

MUAD-Net: A Multi-scale URL Anomaly Detection Network for Malicious URL Identification

Tian Liang

Qingdao University, Qingdao 266071, China

Email: liangtian@qdu.edu.cn

How to cite this paper: Liang, T. (2026). MUAD-Net: A multi-scale URL anomaly detection network for mal malware URL identification. *Journal of Computer Science and Frontier Technologies*, 3(2), 167–176. ISSN Print: 3104-4204, ISSN Online: 3104-4212.

<https://doi.org/10.63313/JCSFT.9080>

Published: 2026-05-30

Copyright © 2026 by author(s) and Erytis Publishing Limited.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Abstract

With the continuous evolution of cyber attack techniques, the propagation of malicious links has become a major threat in cybersecurity. Attackers induce users to visit target resources by constructing malicious URLs, leading to information theft and malware distribution. Effectively identifying malicious URLs in complex network environments has become an important research issue. To address this problem, this paper adopts an attack life-cycle perspective and proposes a deep learning-based malicious URL detection method. Considering the complex structure, short lifecycle, and fast generation speed of malicious URLs, the method first normalizes URL strings and encodes them as character sequences, then maps characters into vector representations through an embedding layer. A multi-scale convolutional structure is used to extract local pattern features from URL character sequences, combined with a sequence modeling mechanism to capture contextual dependencies, achieving automatic learning of malicious URL features. Experimental results show that the proposed method achieves high detection performance and good generalization ability. Through multi-scale feature extraction and model optimization, this paper effectively improves malicious URL detection performance, demonstrating good application value in real-world network scenarios.

Keywords

Alicious URL Detection; Deep Learning; Multi-Scale Feature Extraction; Convolutional Neural Network; Sequence Modeling

1. Introduction

With the continuous expansion of internet applications, URLs have become a crucial gateway for users to access online resources. In real-world cyber attacks, adversaries often construct malicious URLs to lure users into visiting phishing websites, downloading malware, or establishing communication with remote command-and-control servers, thereby conducting malicious activities such as information theft, ransomware attacks, and remote control. Compared with

traditional attack methods, malicious URLs are characterized by low propagation cost, high spreading speed, and diverse techniques, thus occupying a significant position in the cyber attack chain.

In practical network environments, a URL typically consists of components such as domain name, path, and parameters, and may take various complex forms, including short URLs, encoded URLs, obfuscated URLs, and parameterized URLs. These structural characteristics make URLs highly flexible in terms of representation, while also increasing the difficulty of malicious URL detection. Furthermore, URL behavior is often closely related to contextual access information. Features such as access time, source IP address, and access frequency can reflect potential anomalous patterns of URL access behavior, providing important auxiliary cues for malicious URL detection. However, traditional detection methods often fail to fully utilize such contextual information, thereby limiting their overall detection performance.

Existing malicious URL detection methods mainly include blacklist-based detection, rule-based matching, and machine learning approaches. Blacklist-based detection maintains a database of known malicious URLs for matching and identification. Although this method is simple to implement and highly efficient, it struggles to detect newly emerging or variant malicious URLs. Rule-based matching relies on manually designed detection rules, such as URL length, special character ratio, and domain name structure. However, this approach heavily depends on expert knowledge and cannot easily adapt to rapidly evolving attack strategies. With the advancement of deep learning techniques, researchers have attempted to leverage neural network models for automatic feature learning from URL character sequences. For example, convolutional neural networks (CNNs) are used to extract local character patterns from URLs, while recurrent neural networks (RNNs) capture contextual dependencies within URL sequences. Although these methods have improved the accuracy of malicious URL detection to some extent, they still suffer from several limitations, including insufficient mining of URL semantic information, lack of effective fusion mechanisms for heterogeneous features, and degraded detection performance under imbalanced data conditions.

