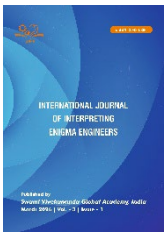
International Journal of Interpreting Enigma Engineers (IJIEE)
https://ejournal.svgacademy.org/index.php/ijiee/
https://doi.org/10.62674/ijiee.2025.v3i01.004
eISSN: 3048-6300
OPEN ACCESS

A COMPREHENSIVE STUDY ON MALICIOUS URL DETECTION: LEVERAGING LARGE-SCALE WEB DATA FOR ACCURATE AND SCALABLE THREAT IDENTIFICATION



Posina Anusha[1], L. Charitha[2]

Original Article

[1]Assistant Professor, Department of AI & DS, Annamacharya Institute of technology & sciences, Tirupati.
[2] Assistant Professor, Department of C.S.E, Annamacharya Institute of technology & sciences, Tirupati.

*Corresponding Author's Email: [1]anusha.ksrm@gmail.com, [2]charithaaits2021@gmail.com

Abstract

The rapid growth of cyber-attacks launched through the internet, such as phishing, spreading of malware, and cyber-attacks involving hacking of websites, has added a sense of challenge in malicious URL detection. Conventional techniques that rely upon blacklists of malicious patterns lack efficient strategies for handling dynamically changing URLs. Keeping these limitations in mind, in the suggested research work, a novel approach has been introduced for malicious URL detection using techniques in deep learning and ensembling, wherein an efficient approach for classifying large-scale data is being proposed using Convolutional Neural Networks, Bidirectional Long Short-Term Memory, and XGBoost. The data on which experiments are carried out is a publicly available large-scale dataset that consists of more than 650,000 URLs, which can be classified as benign, phishing, defacement, and malware types. The model that is proposed in this research work is compared with other approaches using various baseline techniques such as logistic regression, SVM, XGBoost, and CNN. Performance parameters that are used are accuracy, precision, recall, F1 score, ROC curve, and confusion matrix. The experimental results have shown that the proposed model achieves an accuracy of 96%, compared to all the other models, and hence proves that simply by combining the concepts of deep sequential features and gradient boosting, a better model can be obtained that can give better results while detecting malicious URLs.

Index Terms – Malicious URL Detection, Cybersecurity, Deep Learning, CNN, BiLSTM, XGBoost, Hybrid Model, Web Threat Detection.

Introduction

Malicious Uniform Resource Locator (URL) detection is one crucial problem in cybersecurity that benefits significantly from addressing the challenges posed by an increase in threats such as phishing, malware, and web fraud. Conventional approaches to detecting malicious URLs, such as blacklisting, are found to be inadequate in terms of generality and dynamic response to new patterns of attacks. Over the last few years, machine learning and deep learning techniques have provided a solution by their ability to automatically learn patterns from large data sets, as opposed to conventional approaches that are mostly dependent on handcrafted rules.

Contemporary approaches place a corresponding focus on the complexity, in terms of detection, of malicious URLs, as well as the inability of feature engineering techniques to properly tackle the problem, thus necessitating the application of deep learning techniques to address the problem by improving detection capabilities regarding various types of threats, as described in [1] and [2]. Contemporary research indicates the ability of contemporary deep learning techniques, such as CNN, RNN, or transformer networks, to recognize inherent patterns in URLs, without having to heavily rely on preprocessing techniques, thus proving the advantage of deep learning approaches in terms of pattern

recognition, as described in [3] and [4]. Embedding techniques, such as those based on LLMs, have also been proposed for real-time detection, as described in [5].

Apart from deep learning models, the application of hybrid models that integrate machine learning models such as XGBoost, random forest, or support vector machine classifiers with neural feature extraction models has also recorded remarkable performance increases for URL detection accuracy. Hybrid models utilize the strength of statistical classifiers and neural feature extraction models to better differentiate between harmless or malware URLs in different attack scenarios [6], [7]. Other research studies in this area have also been aimed at developing or optimizing different architectures for URL detection models using models such as temporal convolutional networks combined with attention mechanisms in order to address the limitations of conventional CNN/RNN models in feature extraction.

