Malicious URL Detection Using Multi-Layer Hybrid Machine Learning

Dheyab Salman Ibrahim, Hassan Hadi Salih and Israa Mishkal
Department of Computer Science, College of Science, University of Diyala, 32001 Baqubah, Iraq
dr.dheyab@oudiyala.edu.iq, Hassan.hadi@uodiyala.edu.iq, israa_adnan85@student.usm.my

Keywords: Malicious Urls, Lexical Features, URL Obfuscation, Machine Learning.

Abstract: The increasing risk of cyber threats has significantly affected web-based services, especially phishing and malicious URLs, posing serious security challenges for both users and organizations. Previous detection approaches, such as blacklist-based, have difficulty detecting new threats or obfuscated malicious URLs. On the other hand, machine learning techniques (ML) have suffered from highly False Positive (FP) rates. In this study, we propose a hybrid multi-layer architecture for malicious URL detection by integrating blacklist verification, feature selection of URLs, an XGBoost-based classification technique, and rule-based post-processing. This proposed model works based on sequential layers to filter malicious URLs utilizing a blacklist dataset and then extract lexical features from URLs. Selecting features processed using the XGBoost classifier to distinguish between benign and malicious URLs. Then the final stage is to refine the classification results based on a rule-based preprocessing technique to enhance malicious URLs detection systems' performance and reduce FP rates. The experimental results show enhanced performance by achieving higher detection accuracy, lower FP, and improved computational efficiency compared to previous single-layer models. Our contribution in this study is to enhance the cybersecurity systems by providing a scalable and adaptive solution for real-world malicious URL detection systems.

1 INTRODUCTION

The widespread of using internet, cloud services and online transactions has increased the risk of cybersecurity threats, especially phishing attacks and malicious URL. These threats are the most common vectors that utilized to compromise users' data and organizational systems. Malicious URL is designed especially to mimic legitimate websites and trick users into disclosing sensitive information, such as banking details, login credentials, and personal data. Previous research demonstrate that malicious URL attacks continue to include a complexity, URL shortening, employing obfuscation, and domain generation methods to avoid the previous detection models [1], [2]. Consequently, detecting systems for malicious URL threats become a critical challenge in modern cybersecurity. In the previous studies, the presented detection systems are widely adopted because to official time response and easily to use. Most of these presented systems rely on storing dataset of identified malicious URL list, such as blacklist based. However, these exhibited of inherently ineffective against unseen attacks [3]. Furthermore, many attackers can be easily bypassed blacklist mechanisms by new generated threats or slightly modifying exiting URL making these systems insufficient as standalone solutions [4]. Several researchers have addressed these limitations by exploring heuristic and rule-based approaches that analyze URL patterns and structural properties. These models can detect special types of these threats and exhibit to lack generalization capability and detect new attack patterns [5].

In recently, machine learning (ML) has been integrated as powerful tools for detecting many cyber threats such as malicious URL by leveraging data-driven methods to identify hidden patterns in URL features. Several ML methodologies have been utilized for classification tasks in this field. For instants of these methods are Random Foresters, Support Vector Machines (SVM), and Gradient Boosting models [6], [7]. XGBoost has improved a significant attention due to its highly performances, scalability, and ability to handle complex feature extractions [8]. However, ML has some limitations, such as its heavily performance depends on feature engineering and quality of the dataset, and many of

ML approaches may produce high FP rates when dealing with real-world environments [9].

To full these gaps in malicious URL detections systems, many research have focused on using hybrid and multi-layer detection approaches that can combine multiple models to enhance detection accuracy and robustness. Based on using blacklist filtering, feature extractions based on ML and rule-based decision refinement that help to improve detection systems by leveraging their strengths and prevent weaknesses. These techniques provide a more comprehensive detecting mechanism by enabling early on filtering of seen threats, accurate classification of unseen URL, and post-processing to reduce misclassification errors [10]. In this study, we propose a hybrid multi-layer architecture for detecting malicious URL attacks. This approach combines blacklist verification, URL feature extraction, XGBoost-based classification, and rule-based post-processing in sequential manner. Our goal is to improve the performance's detection, reduce False Position Rates (FPR), increase computational efficiency, produce a detection system that mimic real-world cybersecurity applications [11]. Our main contributions are:

1) Designing robust and adaptive a multi-layer hybrid architecture to detection unseen malicious URL threads.
2) Integrating XGBoot methods with rule-based refinement for enhancing detection performance.
3) Increasing detection accuracy and reduce False Position Rates (FPR).
4) Presenting a scalable model to mimic real-time world cyber threats.

