Phishing URL Detection Using Machine Learning and NLP

Faizan Manazir
faizanmanazirf@gmail.com

Muhammad Affan
muhammadaffanpakistani@gmail.com

Aoun Muhammad
aoun.muhammad@iub.edu.pk

Umar Fayyaz
umar.fayyaz@iub.edu.pk

Sehrish Raza
sehrishraza6@gmail.com

Faculty of Engineering
The Islamia University of Bahawalpur
Bahawalpur, Pakistan

***Abstract*—These days phishing attacks are a significant source of cybersecurity threat. They defraud others to provide confidential information such as usernames, passwords, and money by using fraudulent websites or malicious links. Most of the common techniques of identifying these attacks such as blacklist or predetermined rules mostly fail to recognize new phishing web addresses as they emerge. We propose a new solution that combines machine learning and natural language processing (NLP) models to identify phishing URLs more in an effective and efficient way in this paper. We use the top domains on Tranco to extract several characteristics of URLs, including lexical patterns, character statistics, entropy, suspicious keywords, and reputation of the domain. To enhance the accuracy of detection with the minimum computational costs, we assembled an ensemble model based on the Random Forest and XGBoost classifiers. Besides, the system has the benefit of providing real time URL analysis in an interactive interface, which is conveniently applicable in both web browsers and security systems. As our results reveal, our method as a way of tackling this issue beats some of the existing methods in terms of accuracy, preciseness, and recall. On a larger scale, we have an opportunity to enhance the security of the Internet by offering a fast, reliable, and scalable method of detecting acts of phishing.**

***Index Terms*—Phishing URL Detection, Machine Learning, Natural Language Processing, Ensemble Learning, Cybersecurity**

I. Introduction

As the services offered by the internet are growing, communication, shopping, banking, and sharing of information are becoming more and more dependent on the online platforms among the users. Despite the convenience offered by these services, new opportunities to cybercriminals have been established. Phishing is one of the most widespread and dangerous cyber threats in which the attacker tries to steal the user by deceiving them to give sensitive data by redirecting the user to a rogue site or a malicious site. These attacks are primarily executed by means of well-designed phishing URLs that resemble real web addresses to a great extent. /image Phishing attacks have been on the increase and are becoming more sophisticated since in the past few years. To overcome conventional protection, attackers keep on coming up with novel URLs. The traditional types of phishing detection, including blacklist-based phishing detection systems and rule-based phishing detection systems, are based on the knowledge of known bad URLs or hard-coded rules. Although these approaches can prevent any known attacks, they usually do not recognize any newly introduced phishing URLs which are also known as zero-day attacks. This weakness limits their applicability in the real world whereby phishing methods are fast developing. [1], [2].

To address these issues, the problem of phishing URLs detection has been studied more frequently by researchers as a machine learning-based method. The machine learning models can comprehend the patterns in large data sets and discover the hidden association in the structure of the URL. A number of studies have indicated that the length of URL, domain structure, special characters and lexical patterns are some of the features that may be used to serve as a useful indicator of the differences between phishing and legitimate URLs. [3], [4].Nonetheless, when using one machine learning model, there is always a possibility of biased predictions, or worse, worse generalization.

Phishing detection has also been improved with the help of natural language processing, which examines textual and statistical characteristics of URLs. Using NLP to extract features is useful in optimalizing small patterns e.g. character distribution and entropy which are typically present in malicious URLs. [5], [6]. Although these results are promising, most of the existing methods do not support the real-time or demand a significant amount of computer resources, and these cannot be easily implemented in real-life.

Ensemble learning methods are more recently of interest because of the capability to use a combination of classifiers and enhance the predictive accuracy. Ensemble based system can have higher performance compared to individual classifiers by exploiting the capabilities of the respective models. [7], [8].Nevertheless, a number of proposed systems remain much more accuracy-oriented without referring to the issues of real-time functionality and deployment.

© 2026 TechShield Publications
DOI: https://doi.org/10.67535/tsp.000002.009
Published under the ISAISS 2026 Proceedings

This work presents a proposal of a real-time phishing URL detection system, which incorporates machine learning and natural language processing methods in an ensemble system of learning. The system is developed in a manner that is light, precise and appropriate to be used in real time. The proposed approach will overcome the shortcomings of the previous researches because it will emphasize real-world application and efficient feature extraction, which will offer stable security against phishing attacks.

