Machine Learning-Based Malicious URL Detection System

Laxminath Tripathy[1], Yugal Kumar[2], Dibya Ranjan DasAdhikary[3]

{laxminathtripathy@soa.ac.in[1], Yugal.kumar@nmims.edu[2], dibyranjandasadhikary@soa.ac.in[3]}

ITER, Siksha 'O' Anusandhan (Deemed to be) University, Bhubaneswar, Odisha, India[1,3]

Department of Computer Engineering, STME, NMIMS University, Chandigarh[2]

Abstract. Cyber threats have been heightened due to the need to have effective ways of detecting and thwarting malicious links that are often employed to perform phishing, malware downloading, and unauthorized data access. This paper describes a machine learning-based approach to detecting malicious URLs, utilizing a substantial amount of lexical, structural, and semantic data. It was verified using the random forest, extra trees, and XGBoost classifiers on a balanced dataset, which had been trained using the system. The experiments displayed that our model is efficient, accurate, and has low error rates. It classified URLs into five categories, namely benign, defacement, legitimate, malware, and phishing. It offers a scalable and effective method for identifying dangerous links at the initial stage and raising cybersecurity.

Keywords: Malicious URL Detection, Phishing Links, Machine Learning, Cybersecurity, URL Classification

1 Introduction

As a result, internet users are growing at a very high rate, which is leading to a high rate of cyberattacks. The main intention of a malicious link is to damage systems by sending fake websites, enabling them to download malware, receiving logins, or maliciously gaining access to systems without permission. The users can now easily click the malicious links without their notice since cybercriminals can present them as legitimate links [12]. Conventional strategies of counteracting cyberattacks, like blacklists or manual inspections, are not quite as effective as other threats are constantly evolving. To overcome these challenges, our Malicious Link Detection project suggests a multifaceted system that has succeeded in the detection of malicious URLs using machine learning methods. The system splits the URLs, which are based upon structural and statistical characteristics of the URLs, deriving a set of applicable characteristics, and trains models to classify the safe and malicious. Our research is influenced by the increasing rates and severity of cyberattacks that involve the use of malicious links [3–6].

The objective of this research is to develop a classifier capable of detecting and labelling URLs as legitimate or malicious. To automate this process, we employ machine learning algorithms to build a lightweight, extensible system that can be regularly updated and improved [7–9]. Relying solely on user awareness is insufficient, as most attacks succeed despite warnings. A

IICTDS 2025, December 19-20, Chandigarh, India
Copyright © 2026 EAI
DOI 10.4108/eai.19-12-2025.2362398

more effective approach is to implement automated systems that analyse and evaluate links before users interact with them [10–11].

The remaining sections of this paper are organized as follows: the existing literature is reviewed in Section 2. The detailed proposed methodology is discussed in Section 3. Section 4 discusses implementation. The experiment details are discussed in Section 5. Section 6 presents results and evaluation. Section 7 concludes the paper and outlines future directions.