The remaining paper is organized as following: Section 2 reviews related work. Section 3 describes the proposed methodology. Section 4 presents the results and dissociation, and Section 5 shows the conclusion and future works.

2 RELATED WORKS

Several studies have been presented malicious URL detection systems due to widespread phishing attacks and cyber threats in web-based services. These studies can be categorized into blacklist, heuristic and rule-based, ML, deep learning (DL), and hybrid frameworks techniques (seen in Table 1).

2.1 Blacklist-Based Technique

It is the earliest presented technique that widely used in detection malicious URL threats. This technique relies on maintaining dataset of seen threats and matching incoming URLs against these lists. Many existing studies achieve computational efficient and miming real-world filtering; however, there are some limitations that lie in their inability to detect zero-day attacks or unseen malicious URL. Ma et al. demonstrated that blacklist-based models exhibit to generalize to unseen attacks and significantly limit their effectiveness in dynamic environments [12]. Moreover, attackers can easily avoid detection systems by applying slight modifications to URL or generating new threats. Therefore, the blacklist approaches insufficient when utilized alone in detection systems.

2.2 Heuristic and Rule-Based Techniques

These limitations parallel heuristic and rule-based approaches, which rely on structures URL analysis and identification of suspicious patterns. This paper presented a framework that used handcrafted rules to recognize phishing URL-based structural characteristics [13]. This approach can be effective in detecting seen attacks patterns; however, it lacks adaptability and exhibits to generalization unseen threats. Moreover, maintaining and updating rule sets require time consuming and more experiences in the field.

2.3 Machine Learning (ML) Techniques

Utilizing ML techniques have been improved detection of malicious URL systems by learning patterns from data. These ML approaches typically involve extraction feature, including lexical features (special characters and URL length), host-based features (DNS records, domain age), and content-based features. The authors applied differences ML methodologies, such as SVM, Random Forest, and Naïve Bayes. They achieved a higher performance [14]. Also, [15] provided a comprehensive review that highlighting the effectiveness of using ML in large scale URL detection systems. XGBoot is one of the efficiency ML methods that have demonstrated superior performance because of its ability to handle high-dimensional data, reducing complexity feature interactions and overfitting [16].

Table 1: Present comparison of existing malicious URL detection methods.

Ref	Approach Type	Techniques Used	Advantages	Limitations
[12]	Blacklist-Based	URL database matching	Fast, low computational cost	Cannot detect zero-day attacks, easily bypassed
[13]	Rule-Based / Heuristic	URL structure analysis, handcrafted rules	Simple, interpretable	Poor generalization, requires manual updates
[14]	Machine Learning	SVM, RF, NB with feature extraction	Good accuracy, adaptable	Depends on feature quality, false positives
[15]	ML Survey	Large-scale ML models	Scalable, effective in classification	Limited real-time adaptability
[16]	Ensemble Learning	XGBoost (Gradient Boosting)	High accuracy, handles complex features	Needs tuning, computational overhead
[17]	Deep Learning	CNN, LSTM	Automatic feature extraction, high accuracy	High computational cost, large data required
[18]	Hybrid (Rule + ML)	Fuzzy logic + ML	Improved accuracy, robustness	Increased system complexity
[19]	Hybrid Optimization	PSO/GA + ML	Better feature selection, improved performance	Higher computation, slower training

2.4 Deep Learning DL Techniques

Recently, the most effective detection malicious URL systems have been leveraging DL methodologies, such as Recurrent Neural Networks (RNN), Convolutional Neural Networks (CNN), and long Short-Term Memory (LSTM). They have been utilized to automatically extract features from raw URL strings and eliminating the need for manual feature engineering. In this paper [17], the authors proposed a DL approach that achieved performance detection by learning complex representations of URL. However, DL has some limitations, such as require large dataset for training, high computational resource, a long training time. These limitations may limit in real-time or resource constrained environments.