II. Literature Review

The problem of phishing URL has been given great interest by the research community because fraud and cybercrime are on the perpetual increase online. Several methods have been suggested over the years to detect phishing attacks starting with the classical rule-based models and followed by sophisticated machine learning and deep learning models. Majority of the available literature is based on examining the features of URL, domain and textual patterns in order to distinguish between phishing and legitimate URLs.

A number of scholars have delved into machine learning-based systems of detecting phishing URLs. The paper by Rehman et al. [1] suggested a real-time phishing detection system with the use of machine learning classifiers and showed that feature-based models can be effective in detecting phishing URLs provided that they are trained on large datasets. A single study conducted by Awotunde et al. [2] evaluated a variety of machine learning algorithms and summarized their findings by stating that ensemble-based algorithms are more effective than single classifiers. The performance of these approaches is highly dependent on feature selection and data quality, although they demonstrate promising results.

Other works have highlighted real-time detection. The machine learning-driven framework proposed by Alhemyari et al. [7] is aimed at detecting phishing attacks in real time. Their research focused on minimizing response time while preserving detection accuracy. In a similar direction, Shoaib and Umar [4] examined URL-based phishing detection using lexical and structural characteristics. Although these approaches are suitable for online detection, they may struggle to detect previously unseen phishing URLs that employ new obfuscation techniques.

Phishing detection has also been widely explored using natural language processing techniques. Sahingoz et al. [3] demonstrated that NLP-based feature extraction, including character distribution and entropy analysis, improves detection performance. Buber et al. [5] further investigated URL tokenization and character-level analysis, showing that NLP methods can detect sophisticated phishing URLs. Later studies by Mariappan et al. [6] confirmed that combining NLP features with machine learning classifiers yields better phishing detection results than using machine learning alone.

In recent years, deep learning-based approaches have gained popularity due to their ability to automatically learn complex patterns. Asiri et al. [8] developed a deep learning-based phishing detection system and achieved high classification performance. Similarly, Linh et al. [9] proposed a real-time phishing detection system using deep neural networks deployed within a browser environment. Despite their high accuracy, deep learning models often require large datasets and substantial computational resources, which can limit their applicability in lightweight or real-time systems.

Although significant progress has been made, several challenges remain unresolved. Many existing methods lack real-time flexibility, involve high computational cost, or fail to generalize well to previously unseen phishing URLs. Furthermore, several studies rely on a single detection approach, which can reduce system robustness. These limitations highlight the need for a practical, real-time, and hybrid phishing URL detection system. To address these gaps, the proposed work integrates machine learning and NLP-based features within an ensemble learning framework while prioritizing real-time usability.

TABLE I
Summary of Selected Phishing Detection Studies

Paper (Reference)	Technique / Method	Contribution	Limitation
Rehman et al., 2025 [1]	ML-based URL analysis	**Real-time phishing detection using ML and fast feature extraction**	**Only URL features considered, no NLP analysis**
Sahingoz et al., 2019 [3]	ML classifiers, lexical URL features	**Highlighted importance of URL length, entropy, character statistics for ML-based detection**	**Dataset limited, may not generalize to new phishing patterns**
Alhemyari et al., 2025 [7]	Ensemble ML, real-time detection	**Real-time detection framework with low latency and high accuracy**	**Computational cost high for large-scale deployment**
Mittal et al., 2022 [10]	ML + NLP	**Combined NLP with ML classifiers to detect phishing URLs and emails, improving accuracy**	**Focused on email content, limited URL dataset evaluation**

Table I summarizes selected phishing detection papers, highlighting their techniques, contributions, and limitations.

It is important to note that references to real-time detection in this paper pertain exclusively to the cited works and do not imply real-time deployment claims for the proposed system.

III. Proposed Methodology

This section describes how the proposed phishing URL detection system is to be designed and how it is going to work. Its implementation procedure follows the backend code precisely and consists of all the key steps, including data preprocessing, generation of features, model learning, and evaluation, as well as real-time interaction. The general aim of the suggested solution is to have an automated, effective, and dependable solution to the detection of phishing URL in a real-world situation.