To address the above issues, this paper proposes a deep learning-based malicious URL detection method, named MUAD-Net. The method constructs a multi-branch feature extraction architecture to jointly model local pattern features, sequential dependency features, and structural semantic information from URL character sequences. In addition, an attention mechanism is introduced to achieve adaptive fusion of multi-dimensional features, thereby enhancing both the detection performance and generalization ability of the malicious URL detection model.

2. Related Works

Early malicious URL identification mainly relied on blacklist and rule-based matching methods, which determine whether a link is malicious by maintaining a

known set of malicious URLs or constructing specific rule templates. Garera et al. [1] conducted early research on the detection and measurement of phishing attacks and analyzed the role of URLs in phishing website identification. Whittaker et al. [2] proposed a large-scale automated phishing page classification method, promoting the evolution of phishing detection from manual rules to automatic analysis. Although these methods are simple to implement and convenient for engineering deployment, they exhibit weak adaptability to scenarios such as newly generated malicious URLs, domain variants, and short-link redirections.

In the machine learning phase, researchers began to construct classification models using lexical features, domain features, and structural features of URLs. Ma et al. [3] modeled malicious website detection as a supervised learning task, achieving malicious webpage identification by analyzing URLs and related attributes. Sahingoz et al. [4] employed URL lexical features for phishing detection, demonstrating the effectiveness of machine learning methods in malicious URL identification. Verma et al. [5] emphasized efficient detection of malicious URLs through fast feature extraction. Meanwhile, some studies attempted to combine online learning and lightweight models to achieve malicious URL detection better suited for dynamic environments. Blum et al. [6] proposed an online learning method based on lexical features for phishing URL detection. April et al. [7] achieved phishing website identification using URL features combined with machine learning methods. These studies indicate that even without accessing webpage content or behavioral information, effective malicious URL detection can still be achieved based solely on URL strings.

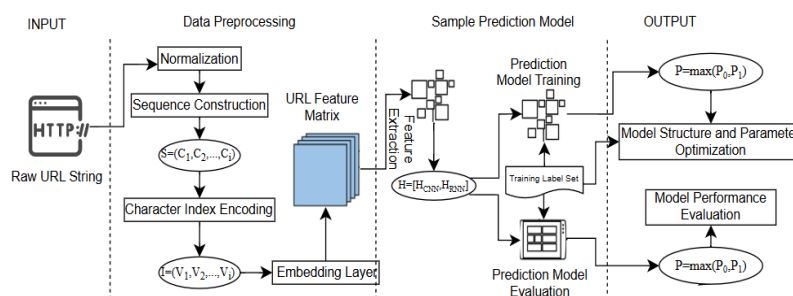
With the advancement of deep learning techniques, malicious URL detection has gradually entered the stage of automatic feature learning. Le et al. [8] proposed URLNet, which models URLs by fusing character-level and word-level representations, becoming one of the representative works in malicious URL detection. Liu et al. [9] utilized deep learning models to model URL character sequences, achieving automatic malicious URL identification. Compared with traditional feature engineering methods, these approaches can directly learn discriminative patterns from raw URL strings, reducing the limitations imposed by manual feature design. Furthermore, Huang et al. [10] employed deep learning methods for malicious URL detection, further validating the effectiveness of deep models in complex URL scenarios. Aljofey et al. [11] proposed a deep learning-based malicious URL detection method, enhancing detection performance through stronger sequence modeling capabilities. Srinivasan et al. [12] employed character-level deep representations for phishing URL detection, demonstrating that character-level pattern learning is significant for identifying complex malicious links. Liu et al. [13] proposed a deep learning-based malicious URL detection method that extracts multi-level structural features from URLs using convolutions at different granularities. In addition, Reyes-Dorta et al. [14] provided a systematic