Furthermore, enhancing detection performance needs to propose hybrid approaches to combine multiple techniques. Aburrous et al. presented an intelligent phishing detection model by integrating fuzzy logic with rule-based and using ML. it demonstrates the improvement of accuracy and robustness [18]. Many recent studies have explored hybrid frameworks that combine optimization algorithms, including genetic algorithm and particle swarm optimization (PSO) with leveraging ML methods to enhance selections and parameters by optimizing feature extraction [19]. These models have presented significant improvements in detection accuracy, recall, and precision comparing other the studies of art.

Moreover, other studies have focused on using multi-layer architectures by integrating various detection methods in a sequential manner [20]. These methods typically have employed an initial filtering stage, such as blacklist technique, and then followed by ML classification and further refinement using additional rules or optimization techniques. These approaches aim to balance detection speed and performance while reducing false positives (FP) and enhancing computational resources.

However, there are several challenges remain unresolved. Many exiting studies rely heavily of static datasets and feature engineering. These reasons limit these approaches adaptability to emerging threats. ML may suffer of FP rates in real-world detecting, and DL introduces significant computational costs. Moreover, other existing systems lack effective integration between intelligent classification approaches and filtering mechanisms within a unified architecture. As a result, there is a strong reason for developing a hybrid multi-layer framework by combing blacklist verification, ML such as XGBoot, and rule-based post-processing to enhance robustness and performance that can detect malicious URL.

3 METHODOLOGY

To address the limitations that mention in previous section above, we propose a multi-layer hybrid blacklist XGBoot and rule post-processing framework to improve malicious URL threads detection. This framework integrates four sequential layers namely: blacklist verification, URL feature extraction, XGBoost-based classification, and rule-based post-processing for enhancing performance, reducing FP, and ensure efficient real-time processing.

3.1 System Architecture Overview

The multi-layer a hybrid approach is structured as pipeline (Fig. 1). Each layer performs a specific role in detection process. The first layer is to process incoming URLs by filtering module, then extraction feature engineering, ML- classification, and final decision refinement. These layers design ensures that simple cases are handled early, while complex cases are analyzed using more sophisticated mechanism.

3.2 Blacklist Verification Layer

In this layer, each incoming URL is checked with predefined blacklist dataset to find seen malicious URLs. If it matches, the URL is immediately detected as a malicious attack. This layer significantly reduces computational overhead and increases responding time for previously identified threats. On the other hand, if URLs are not found in blacklist, this layer is forwarded them to the next layer for deeper analysis.

3.3 URL Feature Extraction Layer

The next layer is for extracting discriminative features from URL threats. The extraction features are separated into three mean categories namely: lexical, host-based, and structural features (as shown in the Table 2).

The extracted features from this layer transform into a numerical feature vector suitable for XGBoot-based classification.

3.4 XGBoost Classification Layer

In this stage, the processed extraction features from previous layer to feed an XGBoot-based classifier for binary class classification either benign or malicious. This layer is chosen based on high performance, scalability, and ability to capture non-linear feature interactions. XGBoot is trained using labeled datasets of benign and malicious URLs, optimizing a loss function based on classification error. The final result prediction of this layer is obtained using an ensemble of decision trees, where each tree contributes to enhancing the overall model accuracy. The prediction can be represented as (1).

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i), f_k \in \mathcal{F} . \quad (1)$$

Where:

- $\hat{y}_i$ is the predicted output;
- f_k represents the k-th decision tree;
- $\mathcal{F}$ is the space of regression trees.

Table 2: The features categories that extracted from the 2nd layer.

Lexical	Host-based	Structural
Special characters	DNS records	Presence of Https
URL length digit to letter ratio	Domain age IP address	Number of subdomains path depth
Presence of suspicious tokens	WHOIS information	URL entropy

3.5 Rule-Based Post-Processing Layer

In this layer, we utilized a set of rule-based conditions to refine the output of XGBoot layer. This layer is designed to reduce FP rates and enhance decision reliability. for instance, URLs flagged with borderline probabilities by XGBoost are evaluated using pre-defined security rules, such as suspicious keyword presence or domain reputation thresholds. This proposed model ensures to achieve higher robustness and interpretability.

