Experimental Evaluation of URL-Based Machine Learning Techniques for Phishing Website Detection

K. Nabeel Ur Rehman
Student, Department of
Information Technology (IT),
Shadan College of Engineering and Technology,
Affiliated to Jawaharlal Nehru Technological University
Hyderabad (JNTUH), Hyderabad, India
nabeelrehman1276@gmail.com

S. Saad Ali
Student, Department of
Information Technology (IT),
Deccan College of Engineering and Technology,
Affiliated to Osmania University (OU), Hyderabad, India
syedsaad3876@gmail.com

***Abstract* - Phishing websites pose a major security risk by using deceptive URLs to steal user information. This work presents a machine learning–based system that detects phishing sites using URL-derived features. The system is developed using a URL-based feature set and trained with supervised machine learning techniques to effectively distinguish phishing websites from legitimate ones. Logistic Regression, Decision Tree, XGBoost, and Random Forest classifiers were evaluated, with the Random Forest model demonstrating the most reliable performance and consequently selected for deployment. The system extracts selected URL features from user-input links and classifies them as "Legitimate" or "Phishing," while generating a probability-based confidence score for clearer user interpretation. The deployed web application performs consistently, achieving approximately 0.71 confidence in real-time predictions and correctly identifying URLs across all test cases. The solution remains lightweight and easy to use, with future improvements planned for real-time web scraping and extended model comparisons.**

Index Terms* – *Phishing Detection, Machine Learning, URL Features, Random Forest, Cybersecurity

I. INTRODUCTION

Phishing has evolved into one of the most persistent and damaging cyber threats, targeting millions of users through deceptive websites crafted to mimic legitimate online services. These attacks commonly aim to steal credentials, financial information, or personal data by tricking users into interacting with malicious links. The widespread accessibility of phishing kits and the minimal technical skill required to deploy such attacks have contributed to a significant increase in phishing incidents across global digital ecosystems [1]. This continued growth highlights a critical need for robust, accurate, and adaptive detection mechanisms capable of analyzing suspicious web activity in real time.

Traditional defense mechanisms such as URL blacklists, heuristic filters, and browser-based warnings have proven increasingly inadequate due to the fast-changing and highly dynamic nature of phishing websites. Studies indicate that attackers frequently modify URL structures, hosting patterns, and redirection behaviors to evade signature-based and rule-based detection techniques [1]. Because these methods depend heavily on manually curated datasets or fixed rules, they struggle to detect newly emerging or zero-day phishing URLs, leaving users vulnerable to sophisticated attacks.

Machine learning (ML) has emerged as an effective alternative to overcome these limitations by automatically learning discriminative features from large volumes of benign and malicious URL data. ML-based systems can analyze lexical, domain-based, and behavioral characteristics of URLs, enabling more accurate detection of phishing attempts even when adversaries modify superficial patterns [2]. Research also highlights that ML models can capture subtle relationships within URL features such as abnormal domain length, suspicious redirection paths, or inconsistent HTTPS usage that are often overlooked by traditional approaches [2]. These capabilities have positioned ML as a leading framework for developing intelligent and adaptive phishing detection systems.

Recent studies further emphasize the escalating sophistication of phishing techniques, including the increased use of HTTPS certificates, fast-flux hosting, and visually deceptive domain names. These evolving trends create new challenges for security practitioners, making it essential to design detection systems that can generalize well to unseen threats and adapt to changes in attacker behavior. According to emerging cybersecurity analyses, the continual refinement of phishing strategies demands advanced, flexible, and data-driven solutions capable of operating in dynamic online environments [3].

Motivated by these challenges, this work leverages URL-based machine learning methods to develop a practical phishing detection system suitable for lightweight deployment. By focusing on URL-derived attributes rather than full webpage content, the system enables faster inference, reduced computational overhead, and improved usability for real-time applications. This study aims to contribute to the growing research on ML-driven cybersecurity solutions by evaluating multiple ML algorithms, selecting the best-performing model, and integrating it into an accessible web-based detection platform.

II. RELATED WORK

Research on phishing detection has increasingly focused on analyzing URL-based characteristics and applying machine learning to classify malicious websites. Ali *et al.* [4] examined several machine learning models combined with wrapper-based feature selection and reported that selecting relevant URL attributes significantly improves the accuracy of phishing URL detection. Their findings emphasize the importance of engineered lexical and host-based features in enhancing classifier performance.