On the one hand, comprehensive studies conducted to comparatively evaluate these systems emphasize the advantages of these ensemble-based systems in boosting performance measures such as precision, recall, and F1-score with benchmark tests conducted on different varieties of URL-based threats [9]. Nevertheless, new research on the domain highlights a variety of issues such as those of sparsity, explainability of these systems, and efficiency in feature representation, providing new directions to this domain of study [10].

In our work, we extend this recent research based on a hybrid system with combinations of CNN, BiLSTM, and XGBoost components with a variety of baselines such as those using logistic regression, random forests, and gradient boosting.

Our main contributions are as follows:

- Novel CNN-BiLSTM-XGBoost Hybrid Detection Framework: A new malicious URL detection framework based on a proposed combination of CNN, BiLSTM, and XGBoost machine learning algorithms has also been designed. In this framework, instead of using a deep neural network with a softmax layer for classifying malicious URL attacks, gradient-boosted trees are used to increase prediction robustness.
- End-to-End Learning from Raw URLs without Heavy Handcrafted Rules: The approach made possible through our solution can significantly enable effective learning directly from raw URLs, where we can benefit from "deep" neural feature extractions together with "lightweight" statistical representations, minimizing our dependence on "heavy" handcrafted feature extractions and associated rules.
- Large Scale Multi-Class Evaluation on Realistic Imbalanced Data: In this section, extensive experiments have been carried out using a large-scale dataset consisting of more than 650,000 URLs. These URLs belong to the class of benign, phishing, defacement, and malware data. The model shows strong evaluation with realistic class imbalance issues, which were a drawback of most of the previous models.
- Demonstrated Performance Superiority and Practical Reliability: The hybrid approach’s performance demonstrates a high level of overall accuracy, standing at 96%, which can be said to mark a significant improvement relative to existing machine learning and deep learning techniques like decision trees, linear regression, support vector machines, and deep learning.

Literature Survey

The growing popularity of phishing, malware, ransomware, and defacement attacks via deceitful web links has made malicious URL detection a critical research issue in cybersecurity. Blacklist systems of the past are no longer adequate, as attacks can now create new malicious URLs or can compromise the existing URLs by manipulation to avoid detection.

The use of transformer-based contextual URL representation architectures has become a significant development in this direction. Chen and Meng [11] introduced a metadata-based malicious URL detector model based on the RoBERTa-Large together with multi-source network threat intelligence. Their system combines the contextual subword embedding and metadata attention mechanism that enables them to better classify benign, phishing, malware, and deface URLs.

The model demonstrated a high level 98% accuracy, which is higher than the classical ML and DL baselines. Also, the authors involved SHAP and LIME explainability in trying to interpret the decision to make a prediction. Nevertheless, the research notes that it is difficult to cope with unseen zero-day attacks, and it can take a lot of power to run massive transformer models, which may limit real-time use.

In addition to detection models, threat intelligence extraction,n and streaming architecture, the absence of open-source end-to-end CTI platforms has also been suggested. Balasubramanian et al. [12] developed a real-time cyber threat intelligence platform called TSTEM that is based on cloud computing technology. The platform gathers actionable intelligence in microservices, Kafka, ELK, Scrapy, Tweepy, Terraform, and NLP-based, multi-stage classification in an autonomous manner. Besides the development of models and platforms, survey studies have also tried to bring together the advancements made in the field and find the gaps in malicious URL detection. A complex modality-based taxonomy is introduced by Tian et al. [13] and divides detection methods based on the lexical features of URLs, the content of the HTML pages, and visual modalities. In their work, they can be seen to be filling gaps in earlier surveys by identifying the lack of transformer and LLM-based defenses and by keeping public datasets and open-source repositories. However, their structure relies on publicly accessible resources, which can be obsolete as cyber threats develop rapidly.