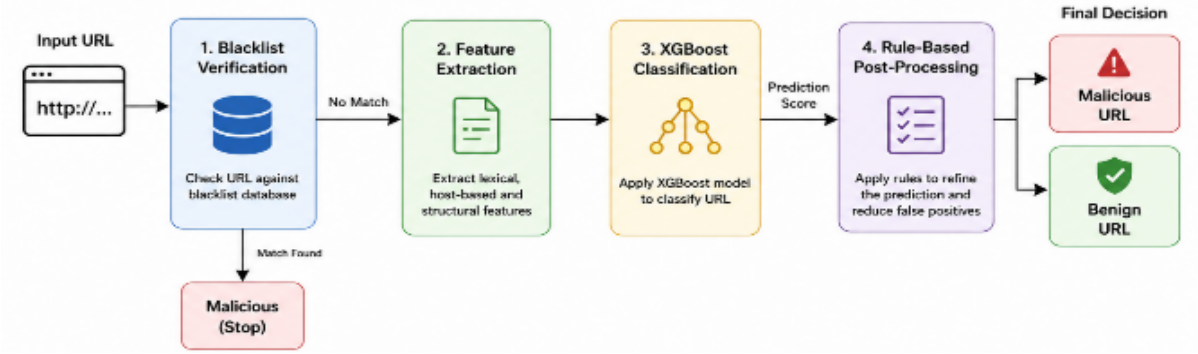

Figure 1: The overall architecture of the proposed multi-layer malicious URL detection system.

3.6 Decision Output

The final classification decision is obtained by integrating the outputs generated at each processing stage. If either the blacklist module or the XGBoost classifier identifies a URL as malicious with high confidence, the URL is immediately classified as malicious. Otherwise, it is classified as benign. This multi-stage decision strategy improves both detection accuracy and system reliability while maintaining computational efficiency. The overall workflow of the proposed framework is illustrated in Figure 1.

3.6.1 Decision Framework

The proposed framework follows a hierarchical decision strategy in which multiple detection mechanisms cooperate to classify an input URL. The classification process begins with a lightweight blacklist verification and proceeds through feature extraction, machine learning classification, and rule-based refinement before producing the final decision. This sequential architecture reduces unnecessary computations while improving classification accuracy and minimizing false-positive detections.

3.6.2 Classification Procedure

3.6.2.1 Input URL

The classification process begins by receiving an input URL. The URL is first compared with a predefined blacklist containing known malicious websites. If a match is found, the process terminates immediately and the URL is classified as malicious, thereby reducing computational cost. Otherwise, the URL proceeds to the next processing stage.

3.6.2.12 Feature Extraction

In the second stage, a comprehensive set of discriminative features is extracted from the input URL. These include lexical, host-based, and structural features, as described in Section 3.3. The extracted features are then transformed into numerical feature vectors for machine learning classification.

3.6.2.3 XGBoost Classification

The extracted feature vectors are classified using the XGBoost algorithm. The classifier produces a prediction score indicating the probability that the URL is malicious or benign. XGBoost was selected because of its ability to model nonlinear relationships and complex feature interactions while maintaining high computational efficiency.

3.6.2.4 Rule-Based Refinement

The prediction generated by the XGBoost classifier is further refined using predefined security rules. This stage reduces false-positive classifications by incorporating additional information, including domain reputation, suspicious keywords, and probability thresholds.

3.6.2.5 Final Decision

Finally, the outputs of all previous stages are combined to generate the final classification result. Each URL is labeled as either Malicious or Benign. By integrating blacklist filtering, machine learning classification, and rule-based refinement, the proposed framework achieves an effective balance between computational speed, detection accuracy, and suitability for real-time cybersecurity applications.