2 Literature Survey

Over the recent past, a number of researchers have come up with different algorithms of detecting bad links or URLs. Some are speed-oriented, others accuracy-oriented, and most of them are oriented to balance the two [12 -14]. It is one of the disciplines that have been evolving continuously, with scientists perfecting the features of models, testing them, and evaluating their performance on different levels. In the article by Emine Uczar et al. (2022), a convolutional Neural Network (CNN) model was employed on the ISCX-URL-2016 dataset. It contains approximately 114400 URLs in five categories. CNN performed better than LSTM because it is more effective in capturing URL patterns [10]. Jun Chen Chong et al. (2025) employed KNN, RF, DT, and employed genetic algorithm to select features with genetic algorithm using 10-fold cross-validation, which avoids irrelevant or redundant features [13]. Nuria Reyes-Dorta et al. [14] proved the effectiveness of classical ML models with the accuracy of over 93%. Another important fact of these studies is that models have to be frequently updated in order to be efficient in terms of the constantly changing environment of internet threats. On the whole, Lexical features themselves can be rather potent, and ensemble strategies provide one with the means to enhance performance with minimum overhead. A similar type of research was carried out by Ranjit Patgiri et al. (2019) who used SVM, Naive Bayes, KNN, and Decision Trees [3]. Apoorva Joshi et al. (2019) suggested a slightly different method, which minimizes the latency [5]. Their results were impressive: only 0.1% false negatives and an AUC of 0.98, which is pretty exceptional for a lightweight system. Another stacked LSTM and XGBoost model that was proposed by Saba Aslam et al. (2024) is a two-phase stacked model [12]. It is noteworthy that the method is independent of phishing characteristics, but it has a high accuracy on a benchmark dataset. Y. Tian et al. introduce a detailed overview of malicious URL detection methods that encompass lexical, content-based, network/DNS-based and deep learning methods [15]. Their work focuses on the recent shift to Transformer and GNN-based models and the necessity to have unified benchmark datasets. F. Turk (2025) makes comparisons of traditional machine-learning, deep-learning, and optimization-driven hybrid architecture of malicious URL classification [16].. The study demonstrates that feature-selection optimizers such as GA/PSO, improve both detection rate and computational efficiency. The results justify the relevance of feature-engineering and hybrid models in modern URL detection systems. Q. E. Haq et al. (2024) propose a deep 1-D CNN architecture to detect phishing URLs using end-to-end character-level inputs [17]. The work shows improved AUC over classical machine-learning baselines and discusses techniques for dataset imbalance. It provides a theoretical comparison point for deep-learning-based URL detectors. This paper is useful to contextualize the research gap addressed by the present study. Table 1 summarizes the key findings from recent studies, listing each Author, the Model(s) used, the Main Findings, and the achieved Accuracy. It provides an overview of prior research results in one place.

Table 1. The Summary of Literature

AUTHOR	MODEL NAME	KEY FINDINGS	ACCURACY
Emine Uçar et al. (2022) [10]	Convolutional Neural Network (CNN)	It used the ISCX-URL-2016 dataset with about 114,400 URLs in five categories. CNN worked better than LSTM by capturing URL patterns more effectively.	95.37%
Apoorva Joshi et al. (2019)[5]	Ensemble classification using bagging algorithms	Relies solely on static lexical features extracted from URL strings, avoids the need for crawling URLs or blacklist lookups, reducing latency. Achieved a low false negative rate of 0.1%.	92.00%
Saba Aslam et al.(2024)[12]	LSTM,XGBoost	Two-phase stacked model.No phishing-specific features used.Robust accuracy on benchmark dataset.	96.04
Jun Chen Chong et al. (2025)[13]	KNN, RF, DT	Applied Genetic Algorithm for feature selection.Used 10-fold Cross Validation.Removes irrelevant or redundant features	90.91%
Nuria Reyes-Dorta et al. (2024)[14]	Logistic Regressio, Decision Tree, SVM, Neural Networks	Achieved true positive rates exceeding 90% using traditional ML models. Approach for malicious URL detection	93.18%

3 Methodology

This section outlines the methodology followed in designing and implementing the Malicious Link Detection. The approach integrates feature engineering, machine learning model development, and real-time web-based prediction. The detail steps are shown in Figure 1.

3.1 System Architecture

The proposed architecture of the Malicious Link Detection comprises the following stages:

a) Data Collection

A publicly available and balanced dataset of labelled URLs (malicious and benign) was sourced from platforms such as Kaggle and the UCI Machine Learning Repository. The dataset includes URLs labelled into five categories: benign, defacement, legitimate, malware, and phishing.

b) Preprocessing and Feature Extraction

The URLs were subjected to extraction of more than 30 lexical, structural and semantic features to analyze it. These constituted three groups namely Basic Lexical Features: URL length, hostname length, number of digits, number of special characters, and presence of an IP address or

