones, "A Comprehensive Review on Phishing Detection Techniques," IEEE Communications Surveys & Tutorials, vol. 15, no. 4, pp. 2091–2121, 2013.

[3] M. Cova, C. Kruegel, and G. Vigna, "Analysis and Detection of Drive-by-Download Attacks and Malicious JavaScript Code," in Proc. 19th Int. Conf. World Wide Web, ACM, 2010, pp. 281–290.

[4] R. Heartfield and G. Loukas, "A Review of Social Engineering Attacks and Defensive Mechanisms," ACM Computing Surveys, vol. 48, no. 3, p. 37, 2015.

[5] Symantec, "Internet Security Threat Report (ISTR) 2019." Available: https://www.symantec.com/content/dam/symantec/docs/reports/istr-24-2019-en.pdf [Accessed Oct. 2019].

[6] S. Sheng et al., "Empirical Evaluation of Phishing Blacklists," in Proc. 6th Conf. Email and Anti-Spam (CEAS), 2009.

[7] C. Seifert, I. Welch, and P. Komisarczuk, "Static Heuristic Detection of Malicious Web Pages," in ATNAC 2008, IEEE, pp. 91–96.

[8] S. Sinha, M. Bailey, and F. Jahanian, "Effectiveness of Reputation-based Blacklists," in Proc. Int. Conf. Malicious and Unwanted Software (MALWARE), IEEE, 2008, pp. 57–64.

[9] J. Ma et al., "Large-scale Online Learning for Identifying Suspicious URLs," in Proc. 26th Annu. Int. Conf. Machine Learning, ACM, 2009, pp. 681–688.

[10] B. Eshete, A. Villafiorita, and K. Weldemariam, "Holistic Detection of Malicious Web Pages," in Security and Privacy in Communication Networks, Springer, 2013, pp. 149–166.

[11] S. Purkait, "A Review of Phishing Countermeasures and Their Effectiveness," Information Management & Computer Security, vol. 20, no. 5, pp. 382–420, 2012.

[12] Y. Tao, "Log-based Detection of Suspicious URLs and Devices," Ph.D. dissertation, School of Computing Science, 2014.

[13] G. Canfora et al., "System Call-based Detection of Malicious Web Pages," in Availability, Reliability, and Security in Information Systems, Springer, 2014, pp. 226–238.

[14] L. Breiman, "Random Forests for Classification and Regression," Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.

[15] T. G. Dietterich, "An Overview of Ensemble Learning Methods," in Proc. Int. Workshop Multiple Classifier Systems, Cagliari, Italy, 2000, pp. 1–15.

[16] PhishTank, "Phishing Database API." Available: https://www.phishtank.com/developer_info.php [Accessed Nov. 2019].

[17] URLhaus, "Malware URL Database Dump." Available: https://urlhaus.abuse.ch/downloads/csv/ [Accessed Nov. 2019].

[18] Majestic, "Dataset of Top Million Websites." Available: http://downloads.majestic.com/majestic_million.csv [Accessed Oct. 2019].

[19] Kaggle, "Malicious and Non-Malicious URL Dataset." Available: https://www.kaggle.com/antonyj453/urldataset#data.csv [Accessed Nov. 2019].

[20] Google Drive, "Chrome.zip Dataset." Available: https://drive.google.com/file/d/13G_Ndr4hMFx_qWyTEjHuOyJmHFWD0Gud/view [Accessed Dec. 2019].