4 EXPERIMENTAL RESULTS AND DISCUSSION

4.1 Experimental Setup

We evaluated the proposed hybrid multi-layer malicious URL detection framework using a publicly available benchmark dataset containing both benign and malicious URLs. The dataset includes lexical, host-based, and structural features extracted from real-world web traffic. All experiments were implemented in Python using an XGBoost classifier. The dataset was randomly divided into training and testing subsets in an 80:20 ratio. Model performance was evaluated using standard metrics, including accuracy, precision, recall, F1-score, and false positive rate (FPR). All experiments were conducted on a standard CPU-based computing environment without GPU acceleration.

4.2 Results and Comparative Analysis

The experimental results obtained by the proposed hybrid framework are summarized in Table 3. The model achieved an accuracy of 98.7%, together with a precision of 98.3%, recall of 97.9%, F1-score of 98.1%, and a low false positive rate of 1.2%. These

results demonstrate that the proposed framework provides both high detection capability and reliable discrimination between benign and malicious URLs.

Table 3. Performance of the proposed hybrid multi-layer framework.

Metric	Value
Acc.	98.7%
Precision	98.3%
Recall	97.9%
F1-score	98.1%
FPR	1.2%

To further evaluate the effectiveness of the proposed approach, its performance was compared with several state-of-the-art machine learning and deep learning methods. The comparison includes Support Vector Machine (SVM), Random Forest, Long Short-Term Memory (LSTM), and the baseline XGBoost classifier. The results are presented in Table 4.

The proposed framework achieved the highest classification accuracy (98.7%) while simultaneously obtaining the lowest false positive rate (1.2%). Compared with the baseline XGBoost model, the hybrid framework improved accuracy by 1.3 percentage points and reduced the false positive rate from 1.8% to 1.2%. These improvements demonstrate that combining blacklist filtering, feature extraction, XGBoost classification, and rule-based refinement provides more robust and reliable malicious URL detection than using a single classification model.

Table 4: Comparison of the proposed framework with state-of-the-art methods.

Method	Accuracy	FPR	Remarks
SVM	92.1%	5.8%	Sensitive to feature quality
Random Forest	94.5%	4.3%	Moderate performance
Deep Learning (LSTM)	96.8%	2.5%	High computation cost
XGBoost (baseline)	97.4%	1.8%	Strong performance
Proposed Model	98.7%	1.2%	Best overall performance

5 DISCUSSION

In the experiment finding results, the proposed hybrid multi-layer model clearly outperforms the traditional and standalone ML-based approaches. The improvement in performance proposed approach mainly due to:

1) Blacklist layer efficiency that stored all seen malicious URLs that are filtered instantly, reducing system load and improving response time.
2) Feature representation methods that combine all types of extraction features to enhance the discriminative power of the proposed model.
3) XGBoost strength effectively utilizes to capture non-linear relationships between URL features.
4) Rule-Based Refinement uses to significantly reduces FP rates by correcting borderline predictions from the ML model.

The proposed model achieved enhancement in the reduction of the False Positive Rate (FPR=1.2%), which is considered critical in real-world cybersecurity systems where insecure URLs are blocked that can be affecting users' experience and system reliability. Unlikely the existing single model, the proposed archenteric is more effective a malicious URLs detection system due to each layer contributes differently, including Fast filtering blacklist, intelligent classification XGBoost, and decision refinement rules. This provides a balanced trade-off between robustness, performance, and generalizations.

6 CONCLUSIONS

This study proposes a hybrid multi-layer framework for enhancement the malicious URL detection system. It designs to address the limitations that existed in the previous detection systems including using blacklist-based single ML methods. The proposed approach utilizes multi-layer by integrating the blacklist verification, features extractions, ML-based classification, and rule-based post-processing. This approach improves detection performance while reducing False Position Rates (FPR), especially in identifying unseen generated and obfuscated malicious URLs. This enhancement achieved by combining complementary detection strategies, where each layer contributes to progressively filtering and classifying URLs with higher precision. The experiment analyses demonstrate that the proposed approach improves the detection accuracy and reduces FPR. However, this model still faces some challenges, such as handling highly dynamic adversarial URL generation techniques and maintaining real-time performance under big data

web traffic. In future work, we will focus on integrating DL methods to expand features sets with semantic and behavioral analysis. Additionally, we try to optimize computational efficiency for deployment in real-time cybersecurity environments.