Figure 1 illustrates the complete processing pipeline of the proposed system, starting from dataset input and feature extraction to final phishing prediction through an interactive command-line interface.

Following the machine learning prediction, an additional domain-level verification step is applied to ensure that URLs belonging to highly trusted domains are not incorrectly flagged as phishing.

A. Dataset Preparation

The proposed system was evaluated using the PhishStorm dataset. Initially the dataset was properly cleaned by removing

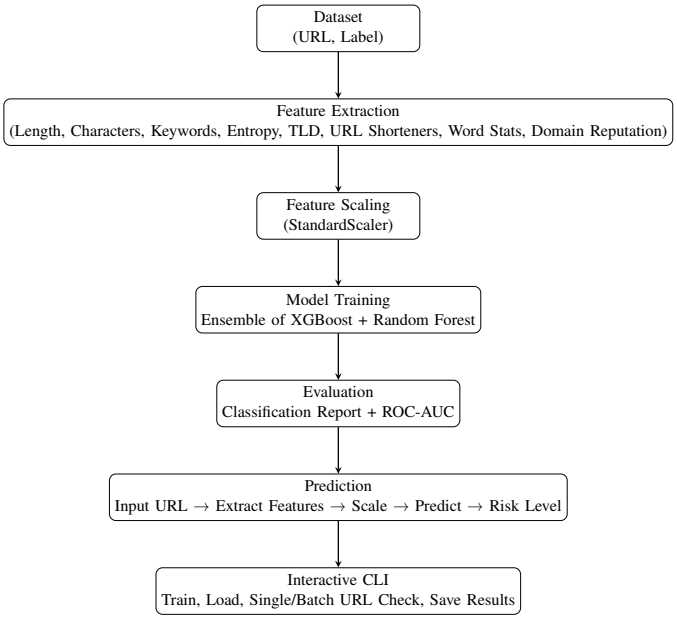

Fig. 1. Proposed Methodology Workflow for Phishing URL Detection

the duplicates, invalid records, and missing entries. Reputable and well-labeled URLs were only kept to ensure the quality of data.

Normalization of the URLs, elimination of noisy samples, and validation of labels were included in the preprocessing pipeline. URL normalization was performed by changing all of the letters to lowercase, as well as by eliminating prefixes of the protocol. The use of synthetic data generation techniques was also avoided to maintain the real-life phishing trends.

The samples contained labeled phishing and legitimate URLs. In preprocessing only `url` and `label` attributes were used. Records containing none or duplicate URL were deleted and labels of other formats (e.g., "phish", "legitimate", "benign") were merged into binary representation.

Tranco top-domain list was added as a domain reputation feature when training the model. The list was considered as a static object outside the system and applied throughout the training and testing. This feature was built without any labels or any information of the test set hence there was no leakage of the data. The Tranco list lacks phishing tags and was only used as a popularity based source instead of a ground truth source. The Tranco list had a fixed snapshot that was employed in all experiments regardless of the time when the data was recorded.

B. Feature Extraction

The extraction of the features is performed with the help of the `URLFeatureExtractor` module that converts raw URLs to numerical values that can be used in machine learning models. The features obtained are divided into few categories:

1) **Length-based features:** These are the total URL length, domain length, path length, and query length. Phishing URLs have a tendency of being overly long or containing superfluous path segments.
2) **Character distribution characteristics:** Dots, hyphens, slashes, question marks, equals signs, the '@' symbol, digits, and alphabetic characters are counted to determine abnormal usage of characters.
3) **Digit-to-letter ratio:** The ratio of digit characters to alphabetic characters is calculated, since phishing URLs often contain a higher number of digits.
4) **Security-related features:** Features such as the presence of HTTPS, use of IP addresses instead of domain names, and the number of subdomains are extracted.
5) **Suspicious keyword analysis:** Typical phishing keywords such as *login, verify, secure, account, bank, password, alert*, and related terms are identified within the URL string.
6) **Entropy features:** The entropy of both the domain name and the entire URL is calculated to measure randomness and obfuscation.
7) **Top-Level Domain characteristics:** The length of the TLD is tracked and suspicious TLDs are flagged.
8) **URL shortening detection:** Redirection-based phishing attacks are detected by checking URLs against commonly used URL shortening services.
9) **Word length statistics:** Maximum and average word lengths are calculated by splitting the URL using standard delimiters.
10) **Domain reputation check:** Each domain is verified against the Tranco top domain list to identify trusted websites.