Safi *et al.* [5] conducted a detailed literature review on phishing detection methods and highlighted limitations in traditional approaches. They noted that heuristic and blacklist-based systems struggle against newly generated phishing URLs due to their static nature. Their review further emphasized that machine learning provides better adaptability by learning patterns directly from labeled datasets, making it more suitable for detecting evolving phishing attacks.

Yerima and Alzaylaee [6] explored deep learning models, particularly convolutional neural networks (CNNs), for URL-based phishing detection. Their work demonstrated that deep models can learn richer representations of URL structures. However, they also acknowledged practical challenges, including higher computational requirements and the need for larger datasets to achieve stable performance.

Krishna *et al.* [7] provided a survey focused specifically on machine-learning-based URL analysis for phishing detection. They reported that ensemble methods such as Random Forest and boosting algorithms consistently achieve strong and reliable performance across multiple datasets. Their survey supports the effectiveness of URL-driven feature analysis and reinforces the suitability of machine learning for lightweight, real-time phishing detection systems.

Overall, prior studies consistently show that machine learning techniques particularly those relying on URL feature engineering offer a practical and effective approach for phishing detection. These insights motivated the use of multiple machine learning models in this study and informed the final selection of Random Forest as the most reliable algorithm for the URL-based detection system.

III. METHODOLOGY

The methodology for phishing website detection in this work is centered on analyzing URL-based features and applying machine learning techniques to classify websites as legitimate or phishing. The approach follows a structured pipeline inspired by prior studies on intelligent web threat detection. Existing research emphasizes that phishing websites often exhibit distinctive URL characteristics, redirection behaviors, and hosting anomalies that can be systematically captured and modeled through feature-driven learning methods [8]. Motivated by these insights, the system focuses on extracting relevant URL attributes that reflect lexical, domain-related, and structural patterns commonly associated with phishing attacks.

To develop a reliable detection model, multiple machine learning algorithms were evaluated to determine the most effective classifier for the task. Previous work demonstrates the suitability of supervised learning algorithms in identifying phishing behavior, particularly when URL-based indicators are used as primary predictive features [9]. In alignment with these findings, the methodology involves training several algorithms including Logistic Regression, Decision Tree, XGBoost, and Random Forest on the processed dataset and comparing their performance. The emphasis is placed on selecting a model that balances accuracy, interpretability, and deployment efficiency.

Random Forest emerged as the most robust model among those tested, consistent with observations in earlier research where ensemble-based methods often outperform individual classifiers for phishing detection tasks [10].

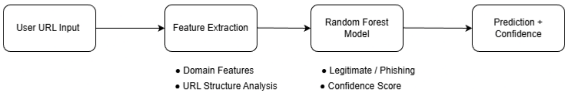


Fig. 1. System architecture of the proposed phishing detection system.

After identifying Random Forest as the optimal choice, the model was integrated into a lightweight web application capable of performing real-time URL classification. The deployed system processes user-input URLs, extracts the required features, and generates a prediction accompanied by a confidence score. This methodology ensures that the system remains fast, scalable, and effective for practical use in detecting phishing websites based primarily on URL characteristics.

A. Overview of the System

The proposed phishing detection system is designed to classify websites based solely on URL-derived characteristics, enabling fast and lightweight analysis without requiring full webpage content. The system operates through a sequential pipeline that begins with user input and concludes with a machine learning–based prediction delivered through a web interface. When a user submits a URL, the backend initiates a feature extraction process that interprets the structure, lexical patterns, and domain-related attributes of the URL. These extracted attributes form the input vector for the pre-trained Random Forest classifier, which evaluates the URL and predicts whether it is legitimate or phishing.

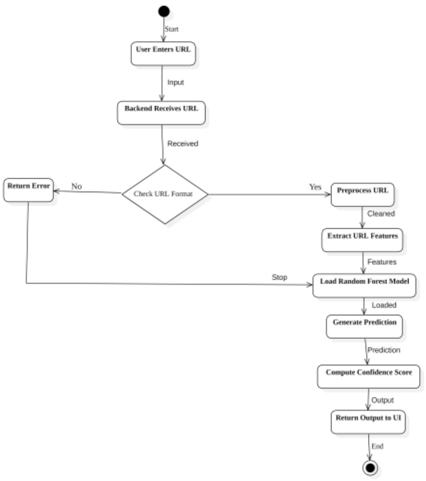


Fig. 2. Workflow diagram.