REFERENCES

[1] Y. Khonji, A. Iraqi, and A. Jones, "Phishing detection: A literature survey," IEEE Commun. Surveys Tuts., vol. 15, no. 4, pp. 2091-2121, 2013.

[2] T. Moore and R. Clayton, "Examining the impact of website take-down on phishing," in Proc. eCrime Researchers Summit, 2007.

[3] H. R. Hassan, D. S. Ibrahim, and Z. T. Mustafa Al-Ta'i, "Advanced Human Activity Recognition Using Pose Estimation and Deep Learning Techniques," in 2024 Antennas Design and Measurement International Conference (ADMInC), Saint Petersburg, Russian Federation, 2024, pp. 94-100, [Online]. Available: https://doi.org/10.1109/ADMInC63617.2024.10775304.

[4] R. S. Rao and A. T. Ali, "Evasion techniques in phishing URL generation," J. Cyber Secur. Technol., 2018.

[5] M. Gupta, A. Kumar, and R. Rastogi, "Detecting phishing websites using heuristic rules," Int. J. Comput. Appl., 2016.

[6] D. S. Ibrahim, F. K. Zaidan, J. Kadum, H. H. Saleh, L. T. Rasheed, and W. S. Nsaif, "Routing Protocols-Based Clustering in WSNs," in 2021 4th International Iraqi Conference on Engineering Technology and Their Applications (IICETA), Najaf, Iraq, 2021, pp. 201-205, [Online]. Available: https://doi.org/10.1109/IICETA51758.2021.9717419.

[7] M. S. K. Sahoo, G. Sahoo, and S. Mohanty, "Malicious URL detection using machine learning: A survey," Int. J. Appl. Eng. Res., 2017.

[8] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min. (KDD), 2016.

[9] N. Jain and V. Gupta, "Performance analysis of ML classifiers for phishing detection," Procedia Comput. Sci., 2019.

[10] S. Abu-Nimeh, D. Nappa, X. Wang, and S. Nair, "Evaluation of classification techniques for phishing detection," Inf. Secur. J., 2007.

[11] P. K. Roy and S. Chandra, "Hybrid intrusion detection systems for cyber threats," IEEE Access, 2020.

[12] D. S. Ibrahim, S. T. Hasson, and P. A. Johnson, "Selecting an Optimal Cluster Head using PSO Algorithm in WSNs," in 2022 International Conference on Software, Telecommunications and Computer Networks (SoftCOM), Split, Croatia, 2022, pp. 1-4, [Online]. Available: https://doi.org/10.23919/SoftCOM55329.2022.9911416.

[13] A. M. Asiri and N. Marriwala, "A Literature Survey on LEACH Protocol and Its Descendants for Homogeneous and Heterogeneous Wireless Sensor Networks," in Algorithms for Intelligent Systems, Springer, 2021, pp. 281-295.

[14] A. Sahingoz, B. Buber, O. Demir, and B. Diri, "Machine learning based phishing detection from URLs," Expert Syst. Appl., 2019.

[15] K. Sahoo and B. Gupta, "A survey on malicious URL detection techniques," J. Netw. Comput. Appl., 2017.

[16] T. Chen and C. Guestrin, "XGBoost: Extreme gradient boosting," in Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min. (KDD), 2016.

[17] H. Aljofey, Q. Jiang, H. Qu, M. Huang, and J. Niyigena, "Deep learning-based phishing URL detection," IEEE Access, 2020.

[18] K. Aburrous, M. A. Hossain, K. Dahal, and F. Thabtah, "Intelligent phishing detection system using fuzzy rules," Expert Syst. Appl., 2010.

[19] M. Mafarja and S. Mirjalili, "Feature selection using hybrid PSO and GA for classification problems," Appl. Soft Comput., 2018.

[20] M. Rasulmukhamedov, A. Tukhtakhodjaev, and O. Turdiev, "Application of Machine Learning Algorithms for Optimizing Document Workflow Management in Railway Freight Transportation," in Proc. Int. Conf. Applied Innovation in IT, vol. 13, no. 2, pp. 51-57, 2025.