review of machine learning methods for malicious URL detection, summarizing current trends in feature construction, model design, and experimental evaluation. Barik et al. [15] proposed a web phishing URL detection model in 2025 that combines feature optimization with deep learning to enhance phishing link identification. Do et al. proposed a malicious URL detection model based on temporal convolutional networks, demonstrating that temporal convolutional structures are also suitable for URL sequence modeling. Hu et al. proposed a Hybrid Binary Neural Tree model, applying tree-structured networks to malicious URL detection, offering new insights for deep structural modeling. Chen et al. further discussed the evaluation of malicious URL embedding methods from a representation learning perspective, indicating that URL detection is gradually moving from a pure classification task toward more fine-grained semantic modeling. In summary, malicious URL detection research has gradually evolved from static detection methods based on blacklists and rule templates to automated detection methods based on deep representation learning. Although existing deep learning-based methods have achieved considerable success in malicious URL detection, most approaches still focus on single-scale or single-type feature representations, leaving further research space in multi-scale and multi-type feature representation.

3. Methodology

This section presents the overall architecture and core modules of MUAD-Net, the proposed malicious URL detection method. As illustrated in Figure 1, the method consists of two main stages: URL data preprocessing and malicious URL prediction. In the preprocessing stage, raw URL strings are normalized, converted into character sequences, and transformed into feature matrices via an embedding layer. In the prediction stage, the model learns deep feature representations and outputs the final classification results through a Softmax classifier.

Figure 1. Workflow of MUAD-Net



3.1. URL Sequence Representation

A URL typically consists of several components, including protocol, subdomain, domain, port, path, query parameters, and fragment. Unlike natural language text,

URL strings often contain special characters (e.g., “/”, “.”, “?”, “=”), digits, and encoded characters (e.g., “%20”). These characteristics make URLs flexible in form but also increase the difficulty of detection. Therefore, a key challenge in malicious URL detection is how to preserve the original structural information while converting URLs into vector representations suitable for deep learning models.

In this paper, we adopt a character-level modeling approach for URL sequence representation. Raw URL strings are mapped into fixed-length character sequences and then transformed into continuous vector representations via an embedding layer. This process consists of three steps: character sequence construction, character index encoding, and character embedding.

3.2. Multi-scale Feature Extraction

After obtaining the vectorized URL representation, MUAD-Net further extracts deep semantic features from the character sequence. URLs contain both local character patterns and long-range sequential dependencies. A single feature extraction model is often insufficient to capture such multi-level structural information. To address this issue, this paper designs a multi-scale feature extraction architecture that integrates a Convolutional Neural Network (CNN) and a Bidirectional Long Short-Term Memory (Bi-LSTM) network, as shown in Figure 2

Local Feature Extraction. A 1D CNN is introduced to capture local patterns in URL strings. Attackers often embed deceptive keywords (e.g., “login”, “verify”, “secure”) as contiguous character sequences. Convolutional operations can automatically learn these discriminative patterns within a sliding window. Given the input matrix X and a convolution kernel of size k , the convolution operation is defined as:

$$h_t = f(W_k \cdot X_{t:t+k-1} + b) \quad (1)$$

where W_k is the kernel weight, b is the bias, and $f(\cdot)$ is the ReLU activation function. To handle pattern length variations, we adopt a multi-scale convolution strategy. Parallel convolution branches with kernel sizes 3, 5, and 7 are used. Smaller kernels capture fine-grained character combinations, while larger kernels perceive longer local semantic patterns. After convolution, max pooling is applied to extract the most salient features and reduce dimensionality. The outputs from different kernel sizes are then concatenated to form the local feature vector F_{cnn} :

$$F_{cnn} = [z_1, z_2, z_3] \quad (2)$$

where z_i represents the features extracted by the i -th kernel size.