The overall design prioritizes efficiency and usability. Since the system relies only on URL features, it avoids heavy operations such as HTML parsing or content scraping, allowing for quick responses and compatibility with lightweight deployment environments. The methodology ensures that the system is capable of real-time detection while maintaining accuracy and stability. The final output presented to the user includes both the predicted class and an associated confidence score, offering a clear and interpretable assessment of the URL's legitimacy.

B. Dataset Description

The machine learning models developed in this study were trained and evaluated using the UCI Phishing Websites Dataset, a widely adopted benchmark for URL-based phishing detection research. The dataset contains a comprehensive set of labeled website records, each represented through a collection of handcrafted features designed to capture characteristics commonly associated with phishing activity. These features encompass lexical patterns, domain-related attributes, redirection behaviors, and abnormal URL structures, allowing the model to learn meaningful distinctions between legitimate and phishing URLs.

In its original form, the dataset is provided in ARFF format and includes 30 feature attributes along with a binary target label indicating whether a URL is phishing or legitimate. For practical integration into the system pipeline, the dataset was converted into CSV format to facilitate preprocessing, feature handling, and model training. The dataset's structured nature and clearly defined attributes make it well-suited for training traditional machine learning algorithms such as Logistic Regression, Decision Trees, XGBoost, and Random Forest.

The diversity of URL examples in the dataset enables the learning models to generalize effectively across common phishing indicators. This provides a reliable foundation for developing a lightweight detection system capable of processing user-submitted URLs and generating accurate predictions in real time.

C. Feature Extraction

In the proposed system, feature extraction serves as the core mechanism that converts raw URLs into structured numerical inputs suitable for machine learning classification. Since the model operates entirely on URL-based characteristics rather than full webpage content, the extraction process focuses on identifying lexical, domain-level, and behavioral patterns that historically distinguish phishing websites from legitimate ones. This approach enables lightweight, fast, and reliable detection while avoiding the overhead and reliability issues associated with HTML parsing, rendering, or dynamic content inspection.

The extraction process used in this work is based on the 30 handcrafted features defined in the widely adopted UCI Phishing Websites Dataset. These features capture multiple dimensions of URL behavior, including address structure, domain composition, protocol usage, redirection patterns, and abnormal URL behavior. For example, the system examines whether the URL uses an IP address instead of a domain name, whether it contains suspicious symbols such as "@", whether prefix–suffix anomalies appear in the domain, and whether the URL exhibits multiple redirections. Additional lexical indicators, such as URL length, number of subdomains, presence of shortening services, and HTTPS token misuse, help reveal obfuscation strategies commonly adopted by phishing sites.

Beyond these lexical patterns, domain-related metadata contributes significantly to the overall feature representation. The system analyzes aspects such as SSL certificate status, domain registration length, age of domain, DNS record validity, and indexed status in search engines. These features help identify malicious infrastructures that typically rely on newly registered, short-lived domains. Behavioral and structural indicators are also included, such as presence of pop-up windows, right-click disabling scripts, iframe usage, and abnormal form-handling behaviors, which frequently accompany phishing pages attempting to capture user credentials.

Each of the 30 feature attributes including request URL structure, anchor tag properties, external link composition, redirection behavior, statistical reports, and page ranking indicators is systematically derived from the input URL using predefined rules that mirror the UCI dataset definitions. This ensures that the feature vector extracted from real-time user input is fully aligned with the format used during offline model training, preserving consistency between training and deployment.

Once computed, these features are organized into a fixed-length vector and passed to the Random Forest classifier, enabling rapid inference in real time. Because these URL-based features require no external API calls or content retrieval, the overall extraction pipeline remains computationally efficient, robust to network constraints, and well suited for practical deployment in lightweight phishing detection environments.

D. *Machine Learning Models*

To identify the most effective algorithm for phishing URL detection, several supervised machine learning models were evaluated during experimentation. Each model was trained on the preprocessed dataset derived from the UCI Phishing Websites Dataset, ensuring uniform input representation and comparable evaluation conditions. The selected algorithms represent a diverse range of learning paradigms, allowing the system to assess both simple and complex decision-making behaviors.