HTTPS. The intermediate attributes were the use of URL shortening services, query string length and presence of brand names, and the more sophisticated semantic attributes, which included path entropy, number of subdomains, repetition patterns and also sensitive keywords, such as, login, or bank. The preprocessing stage was used to guarantee quality and consistency of the data such as elimination of duplicate entries, the treatment of missing or corrupted data and scaling of numerical values.

c) Model Training and Validation

The extracted features were put into three classification models, namely Randome Forest, ExtraTrees and XGBoost, which were trained on the extracted features to identify malicious URLs. The training procedure consisted of splitting the data into training and testing which is usually based on an 80-20% ratio. The 10-fold cross-validation was used to conduct training to guarantee robustness of the model and to reduce overfitting. Also, each algorithm was optimized using the hyperparameter tuning to enhance predictive accuracy and performance.

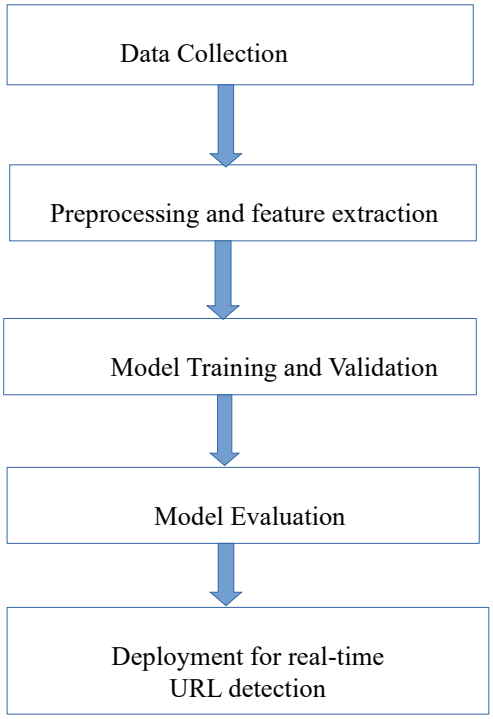


Fig. 1. Proposed Architecture of Malicious Link Detection

d) Model Evaluation

Models are evaluated using different metrics, including accuracy, precision, recall, and F1-score. Each classifier was tested on its ability to predict the correct URL class. The Receiver

Operating Characteristic (ROC) curve was also generated to visualize performance across different feature categories.

e) Real-Time Detection System

The last step is the implementation of the most effective model in a web-based application. The application takes the URLs entered by the user and classifies them real-time as either one of the five categories. It supports the following architecture: Lightweight frontend that receives user input, Backend model inference with Flask or Django, and visualizing the probability of prediction and the outcome of a classification.

3.2 Implementation

The dataset was processed to include a formalized set of features which were split into three categories based on the level of information value and complexity. The first group, Basic Lexical Features, consists of such attributes as the number of digits, the usage of special characters, the use of HTTPS, and recognition of IP addresses in the URLs since they may suggest suspicious structures. Intermediate Features is the second category, which describes features such as shortening of URL, query length, and using brand names, which are usually used in phishing or spoofing techniques.

This paper has been broken down into three major parts; (1) prediction probabilities, (2) accuracy evaluation and (3) classification reporting. The models used Random Forest, ExtraTrees, and XGBoost in the training stage. It has proven effective in the detection and identification of phishing and malware URLs. The prediction step includes five different classes of URLs, namely benign, defacement, legitimate, malware, and phishing. This stage focuses on better accuracy of prediction.

4 Experimentation Detail

In this section, an outline of the experimental configuration used to develop and assess the Malicious Link Detection is provided. The details of feature representation and performance evaluation criteria used with various machine learning models are also discussed elaborately.

4.1 Dataset Description

The dataset used in this study is a balanced, publicly available collection of URLs from the UCI Machine Learning Repository and Kaggle. It includes over 200,000 labelled URL samples divided into five different categories: phishing, malware, legitimate, benign, and defacement. Every entry has been annotated with the appropriate class label and enhanced with more than 30 extracted features that reflect the URLs' structural and statistical properties. Robust training and equitable assessment of the classification models are guaranteed by this balanced distribution.