All extracted features are stored in a structured `pandas` DataFrame. In addition to handcrafted features, character-level NLP features are generated using an n-gram based TF–IDF representation. These textual indicators help identify obfuscated patterns and unusual lexical structures commonly used in phishing URLs. Finally, numerical and NLP-based features are combined into a single feature vector for classification. The approach uses a hybrid representation of features which incorporates hand-crafted statistical features and character modeling based on NLP techniques. From the URL, various structural and lexical features are extracted based on URL length, host name length, path length, number of sub-domains, number of digits, number of letters, digit ratio, number of hyphens, number of dots, use of HTTPS, IP addressing, TLD length, Shannon entropy, use of Suspicious Keywords, and use of Suspicious TLD indicators.

C. NLP-Based URL Representation

Albeit URLs are not sentences in a natural language, structural textual patterns do exist in them, and can be properly explored manually through NLP methods. Character-level n-gram modelling is applied in this work to identify lexical variations and phishing patterns that are hidden.

1) TF-IDF Character N-Gram Modeling: Each URL is decomposed into overlapping character n-grams, and TF-IDF

weighting is applied to represent the importance of each n-gram:

$$\text{TF-IDF}(t,d) = \text{TF}(t,d) \times \log\left(\frac{N}{\text{DF}(t)}\right) \tag{1}$$

This representation enables the detection of subtle manipulations that are often missed by simple lexical features.

2) Entropy-Based Lexical Analysis: To further capture randomness in URLs, Shannon entropy is computed as:

$$H(S) = -\sum_{i=1}^{k} p(c_i)\log_2 p(c_i) \tag{2}$$

Higher entropy values typically indicate obfuscated or automatically generated URLs, which are commonly associated with phishing activity.

D. Dataset Splitting Strategy

Class balancing was done before splitting the data to have equal representation of phishing and valid URLs. The balanced dataset was subsequently separated into training and test sets with an 80 -20 stratified approach to maintain the balance in class in both subsets. The arrangement allows a just performance comparison among classes without having classes dominating in model learning.

E. Feature Scaling

Standardization of numerical features is also done with the help of StandardScaler which makes sure that all the features have unit variance and mean zero. This enhances stability of learning and eliminates control of features with higher numeric values.

F. Model Training

The proposed system employs an ensemble learning approach combining two classifiers:

- **XGBoost:** Configured with 300 estimators, depth of 6, learning rate of 0.1, and subsampling parameters to reduce overfitting.
- **Random Forest:** Uses 300 decision trees with a maximum depth of 15 to provide robust predictions.

Both models are integrated using a soft-voting strategy. Higher weights are assigned to phishing samples during training to improve recall. Performance is evaluated using classification metrics and ROC-AUC score. The proposed system uses an ensemble learning approach. There are two base learners in this approach: an XGBoost classifier and a Random Forest classifier. The Random Forest classifier uses 300 trees along with class weighing to give more importance to phishing examples, whereas the XGBoost uses 300 estimators along with a maximal depth of six and a learning rate of 0.1. The final ensemble is built using soft voting with more importance to XGBoost. The model provides phishing probability scores, and URLs that have probabilities above 0.70 are deemed phishing URLs. The decision threshold used was 0.70, which was not optimized on the test data and was used in all the experiments.

G. URL Prediction

For each given URL features are extracted scaled and then passed on to the trained ensemble model. Results contain classifications as well as risk levels based on probability from LOW to HIGH.

H. Model Persistence

The trained model, the scaler, and the feature metadata can all be stored using joblib and then can be reused for predictions in the

I. Interactive Command-Line Interface

An interactive CLI is also offered to facilitate activities such as training and loading models individual URL verification and bulk URL analysis.