Global Dependency Modeling. Although CNNs are effective at extracting local patterns, they have limitations in modeling long-range dependencies. In URLs, critical information such as path hierarchy and parameter correlations often

requires modeling over long contexts. To address this, we introduce a Bi-LSTM network for sequence modeling. Bi-LSTM constructs both forward and backward propagation paths, enabling the model to utilize both past and future context information. The forward and backward hidden states at time step t are computed as:

$$\vec{h}_t = \text{LSTM}(x_t, \vec{h}_{t-1}) \quad (3)$$

$$\overleftarrow{h}_t = \text{LSTM}(x_t, \overleftarrow{h}_{t+1}) \quad (4)$$

The final hidden state is the concatenation $h_t = [\vec{h}_t; \overleftarrow{h}_t]$. The output of the Bi-LSTM layer is denoted as F_{lstm} , which captures the global sequential dependencies within the URL.

3.3. Fusion, Classification, Optimization

After extracting multi-scale local features and global sequential features, the model integrates these complementary representations. The local feature vector F_{cnn} and the sequential feature vector F_{lstm} are concatenated to form the unified URL semantic representation F . This fusion strategy preserves both local sensitive patterns and global dependency information, significantly enhancing the model's ability to handle complex and obfuscated URLs.

The fused feature vector is then passed through fully connected layers for nonlinear mapping. A Softmax function outputs the probability distribution over the two classes (benign vs. malicious). The prediction is determined by the class with the highest probability:

$$\hat{y} = \arg \max(p_{\text{benign}}, p_{\text{malicious}}) \quad (5)$$

To improve training stability and generalization, the following optimization strategies are adopted. The cross-entropy loss function is used to guide model optimization. Dropout is applied between feature extraction and classification layers to reduce overfitting. Normalization operations are incorporated within convolutional and LSTM layers. The Adam optimizer is employed for parameter updates, which adaptively adjusts learning rates across different parameter dimensions.

Through the above fusion and optimization mechanisms, MUAD-Net effectively leverages multi-level and multi-granularity semantic information from URL strings. It achieves significant improvements in detection accuracy, recall, and generalization ability while maintaining low computational complexity, providing a solid foundation for subsequent experimental evaluation.

4. Experimental

The experimental environment in this chapter is consistent with that described in Chapter 3. Therefore, hardware and software configurations are not repeated here. All experiments are conducted using the computing devices, operating system, and deep learning framework. Model training and analysis are implemented in Python

using the PyTorch deep learning framework.

4.1. Setup

The experimental environment in this chapter is consistent with that described in Chapter 3. Therefore, hardware and software configurations are not repeated here. All experiments are conducted using the same computing devices, operating system, and deep learning framework listed in Chapter 3. Model training and analysis are implemented in Python using the PyTorch deep learning framework.

Regarding training optimization, the Adam optimizer is adopted with an initial learning rate of 1×10^{-3} . The batch size is set to 64, and the model is trained for 50 epochs with a dropout rate of 0.5. For sequence embedding, the maximum URL length is set to 128, and the character embedding dimension is also 128. In terms of network architecture, multi-scale convolutional kernels of sizes 3, 5, and 7 are used, with 64 channels, while the Bi-LSTM layer contains 128 hidden units.

To comprehensively evaluate MUAD-Net, multiple metrics are adopted, including Accuracy, Precision, Recall, and F1 Score. Precision measures the accuracy of detected malicious URLs, Recall reflects the model's ability to identify malicious samples, and the F1 Score balances Precision and Recall. Given the class imbalance problem in malicious URL detection, this study focuses on Accuracy and F1 Score to validate MUAD-Net's detection performance.

4.2. Datasets

To objectively evaluate MUAD-Net, three public authoritative datasets are combined to construct the experimental dataset. Public datasets are widely used in malicious URL detection research, ensuring good reproducibility and comparability of results. Moreover, most samples originate from real internet environments, reflecting actual URL distribution characteristics and providing a reliable basis for evaluating model generalization ability.