4.2 Feature Categories

The feature extraction process was organized into three layers to capture a comprehensive representation of URL characteristics. The Basic Lexical Features included attributes such as the

number of digits, special characters, presence of HTTPS or IP addresses, subdomain count, suspicious words like “login” or “verify,” and overall URL length, along with digit ratios. The Intermediate Features captured patterns often associated with phishing attempts, such as the use of URL shorteners, query string length, and the presence of brand names in the domain or path. Finally, the Advanced Semantic Features provided deeper structural insights, including path entropy, domain repetitions, presence of sensitive keywords within the path, longest repeated character sequences, inclusion of non-ASCII characters, and overall URL depth. This layered approach ensures both surface-level and semantic indicators of malicious behaviour are effectively represented for model training.

4.3 Experimental Setup

The machine learning classifiers trained and evaluated for malicious URL detection included Random Forest (RF), ExtraTrees Classifier, and XGBoost. The training and evaluation process followed a systematic configuration to ensure accuracy and robustness. The dataset was divided into training and testing sets using an 80-20 split, and 10-fold cross-validation was applied to minimize overfitting and validate model performance across multiple folds. Additionally, GridSearchCV is used for hyperparameter tuning to identify the optimal parameter settings for each classifier, thereby enhancing predictive performance.

4.4 System Performance Evaluation

Performance of the models was evaluated using the following metrics:

The definitions are as follows:

$$\text{Accuracy} = (TP + TN) / (TP + FP + FN + TN) \quad (1)$$

$$\text{Precision} = TP / (TP + FP) \quad (2)$$

$$\text{Recall} = TP / (TP + FN) \quad (3)$$

$$\text{F1 Score} = 2 * (\text{Precision} * \text{Recall}) / (\text{Precision} + \text{Recall}) \quad (4)$$

Confusion Matrix Analysis

5 Results and Discussion

In this part, the trained models are evaluated and the analysis of the effectiveness of detecting dangerous URLs are provided. Random Forest, ExtraTrees, and XGBoost are tested and compared as three machine learning algorithms. These models are evaluated based on a large set of evaluation criteria, such as precision, recall, accuracy, and F1-score. These are useful for iden-

tifying URLs in five different categories: phishing, malware, legitimate, benign, and defacement. The accuracy of different models is listed in Table 2 and shown in a bar graph in Figure 2.

Table 2. Accuracy of different Models

Algorithm	Accuracy (%)
SVM	92.15
Random Forest	94.20
XGBoost	96.85
Naïve Bayes	89.60
CNN	95.10
LSTM	95.85
Proposed Model	98.25

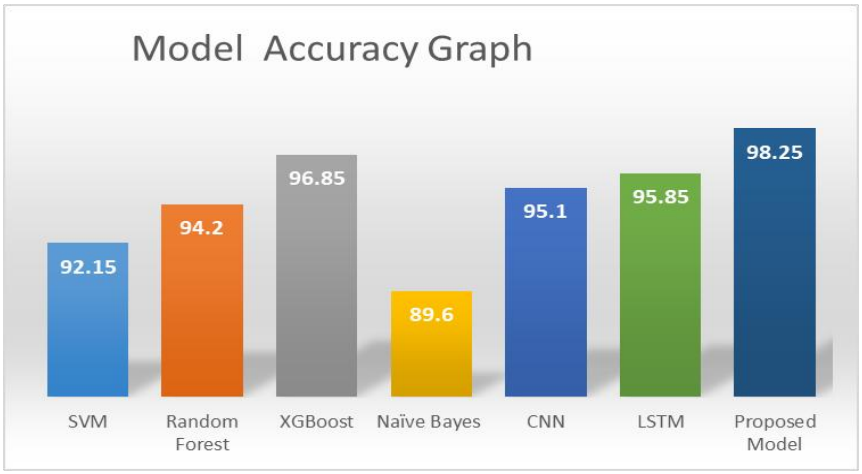


Fig.2. Graph of different Models' Accuracy

The results of the competitive performance of all models are presented with the Random Forest and XGBoost classifiers obtaining the best accuracy. These models performed well, particularly when it came to identifying phishing and malware URLs, which are more challenging to detect because of confusion and resemblance to genuine domains.