J. Domain Whitelisting

As a domain whitelisting layer using the Tranco Top 1m list this prevents false positives in trusted domains during prediction. The whitelisting of domains applied during deployment is done to minimize false positives, and all assessed evaluation metrics are calculated without whitelisting.

IV. Experimental Results

To evaluate the proposed system, 80–20 class-wise hold-out validation is performed. The final test set contains 19,152 URLs with equal class distribution. Precision, recall, F1-score, accuracy, and ROC-AUC are used as performance metrics. Since in real world network traffic, phishing URLs are typically a tiny fraction of all the legitimate ones, to further analyze the performance of our system, apart from the balanced setup, the system is also tested on the imbalanced set without any kind of resampling.

The model was able to keep a high level of recall for phishing URLs in such a scenario, which ensured that the system is able to identify the malicious URL, no matter how rare the occurrence is. Although the level of accuracy dipped slightly because of the imbalance in the data, such a trend is acceptable in security-sensitive applications.

Table II reports the classification performance of the proposed ensemble model. An overall accuracy of 0.95 is achieved, indicating strong detection capability. The high recall for legitimate URLs confirms that trusted websites are rarely misclassified, while the phishing detection results demonstrate reliable identification of malicious URLs.

TABLE II
Performance of the Proposed Phishing URL Detection System

Class	Precision	Recall	F1-score	Support
Legitimate (0)	0.92	0.99	0.96	9,576
Phishing (1)	0.99	0.91	0.95	9,576
Accuracy		0.95		19,152
ROC–AUC		0.994		–

This high ROC-AUC value strengthens the indication that the proposed model performs a good job in distinguishing between phishing and legitimate URL links.

TABLE III
PERFORMANCE COMPARISON OF BASELINE CLASSIFIERS AND THE PROPOSED ENSEMBLE MODEL ON THE BALANCED TEST DATASET

Model	Accuracy	Precision	Recall	F1-score
Random Forest	0.93	0.92	0.94	0.93
XGBoost	0.94	0.95	0.92	0.93
Proposed Ensemble	**0.95**	**0.96**	**0.95**	**0.95**

The ROC–AUC scores obtained by Random Forest, XGBoost, and the proposed ensemble model were 0.985, 0.989, and 0.994, respectively.

To further evaluate real-world deployment scenarios, the proposed system was additionally evaluated on a naturally imbalanced test set without applying any resampling techniques. The imbalanced dataset consisted of 9,602 legitimate URLs and 4,331 phishing URLs, reflecting the class skew commonly observed in real-world environments. The same decision threshold used in the balanced evaluation was retained, and no additional threshold tuning was performed.

TABLE IV
PERFORMANCE OF THE PROPOSED ENSEMBLE MODEL ON THE NATURALLY IMBALANCED TEST DATASET (NO RESAMPLING APPLIED).

Class	Precision	Recall	F1-score	Support
Legitimate	0.96	1.00	0.98	9,602
Phishing	0.99	0.91	0.95	4,331
Accuracy		0.97		13,933
Macro Avg	0.98	0.95	0.96	13,933
Weighted Avg	0.97	0.97	0.97	13,933

The ROC–AUC score obtained on the naturally imbalanced test set was 0.991, indicating strong discriminative capability despite class skew.

V. DISCUSSION

The phishing URL detection system proposed here has high performance as it has an overall accuracy of 95% and high precision and recall of legitimate and phishing URLs. The ensemble model can check the structural anomalies and lexical patterns at the character level, thus it is capable of capturing both the structural anomalies and subtle lexical patterns employed in phishing attacks due to the combination of handcrafted URL features and handcrafted character-level TF-IDF representations.

From a computational point of view, the proposed system is light and can be used for real-time deployment. Feature extraction is based on simple lexical and statistical operations and inference has been realized by tree-based ensemble models, with low latency per URL with standard CPU hardware. Compared to methods based on deep learning, the system does not suffer from expensive training and inference overheads while being able to keep up with the detection performance.

Domain whitelisting is useful to avoid misclassification of trusted websites whereas feature scaling and data balancing enhance model generalization. The fact that the ROC-AUC score is 0.994 proves that the system can differentiate between phishing and legitimate URLs.