Even though transformers are taking over the recent literature, ensemble machine learning techniques are still competitive, particularly for building lightweight security systems. Omolara and Alawida [14] presented DaE2, which is a generalized and effective ensemble system combining AdaBoost, Bagging, Stacking, and Voting. They show that they can detect with high accuracy of 80%, and AdaBoost with 98.5% accuracy, as well as ensemble variants, give high F1 scores(0976-0.980). Frameworks that incorporate a mixture of ML, DL, and optimization methods have also been of recent interest. An optimization-based malicious URL detection benchmarking framework proposed by Turk and Kilicaslan [15] uses ELECTRA-based transformers with LightGBM and CNN+LSTM architecture models. Regardless of such good performance, the framework is computationally expensive and highly sensitive to the dataset, and has little flexibility over adversarially manipulated unseen URLs.

On the same note, Qi et al. [16] performed a comparative analysis of classical NLP-based machine learning models and transformer-based models. The authors have validated that transformers are a lot better than conventional ML processes using TF-IDF and CountVectorizer preprocessing and Naive Bayes, Logistic Regression, FastText, and BERT. Their work has provided some significant materials of benchmarking evidence, but concerns regarding generalization and real-time deployment are yet to be dealt with.

The lightweight architectures have been considered to minimize the computational complexity of the algorithm and maintain transformer-level performance. Kibriya et al. [17] suggested a lightweight deep learning model that adds BERT-based URL embeddings and LSTM-GRU layers in this model. It is novel to combine lightweight transformer embeddings and explainability using LIME. The framework, however, is still dataset-specific, and it has difficulties with the invisible zero-day or adversarial URLs.

Explainable AI has also become a necessity as a security analyst needs to have a transparent decision-making process. Wang [18] was an explainable machine learning study that compared random forest, decision tree, and Logistic Regression. Random Forest obtained the most favorable compromise in both interpretability and accuracy. The authors also highlight the significance of interpretable detection systems, but the method is prone to limitations on the size of the dataset and is not completely adaptable to zero-day.

The use of ensemble feature-engineering frameworks still serves to identify various types of cyberattacks. Mohanty and Acharya [19] suggested a hybrid malicious URL detector scheme that comprises lexical and network features extraction and filter- wrapper features selections. The feature reduction was optimized to enhance their classification accuracy in phishing, spam, and malware attacks. But it is also limited by the inability to be scaled to the extent of automated transformer-based detection techniques, and resistance to undetectable adversarial URLs is low.

Lastly, studies on ransomware detection show that the trends of hybrid ensemble learning and explainability are similar. Singh et al. [20] suggested a hybrid model consisting of a Decision Tree, ANN, and XGBoost with PCA-selected static and dynamic ransomware features. Their model was 99.87% accurate and included SHAP interpretability. Although we have good performance, the technique still relies on benchmark data sets, and it is challenged by the inability to detect evolving zero-day ransomware variants in real-time.

Proposed Methodology

A. Dataset Description

Experiments in this study are conducted on the Malicious URLs Dataset, a publicly available benchmark dataset obtained from Kaggle. This dataset is designed for the research of malicious URL detection and web-based threat classification. It consists of 651,191 URL instances, and each instance holds a full-string URL and a class label that determines the security level of each URL. The dataset has four exclusive categories: benign, phishing, defacement, and malware URLs. Benign refers to URL types that are authentic and safe, and phishing refers to those URLs that are crafted in a way that they can collect user information. Defacement URLs are associated with malware attacks, where attackers change legitimate online content, and malware URLs are generated to deliver malware. The actual class distributions consist of 428,103 Benign URLs, 96,457 Defacement URLs, 94,111 Phishing URLs, and 32,520 Malware URLs. All are provided as raw text format, which means they are not processed, nor are any features extracted. The characteristics of the raw text are helpful because they include the lexical structure of all URLs, which include protocol, domain identification, subdomains, paths, special characters, and query parameters. These are good characteristics because they allow training and testing deep learning and traditional machine learning techniques, such as Logistic Regression and Support Vector Machines, as well as advanced hybrid deep learning models, because of their large size and diverse characteristics.

B. Data Preprocessing

Malicious URL detection requires data preparation, as raw URLs are unstructured, highly varied, and often contain noisy patterns that are not immediately usable by deep learning or machine learning models. The purpose of preprocessing in this work was to make the dataset scalable, lexical, suitable for statistical feature learning, and mathematically interpretable.