The confusion matrix is listed Table 3 which shows about how each model classified the URLs across all five classes. High values along the diagonal indicate strong class-wise performance, particularly for benign and legitimate categories. Due to structural similarities between phishing and defacement there were frequent misclassifications among them.

Table 3. Confusion Matrix

Actual \ Predicted	Benign	Defacement	Legitimate	Malware	Phishing
Benign	84337	10	1	1	1267
Defacement	48	18348	0	0	666

Actual \ Predicted	Benign	Defacement	Legitimate	Malware	Phishing
Legitimate	0	0	69056	0	91
Malware	44	36	12	4120	517
Phishing	2793	627	313	113	35646

The Receiver Operating Characteristic (ROC) Curve is depicted in Figure 3. It compares classification performance across different sets of URL features—lexical, structural, and combined. The ROC curve depicts the trade-off between true positive rate and false positive rate for each model.

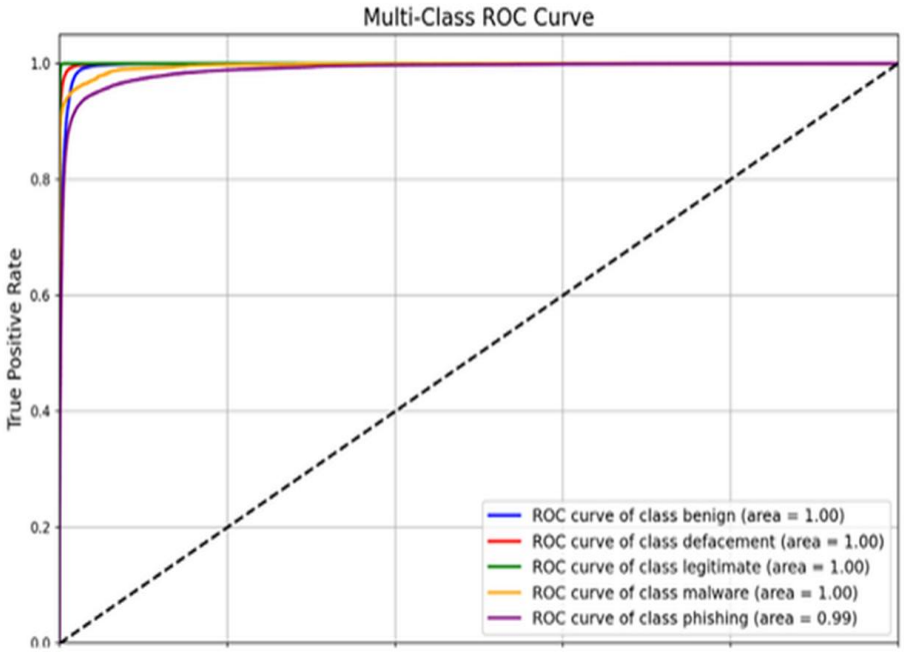


Fig. 3. ROC Curve

The performance of different feature sets employed in the classifiers is demonstrated in ROC curve. There has been some slight improvement in the performance of classification when structural features are taken in combination with lexical features. The best models have an area under the curve (AUC) greater than 0.95, so the model is highly reliable. Figure 4 is a ranked representation of the most important features in the detection of malicious URLs. Features such as URL length, presence of brand names, and path entropy emerged as top contributors in the decision-making process of the ensemble models.