Despite the robustness and the suitability of the system in the real-time implementation, the incorporation of real-time threat intelligence and the assessment of multilingual URLs would increase the effectiveness of the system.ven with this proposed method, accuracy is quite high, but it relies entirely on carefully handcrafted features. Such a dependence on feature engineering might hinder scalability, especially with obfuscating tactics developed by phishers to obfuscate their URL strings. techniques might be of great benefit in such a scenario.

VI. CONCLUSION

The paper provides a system of automatic and efficient phishing URL detection in real-time using a combination of hand-created URL features and character-based NLP representations. XGBoost and Random Forest ensemble model is very accurate, precise and recalls, thus successful in separating between phishing and legitimate URLs. Reliability is increased even further through domain whitelisting and careful dataset preparation which reduce false positives. The high ROC-AUC score makes it clear that the system has high discriminative capability, which allows using it in a real-life application in the area of cybersecurity. All in all, the suggested solution is efficient, scalable, and robust to prevent proactive phishing detection, which can be used to make the online browsing process safer.

VII. FUTURE WORK

The next steps will be to add the features that will be useful in the conditions of the more sophisticated phishing methods, like homograph attacks and obscured URLs. Also, the incorporation of deep learning structures, as well as deploying the system in browser extensions or network-based security platforms will be considered to make it more applicable to real-life use.n particular, deep learning models like CNNs, RNNs, and Transformer-based architectures will be explored in future research for automatically learning the representations directly from raw URLs, dispelling manual feature engineering. We also plan to include threat intelligence feeds, multilingual analysis of the URL, and adversarial robustness evaluation to further enhance real-world applicability.

REFERENCES

[1] A. U. Rehman, I. Imtiaz, S. Javaid, and M. Muslih, "Real-time phishing url detection using machine learning," *Engineering Proceedings*, vol. 107, no. 1, p. 108, 2025.

[2] J. B. Awotunde, C. O. Abikoye, A. L. Imoize, Y. Mejdoub, and O. A. Imran, "An improved phishing url detection using machine learning methods," in *International Conference on Connected Objects and Artificial Intelligence*. Springer, 2025, pp. 481–488.

[3] O. K. Sahingoz, E. Buber, O. Demir, and B. Diri, "Machine learning based phishing detection from urls," *Expert Systems with Applications*, vol. 117, pp. 345–357, 2019.

[4] M. Shoaib and M. S. Umar, "Url based phishing detection using machine learning," in *2023 6th International Conference on Information Systems and Computer Networks (ISCON)*. IEEE, 2023, pp. 1–7.

[5] E. Buber, B. Diri, and O. K. Sahingoz, "Nlp based phishing attack detection from urls," in *International Conference on Intelligent Systems Design and Applications*. Springer, 2017, pp. 608–618.

[6] E. Mariappan, C. J. C. Grace, S. J. P. Gnanaraj, D. E. Paulsyah, and N. Muthukumaran, "Phishing website detection using natural language processing," in *2024 International Conference on Inventive Computation Technologies (ICICT)*. IEEE, 2024, pp. 1639–1643.

[7] A. A. Alhemyari, Z. A. Hassan, M. A. Saeed, R. A. Alselwi, and R. A. Alhemyari, "Real-time phishing attack detection using machine learning," in *2025 5th International Conference on Emerging Smart Technologies and Applications (eSmarTA)*. IEEE, 2025, pp. 1–8.

[8] S. Asiri, Y. Xiao, S. Alzahrani, and T. Li, "Phishingrtds: A real-time detection system for phishing attacks using a deep learning model," *Computers & Security*, vol. 141, p. 103843, 2024.

[9] D. M. Linh, H. D. Hung, H. M. Chau, Q. S. Vu, and T.-N. Tran, "Real-time phishing detection using deep learning methods by extensions," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 14, no. 3, pp. 3021–3035, 2024.

[10] A. Mittal, D. D. Engels, H. Kommanapalli, R. Sivaraman, and T. Chowdhury, "Phishing detection using natural language processing and machine learning," *SMU Data Science Review*, vol. 6, no. 2, p. 14, 2022.