- Label Encoding of URL Classes: The dataset comprises four URL categories: phishing, malware, defacement, and innocuous. Label encoding was used to convert categorical classes to integers, as classification methods require numeric target labels.
  Let that the class variable is:
  $$y \in \{\text{benign, defacement, malware, phishing}\}$$
  The encoded label is expressed as:
  $$\hat{y} = f(y)$$
  where, $f(y)$ maps every class into an integer label:
  $$\{\text{benign} = 0, \text{defacement} = 1, \text{malware} = 2, \text{phishing} = 3\}$$
  This change maintains class identity while facilitating multi-class learning.
- URL Cleaning and Handling Missing Values: Occasionally, URLs may have faulty or empty entries. Consequently, before text transformation, missing values were substituted with an empty string:
  $$u_i = \begin{cases} u_i, & if\ valid \\ "", & otherwise \end{cases}$$
  In addition to ensuring consistent input formatting, this avoids model failure during feature extraction.
- Handcrafted Feature Extraction: A collection of lightweight lexical features was derived to capture the structural aspects of malicious URLs. These characteristics, which include length, digit ratio, special character frequency, and the presence of suspicious keywords, are quantifiable aspects of URL composition.
  The length of a given URL u is calculated as follows:

$$L(u) = |u|$$

Likewise, the quantity of special characters, like dots or hyphens, is determined as follows:

$$C_{dot}(u) = \sum_{j=1}^{|u|} \mathbb{I}(u_j = ".")$$

$$C_{hyphen}(u) = \sum_{j=1}^{|u|} \mathbb{I}(u_j = "-")$$

Additionally, a binary feature was used to warn URLs that contained dubious phrases like login, secure, or verify:

$$S(u) = \begin{cases} 1, & if\ suspicious\ word\ exists \\ 0, & otherwise \end{cases}$$

- TF-IDF Vectorization of URL Text: Character-level TF-IDF was used with n-grams of length 3 to 5 since URLs include sequential character patterns instead of natural language words. The n-gram t in URL u has the following Term Frequency (TF):

$$TF(t, u) = \frac{n(t,u)}{\sum_k n(k,u)}$$

This is how the Inverse Document Frequency (IDF) is calculated:

$$IDF(t) = \log\left(\frac{N}{1+df(t)}\right)$$

where, N is the total number of URLs, and df (t) is the number of URLs contain n-gram t. Consequently, the TF-IDF weight is:

$$TFIDF(t, u) = TF(t, u) \times IDF(t)$$

Malicious character sequences, such as phishing token structures and domain obfuscations, are successfully captured by this representation.

- Feature Selection Using Random Forest Importance: To reduce dimensionality and exact redundant n-grams, Random Forest-based feature selection was applied. Let the TF-IDF feature vector be:

$$X = [x_1, x_2, \ldots\ldots, x_d]$$

The mean importance threshold was used to choose a subset of informative features:

$$X' = \{x_j \mid I(x_j) \geq \mu I\}$$

where, $I(x_j)$ is the importance score from the Random Forest model, and μI is the mean importance across all features.

- Combined Feature Representation: The final input vector was created by concatenating the chosen TF-IDF features with manually created lexical characteristics:

$$X_{final} = [X'_{tfidf} || X_{lex}]$$

where, X'_{tfidf} represents the optimized statistical text features, and X_{lex} is the extracted structural URL attributes

- Train-Test Splitting: Using stratified sampling, the dataset was split into training and testing sets to guarantee an objective assessment:

$$D = D_{train} \cup D_{test}$$

For multi-class malicious URL identification, stratification maintained the original class distribution across both groups.

C. Proposed Model

This study presents a hybrid learning approach that combines the gradient-boosted classification and deep feature extraction to achieve precise, scalable identification of malicious URLs. The presented architecture consists of an XGBoost classifier for reliable multi-class threat recognition, a Convolutional Neural Network (CNN) for learning local lexical patterns, and a Bidirectional Long Short-Term Memory (BiLSTM) network for sequential dependency modeling. Malicious URLs usually contain both localized obfuscation patterns and long-range sequential structures, which inspired this concept. Figure1 depicts the proposed model architecture. Table 1 presents the hyperparameter configuration for the proposed model.