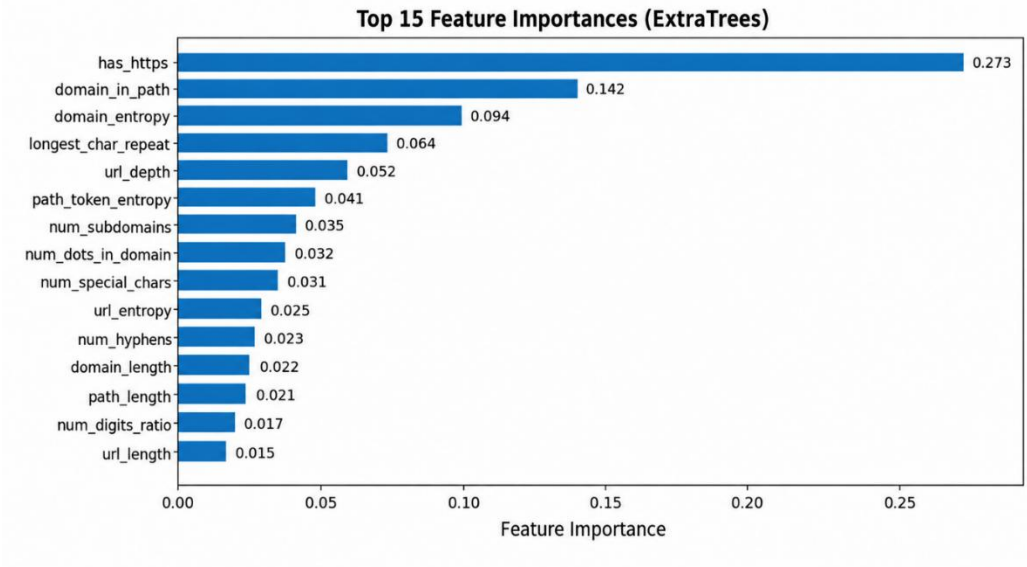


Fig. 4. Illustration of the most used feature

The evaluation demonstrates that ensemble learning techniques are effective and robust for real-time classification of malicious URLs. Their low false-negative rates and scalability make them well-suited for practical deployment in web-based threat detection systems. The execution time of different models have been listed in Table 4.

Table 4. Execution time of different Models

Model	Execution Time (in seconds)
SVM	1.42
Random Forest	2.65
XGBoost	3.51
CNN	9.80
LSTM	11.22
Proposed Model	2.10

6 Conclusion and Future Scope

Our work on malicious link detection aims to help society amid the rising concerns regarding data. Clicking or sharing links without any knowledge of whether the link is safe or not. This mainly focuses on identifying links that are not safe to use. A notable strength of our model is its high accuracy and the correspondingly low margin of error observed during evaluation. With constant updates and maintenance, it could be a great help to society. Future development includes integrating it with global data sets. Making web extensions to ease the detection. Adding a pop-up feature that pops up on link detection. In summary, future work includes: Integration with global URL databases, Browser extensions for real-time detection, and Pop-up alert mechanisms for enhanced usability.

References

[1] Joshi, S. and Joshi, S.M., “Phishing Urls Detection Using Machine Learning Techniques”, Int. J. Comput. Eng. Res. Trends, vol. 6, no. 6, pp. 326–333, Jun. 2019.

[2] M. Aljabri et al., "Detecting Malicious URLs Using Machine Learning Techniques: Review and Research Directions," in IEEE Access, vol. 10, pp. 121395-121417 (2022). doi: 10.1109/ACCESS.2022.3222307.

[3] Patgiri, R., Bhattacharjee, D., Thakare, P., Kotecha, K.: Empirical Study on Malicious URL Detection Using Machine Learning. In: Panda, B., Li, K.C., Balas, V.E., Satapathy, S.C.: Distributed Computing and Internet Technology. ICDCIT 2019. Lecture Notes in Computer Science, Vol. 11319, pp. 363–376 (2019). https://doi.org/10.1007/978-3-030-05366-6_31

[4] Hasan, M.K., Abdulkadir, R. A., Islam, S., Gadekallu, T.R., Safie, N., “A Review on Machine Learning Techniques for Secured Cyber-Physical Systems in Smart Grid Networks. Energy Reports 11, 1268–1290 (2024). https://doi.org/10.1016/j.egyr.2023.12.040

[5] Joshi, A., Das, D., Sharma, A., Singh, P.K.: Efficient Ensemble Learning for Malicious URL Detection. In: Proc. 10th Int. Conf. on Computing, Communication and Networking Technologies (ICCCNT), Kanpur, India, pp. 1–6 (2019).