The first set of models included Logistic Regression and Decision Tree classifiers. Logistic Regression served as a baseline linear model, offering interpretability and fast computation, while the Decision Tree provided a nonlinear, rule-based approach capable of capturing hierarchical decision patterns. Although both models performed reasonably well on structured URL features, their accuracy and robustness were limited compared to more advanced ensemble techniques.

To improve performance, ensemble-based algorithms such as XGBoost and Random Forest were also tested. XGBoost leveraged gradient boosting to model intricate feature interactions, while Random Forest combined multiple decision trees through bagging to reduce variance and improve generalization. Among all models evaluated, Random Forest consistently achieved the highest performance, demonstrating superior stability and reliability across multiple test cases. Its balanced accuracy, robustness to feature noise, and compatibility with lightweight deployment made it the optimal choice for integration into the final phishing detection system.

E. Training Procedure

The training procedure began with converting the UCI Phishing Websites Dataset from its original ARFF format into CSV to enable compatibility with standard Python-based machine learning tools. After conversion, the dataset was inspected for structural consistency, correct labeling, and overall distribution across the two classes. To ensure balanced representation and reduce sampling bias, the dataset was shuffled and split into training and testing sets using a stratified approach. This method preserved the ratio of phishing to legitimate samples across both sets, preventing skewed evaluation results.

Before training the models, all features were verified to ensure proper encoding and numerical representation. Since the dataset’s feature set consists of predefined integer-based values, no additional normalization was required. However, care was taken to maintain consistency between the offline training features and the features extracted during real-time prediction in the deployed application. This alignment ensured that the model trained on the dataset behaved accurately when processing user-submitted URLs through the web interface.

Each machine learning algorithm i.e., Logistic Regression, Decision Tree, XGBoost, and Random Forest was trained sequentially using the same training data. The models were evaluated on the test set using common metrics such as accuracy, stability across iterations, and generalization behavior. While simpler models such as Logistic Regression provided rapid training and interpretability, their performance tended to plateau on complex URL patterns. The Decision Tree model offered better nonlinearity handling but showed sensitivity to feature noise.

To obtain improved accuracy and robustness, ensemble-based models were explored. XGBoost was tested due to its ability to model complex relationships and handle feature interactions effectively. Random Forest, however, demonstrated superior reliability by averaging predictions across multiple decision trees, reducing variance, and avoiding overfitting more effectively than the other models. Its consistent accuracy across multiple evaluations made it the strongest candidate for deployment.

After selecting Random Forest as the final model, it was trained on the full training set and serialized using Python's model persistence mechanisms. The trained model was then integrated into the backend of the web application. During runtime, the application extracts selected URL features, constructs a feature vector identical to the training format, and passes it to the model for prediction. The output includes both the predicted class and a computed confidence score, ensuring that the system operates in real time with stable accuracy.

F. System Deployment

The final stage of the methodology involved deploying the trained Random Forest model into a functional web-based phishing detection system. The deployment framework was designed to ensure that the model could operate efficiently in real time, handling user-submitted URLs with minimal latency. To achieve this, the backend was implemented using Python, where the serialized Random Forest model is loaded during server initialization. This approach eliminates the need for reloading or retraining the model during runtime, significantly improving performance and responsiveness. When a user enters a URL into the web interface, the backend first performs the predefined feature extraction process to generate the corresponding feature vector. This vector mirrors the feature structure used during model training, ensuring consistency between offline learning and real-time inference. Once the feature vector is prepared, it is passed to the Random Forest classifier, which produces a binary prediction indicating whether the URL is legitimate or phishing. In addition to the prediction, the model outputs probability values through its predict_proba() function, which are then processed to generate a user-friendly confidence score.

The confidence score follows a scaled interpretation mechanism to ensure clarity and meaningfulness for end users. Based on the probability associated with the predicted class, the backend applies a transformation formula that adjusts the raw probability into an interpretable score while maintaining an upper and lower bound. This ensures that users receive intuitive feedback regarding the likelihood of a URL being phishing or legitimate. The final output, consisting of the prediction label and the adjusted confidence score, is displayed on the web interface. This deployment strategy enables practical, real-time phishing detection while maintaining accuracy, stability, and ease of use.

IV. IMPLEMENTATION AND EXPERIMENTAL SETUP

The implementation of the proposed phishing detection system is divided into an offline model development phase and an online deployment phase. In the offline phase, the UCI Phishing Websites Dataset is used to train and evaluate several supervised learning algorithms, while in the online phase, the best-performing model is embedded into a web application that provides real-time URL analysis. This separation allows computationally intensive tasks such as model fitting and evaluation to be carried out once, whereas the deployed system focuses on fast inference and user interaction, similar to other practical phishing detection frameworks that distinguish between training and operational environments [11].