Three representative public datasets are selected: the Kaggle Malicious URL Dataset, ISCX URL-2016 Dataset, and URLNet Dataset. The Kaggle dataset contains URL samples from real internet environments, covering phishing websites, malware download links, and scam websites. Approximately 20,000 valid URL samples are extracted from this dataset, with labels verified to ensure quality. The ISCX URL-2016 dataset, released by a Canadian cybersecurity research institution, comes from real enterprise network traffic and reflects URL access behavior in office networks. Approximately 11,000 fully labeled URL samples are selected. The URLNet dataset contains URL samples with obfuscated characters or short-link redirections, widely used in deep learning-based malicious URL detection research. Approximately 15,000 samples are selected to enhance recognition of complex and obfuscated URLs.

After data collection, samples from different sources are integrated, followed by

deduplication and cleaning. Duplicate URLs are removed to avoid sample overlap between datasets. Outliers such as empty URLs, invalid URLs containing only digits or symbols, and incorrectly formatted URLs are filtered out. Ambiguous labels are rechecked to reduce annotation noise. After preprocessing, the final dataset contains approximately 46,000 URL samples, including 26,000 benign and 20,000 malicious URLs. Table 4.1 shows the source and distribution of samples.

Table 4.1. Dataset Sources and Sample Distribution

Dataset	Benign URLs	Malicious URLs
Kaggle	11,500	8,500
ISCX-URL	6,500	4,500
URLNet	8,000	7,000
Total	26,000	20,000

4.3. Results

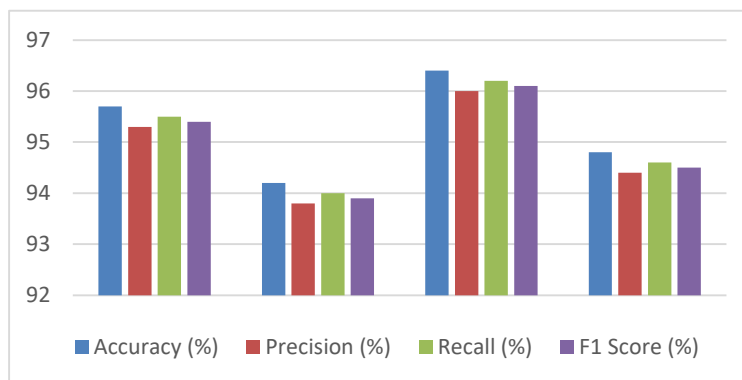
To analyze the contribution of each component to overall performance, an ablation study is conducted by removing key modules, including removing the CNN branch, removing the Bi-LSTM branch, and removing the multi-scale convolution structure. Table 4.2 shows the results.

Table 4.2. MUAD-Net Ablation Study Results

Model Configuration	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)
Full Model	96.6	96.1	96.8	96.4
Without CNN Branch	94.5	94	93.8	93.9
Without Bi-LSTM Branch	94.2	93.7	93.4	93.5
Without Multi-scale CNN	95.1	94.6	94.3	94.4

As shown in Table 4.2, removing the CNN branch reduces the F1 Score by approximately 2.5%, indicating that the CNN plays an important role in extracting local patterns from URL character sequences. Removing the Bi-LSTM branch reduces the F1 Score by approximately 2.9%, demonstrating the importance of sequential modeling for capturing contextual relationships between URL characters. Removing the multi-scale convolution structure also leads to performance degradation, indicating that different kernel sizes effectively extract URL features at different granularities. The ablation results reveal a clear complementary relationship between the CNN and Bi-LSTM branches. CNN extracts local structural features from URL character sequences, while Bi-LSTM captures long-range dependencies. Their combination significantly improves malicious URL detection performance.

Figure 4.2. MUAD-Net Detection Performance on Different Malicious URL Types



To evaluate MUAD-Net's generalization ability across different attack types, multi-type malicious URL classification experiments are conducted. Figure 4.2 shows the results.

MUAD-Net achieves excellent and stable detection performance across multiple typical malicious attack types, including phishing URLs, malware download URLs, spam URLs, and command-and-control server URLs. Specifically, the model performs best on spam URL detection, achieving 96.4% accuracy. It also maintains high accuracy of 95.7% on phishing URL detection, demonstrating strong adaptability to phishing attack scenarios. Even on more challenging tasks such as malware download URLs and command-and-control server URLs, the model achieves 94.2% and 94.8% accuracy, respectively, consistently maintaining high performance.