[6] Aggarwal, A., Rajadesingan, A., Kumaraguru, P.: PhishAri: Automatic Real-Time Phishing Detection on Twitter. In: Proc. IEEE APWG eCrime Research Summit, pp. 1–12 (2012). https://doi.org/10.1109/eCrime.2012.6489521

[7] Alshboul, Y., Nepali, R., Wang, Y.: Detecting Malicious Short URLs on Twitter. In: Proc. 21st Americas Conf. on Information Systems (AMCIS), Puerto Rico, USA (2015).

[8] Altay, B., Dökeroğlu, T., Coşar, A.: Context-Sensitive and Keyword Density-Based Supervised Machine Learning Techniques for Malicious Webpage Detection. Soft Comput. Vol. 23, No. 12, pp. 4177–4191 (2019). https://doi.org/10.1007/s00500-018-3066-4

[9] Anand, A., Gorde, K., Moniz, J.R.A., Park, N., Chakraborty, T., Chu, B.-T.: Phishing URL Detection with Oversampling Based on Text Generative Adversarial Networks. In: Proc. IEEE Int. Conf. on Big Data (BigData*)*, Seattle, WA, USA, pp. 1168–1177 (2018). https://doi.org/10.1109/BigData.2018.8622547

[10] Wu, T., Xi, Y., Wang, M., Zhao, Z.: Classification of Malicious URLs by CNN Model Based on Genetic Algorithm. Appl. Sci. Vol. 12, No.23, 12030 (2022). https://doi.org/10.3390/ app122312030

[11] Joshi, A., Lloyd, L., Westin, P., Seethapathy, S.: Using Lexical Features for Malicious URL Detection—A Machine Learning Approach. arXiv preprint arXiv:1910.06277 (2019).

[12] Aslam, S., Aslam, H., Manzoor, A., Chen, H., Rasool, A.: AntiPhishStack: LSTM-Based Stacked Generalization Model for Optimized Phishing URL Detection. Symmetr*y* Vol. **16, No.**2, 248 (2024). https://doi.org/10.3390/sym16020248

[13] Chong, J.C., Sim, N.Y., Khoh, C.W., Yi, L.T.: Phishing Attack Detection on URLs Using KNN, RF, DT with GA and K-Fold Cross Validation Approach. Int. J. Res. Innov. Social Sci. (Special Issue on Cybersecurity), pp. 1623–1641 (2025). DOI: https://dx.doi.org/10.47772/IJRISS.2025.9010134

[14] Reyes-Dorta, N., Caballero-Gil, P., Rosa-Remedios, C.: Detection of Malicious URLs Using Machine Learning. Wireless Networks, Vol. 30, No. 9, pp. 7543–7560 (2024). https://doi.org/10.1007/s11276-024-03700-w

[15] Tian, Y., Yu, Y., Sun, J., & Wang, Y.: From Past to Present: A Survey of Malicious URL Detection Techniques, Datasets and Code Repositories (2025). arXiv. https://arxiv.org/abs/2504.16449.

[16] Türk, F., & Kılıçaslan, M.: Malicious URL Detection with Advanced Machine Learning and Optimization-Supported Deep Learning Models. Applied Sciences, Vol. 15, No. 18, 10090 (2025). https://doi.org/10.3390/app151810090
[17] Haq, Q. E. u., Faheem, M. H., & Ahmad, I.: Detecting Phishing URLs Based on a Deep Learning Approach to Prevent Cyber-Attacks. Applied Sciences, Vol. 14, No.22, 10086 (2024). https://doi.org/10.3390/app142210086.