For all experiments, the original ARFF dataset was converted into CSV format and loaded into a Python-based environment using standard machine learning libraries. The data was randomly shuffled and split into training and testing sets using a stratified split to preserve the proportion of phishing and legitimate samples across both subsets. This reduces sampling bias and provides a more realistic estimate of generalization performance. Since the features are already encoded as integer values representing categorical or binary conditions, no additional normalization was required. Consistent with previous work on malicious URL detection, the experiments focus exclusively on URL-derived features rather than full page content, which reduces computational cost and makes the system more suitable for lightweight deployment scenarios [12].

Four machine learning models Logistic Regression, Decision Tree, XGBoost, and Random Forest were trained using the same feature representation and evaluation protocol. Hyperparameters were kept at default or lightly tuned settings to reflect realistic deployment conditions rather than exhaustive optimization. Model performance was assessed primarily using accuracy on the held-out test set, along with qualitative observations about stability and robustness across different runs. These comparative results are used to identify the most reliable classifier for real-time deployment, with Random Forest ultimately selected as the final model due to its consistently superior performance.

In the deployment phase, the trained Random Forest model is serialized and loaded by the backend of the web application at server startup. When a user submits a URL, the backend reconstructs the corresponding feature vector using the same extraction logic applied during training and forwards it to the loaded model for inference. The model outputs a predicted class label (legitimate or phishing) along with class probabilities, which are then transformed into an interpretable confidence score before being displayed on the web interface. This implementation setup ensures that the behavior of the deployed system remains aligned with the conditions under which the model was trained and evaluated, enabling accurate and efficient phishing detection in real time.

V. RESULTS AND DISCUSSION

The performance of the proposed phishing detection system was evaluated by comparing multiple supervised learning algorithms trained on the UCI Phishing Websites Dataset. Each model Logistic Regression, Decision Tree, XGBoost, and Random Forest was trained using the same feature representation and tested under identical experimental conditions described in Section IV. To ensure fairness across models, all experiments used the same train–test split strategy, the same feature selection pipeline, and the same evaluation environment. Classification accuracy was chosen as the primary performance indicator, as it remains one of the most widely adopted metrics for benchmarking phishing detection effectiveness in prior research [13]. In addition to providing a straightforward measure of prediction correctness, accuracy also offers an intuitive comparison between linear, tree-based, and ensemble learning strategies. The comparative evaluation therefore reveals how effectively each algorithm captures the underlying patterns in URL-based phishing behavior and highlights the relative strengths of ensemble approaches compared to traditional classifiers.

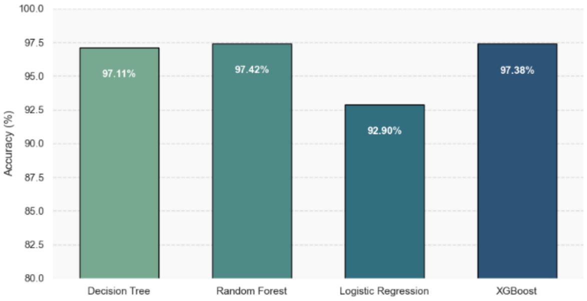


Fig. 3. Accuracy comparison of machine learning models for phishing website detection.