5. Conclusion

This paper addresses the challenges of malicious URL detection in complex network environments. A deep learning-based method named MUAD-Net is proposed, which transforms raw URL strings into fixed-length vector representations through character-level sequence modeling. Multi-scale convolutional neural networks are employed to extract local pattern features, while bidirectional LSTM captures long-range contextual dependencies. These complementary features are fused to generate comprehensive URL semantic representations for classification.

Extensive experiments conducted on public datasets demonstrate that MUAD-Net achieves superior performance across multiple evaluation metrics, with 96.6% accuracy and 96.4% F1 score, outperforming both traditional machine learning methods and state-of-the-art deep learning baselines. Ablation studies further validate the complementary roles of the CNN and Bi-LSTM branches. The model also exhibits strong generalization ability across different attack types and robustness against obfuscated URLs.

Future work will explore lightweight model design for real-time deployment and investigate the integration of attention mechanisms for further performance improvement.

References

- [1] Garera S ,Provov N, Chew M, et al. A framework for detection and measurement of phishing attacks[C]//Proceedings of the 2007 ACM workshop on Recurring malware.2007:1-8
- [2] Whittaker C, Ryner B, Nazif M. Large-Scale Automatic Classification of Phishing Pages[C]//Ndss.2010,10:2010.
- [3] Ma J, Saul L K, Savage S, et al. Beyond blacklists: learning to detect malicious web sites from suspicious URLs[C]//Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining.2009:1245-1254.
- [4] Sahingoz O K, Buber E, Demir O, et al. Machine learning based phishing detection from URLs[J].Expert Systems with Applications,2019,117:345-357.
- [5] Verma R, Das A. What's in a url:Fast feature extraction and malicious url detection[C]//Proceedings of the 3rd ACM on International Workshop on Security and Privacy Analytics.2017:55-63.
- [6] Blum A, Wardman B, Solorio T, et al. Lexical feature based phishing URL detection using online learning[C]//Proceedings of the 3rd ACM Workshop on Artificial Intelligence and Security.2010: 54-60.
- [7] Ali W. Phishing website detection based on supervised machine learning with wrapper features selection[J].International Journal of Advanced Computer Science and Applications,2017,8(9).
- [8] Le H, Pham Q, Sahoo D, et al. URLNet: Learning a URL representation with deep learning for malicious URL detection[J]. arXiv preprint arXiv:1802.03162,2018.
- [9] Liu R, Wang Y, Guo Z, et al. TransURL: Improving malicious URL detection with multi-layer Transformer encoding and multi-scale pyramid features[J].Computer Networks,2024,253:110707.
- [10] Sriram S, Soman K P. Malicious URL detection using deep learning[J]. Authorea Preprints,2023.
- [11] Aljofey A, Bello S A, Lu J, et al. Bert-phishfinder: A robust model for accurate phishing url detection with optimized distilbert[J].IEEE Transactions on Dependable and Secure Computing,2025.
- [12] Srinivasan S, Vinayakumar R, Arunachalam A, et al. DURLD: Malicious URL detection using deep learning-based character level representations[M]//Malware analysis using artificial intelligence and deep learning. Cham: Springer International Publishing,2020:535-554.
- [13] Liu D J, Geng G G, Zhang X C. Multi-scale semantic deep fusion models for phishing website detection[J].Expert Systems with Applications,2022,209:118305.
- [14] Reyes-Dorta N, Caballero-Gil P, Rosa-Remedios C. Detection of malicious URLs using machine learning[J].Wireless Networks,2024,30(9):7543-7560.
- [15] Barik K, Misra S, Mohan R. Web-based phishing URL detection model using deep learning optimization techniques[J].International Journal of Data Science and Analytics,2025,20(5):4449-4471.