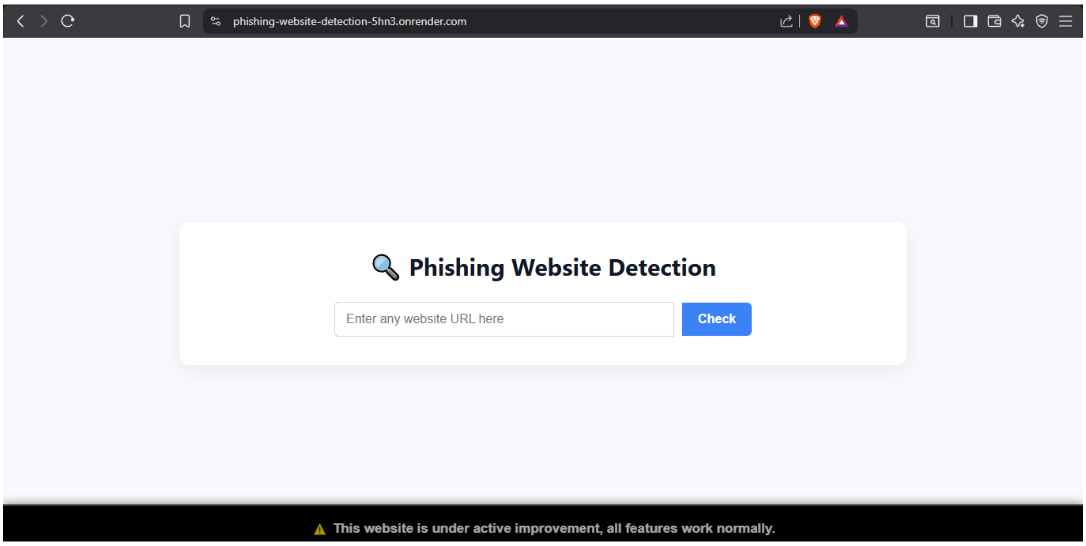
The results show that the Random Forest classifier achieves the highest accuracy among all evaluated models. Logistic Regression performs adequately on linearly separable patterns but struggles with the non-linear relationships present in phishing URL features. The Decision Tree model offers better interpretability but exhibits sensitivity to noise and tends to overfit. XGBoost demonstrates competitive accuracy due to its boosting mechanism, yet in the experiments conducted, it did not surpass the stability and overall performance of the Random Forest model. The accuracy comparison for all models is illustrated in Fig. 3, clearly indicating the superior performance of the Random Forest classifier. These observations align with previous findings in the literature, where ensemble-based models frequently outperform single learners in phishing detection tasks due to their ability to reduce variance and improve generalization [13].

Table I. Performance of Machine Learning Models on the Phishing Website Dataset

Model	Accuracy (%)
Logistic Regression	92.90
Decision Tree	97.11
XGBoost	97.38
Random Forest	97.42

The detailed accuracy values for each classifier are summarized in Table I. These results further confirm the advantage of ensemble-based approaches over individual learners for URL-based phishing detection. To further validate the system's reliability, several real-world and synthetic URLs were tested through the web interface after model deployment.

Beyond quantitative measures, the qualitative behavior of the system during deployment further reinforces its practicality. When tested with a broad mixture of benign, suspicious, and deliberately obfuscated URLs, the Random Forest model maintained stable responses and showed strong resilience to adversarial variations such as excessive subdomains, encoded characters, or misleading domain structures. The system also demonstrated low latency during real-time prediction, with most classifications completing almost instantaneously, which is essential for user-facing cybersecurity applications. These observations suggest that the combination of URL-based features and a robust ensemble model provides a dependable foundation for phishing detection, even under varied and unpredictable real-world conditions. In addition, the system showed consistent generalization across multiple test runs, indicating that its performance is not overly sensitive to minor fluctuations in feature distribution. The model’s robustness against noise and malformed URL structures further highlights its suitability for deployment in dynamic and evolving threat environments.



phishing-website-detection-5hn3.onrender.com

Phishing Website Detection

This website is under active improvement, all features work normally.

Fig. 4. Web interface of the deployed phishing detection system.

For each input URL, the deployed system returns a binary prediction and a confidence score derived from the model’s probability output. Empirically, the system demonstrated consistent behavior and maintained stable confidence values for correctly classified phishing URLs. While the confidence score is not used as a formal evaluation metric, it improves interpretability for end users by indicating how strongly the model favors its decision. This aligns with practical design considerations in security-oriented applications, where user-facing feedback can enhance trust and usability.

Although accuracy provides a clear comparison of overall performance, previous research highlights the importance of additional metrics such as precision, recall, F1-score, and false-positive rate particularly for phishing detection, where misclassifications may have significant consequences [14].

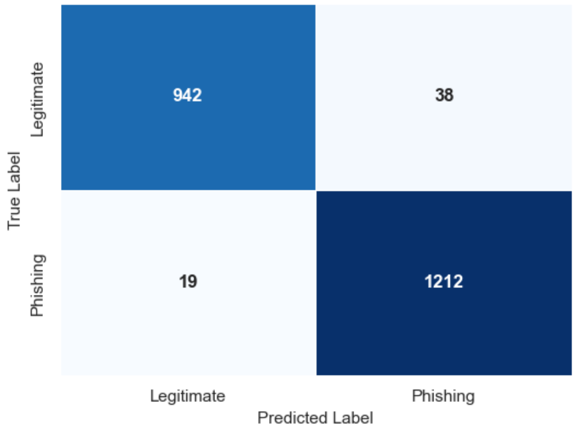


Fig. 5. Confusion matrix of the Random Forest classifier on the phishing URL test set.

In this study, the focus remains on accuracy due to deployment constraints and the nature of the feature set. However, extending the evaluation to include these metrics could offer deeper insights into model reliability, especially when dealing with imbalanced datasets or high-risk prediction scenarios.

Overall, the results demonstrate that URL-based machine learning offers an effective and lightweight solution for phishing detection. The Random Forest model provides the best balance between accuracy, stability, and computational efficiency, making it well suited for real-time deployment. The combined findings from offline evaluations and online testing confirm that the proposed system can reliably distinguish phishing URLs from legitimate ones while maintaining practical usability and response time.

VI. CONCLUSION AND FUTURE WORK

A. Conclusion

The proposed phishing detection system demonstrates that URL-based machine learning techniques can provide an effective, lightweight, and practical defense mechanism against phishing attacks. By training and evaluating multiple supervised learning models on the UCI Phishing Websites Dataset, the study identifies the Random Forest classifier as the most reliable and accurate algorithm for this task. The system integrates streamlined feature extraction with an efficient backend implementation capable of delivering real-time predictions, making it suitable for deployment in environments where rapid decision-making is essential. The web interface enhances usability by enabling users to easily submit URLs and receive immediate predictions along with a confidence score.

B. Future Work

Although the system demonstrates strong performance, several enhancements can be pursued. Future work may incorporate additional evaluation metrics such as precision, recall, F1-score, and false-positive rate to obtain a more comprehensive performance analysis. Integrating content-based features through web scraping or HTML inspection could strengthen detection capability against more sophisticated phishing strategies. Other directions include applying advanced ensemble learning, hyperparameter optimization, or neural architectures to further improve accuracy. Deploying the model as an API service or integrating it into browser extensions or security gateways could broaden real-world applicability. Overall, these enhancements can extend the effectiveness, scalability, and usability of the proposed phishing detection framework.

VII. REFERENCES

[1] Jain and B. Gupta, “A machine learning approach to detect phishing websites using URL-based features,” Journal of Information Security and Applications, 2021.

[2] Aburrous et al., “Intelligent phishing detection system for e-banking using fuzzy data mining,” Expert Systems with Applications, vol. 37, no. 12, 2010.

[3] Alzahrani, “The rising sophistication of phishing: A review of recent trends and detection mechanisms,” IEEE Access, 2022.

[4] W. Ali, S. Saad, and I. Awan, “Phishing website detection based on supervised machine learning with wrapper features selection,” IJACSA, 2017.

[5] A. Safi et al., “A systematic literature review on phishing website detection techniques,” Journal of King Saud University – Computer and Information Sciences, 2023.

[6] S. Yerima and M. Alzaylaee, “High accuracy phishing detection based on convolutional neural networks,” arXiv preprint arXiv:2004.03960, 2020.

[7] A. Krishna et al., “Phishing detection using machine learning-based URL analysis: A survey,” IJERT, 2019.

[8] R. Purwanto et al., “PhishZip: A new compression-based algorithm for detecting phishing websites,” arXiv preprint arXiv:2007.11955, 2020.

[9] S. Marchal et al., “PhishStorm: Detecting phishing with streaming analytics,” IEEE Transactions on Network and Service Management, vol. 15, no. 2, 2018.

[10] T. Choudhary et al., “A machine learning approach for phishing attack detection,” Journal of Artificial Intelligence and Technology, 2023.

[11] S. Basit et al., “A comprehensive survey of phishing detection: Machine learning and deep learning,” IEEE Access, vol. 9, pp. 38229–38246, 2021.

[12] R. Vinayakumar et al., “Detecting malicious URLs using lexical analysis and machine learning,” Springer Journal of Cybersecurity, 2019.

[13] A. Prakash and S. Srivastava, “Comparative analysis of ML algorithms for phishing URL detection,” Procedia Computer Science, 2020.

[14] N. Mohammed et al., “Evaluation metrics for phishing URL classifiers,” Information Sciences, vol. 579, pp. 970–987, 2021.