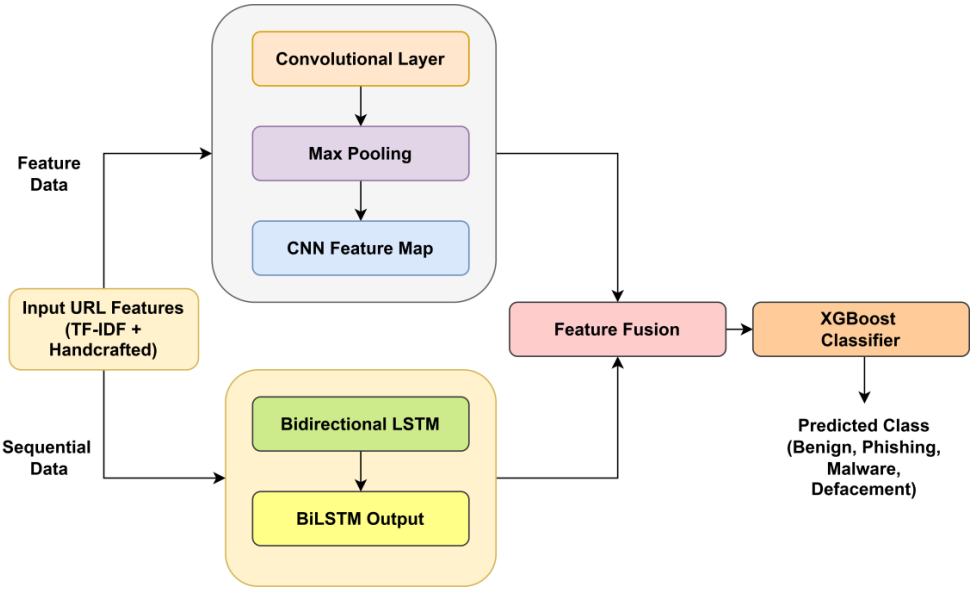


Figure 1: Graphical representation of the proposed model architecture

- Input Representation: Here, a URL instance u_i, first it is transformed into a numerical feature sequence employing the combined feature extraction. The resulting input vector is represented as:

$$X_i \in \mathbb{R}^d$$

where, following TF-IDF selection and handmade feature concatenation, d represents the overall feature dimension. A sequential format appropriate for neural processing is created from the input:

$$X_i = [x_1, x_2, \dots\dots, x_T]$$

where the effective sequence length is denoted by T.

- CNN-Based Local Feature Extraction: To identify discriminative local patterns in URL structures, such as recurring tokens, dubious character pairings, and substrings associated with phishing, the CNN component is utilized. For a convolution filter, the extracted feature map is calculated as:

$$c_k(t) = f(W_k . X_{t:t+h-1} + b_k)$$

where, h is the kernel size, b_k is the bias term.
The convolutional output becomes:

$$C = [c_1, c_2, \dots\dots, c_T]$$

where, m denotes the number of filters.
The most instructive activations are kept by applying a max-pooling operation:

$$p_k = \max(c_k)$$

The pooled representation forms a compact feature vector:

$$P = [p_1, p_2, \dots\dots, p_m]$$

This CNN step effectively highlights high-risk URL fragments that are commonly found in fraudulent domains.

- BiLSTM-Based Sequential Dependency Modeling: Malicious URLs exhibit sequential dependencies throughout the entire URL string, even though CNNs capture only local patterns. In order to model both forward and backward contextual information, a BiLSTM network is shown.
The following provides the forward LSTM hidden state:

$$\overrightarrow{h_t} = LSTM\left(x_t, \overrightarrow{h_{t-1}}\right)$$

Similarly, the backward hidden state is calculated as:

$$\overleftarrow{h_t} = LSTM\ (x_t\ , \overleftarrow{h_{t+1}})$$

Both directions are concatenated to produce the BiLSTM output:

$$h_t = [\overrightarrow{h_t}\ ||\ \overleftarrow{h_t}]$$

The final sequential embedding is expressed as:

$$\mathrm{H}\ [h_1, h_2, \ldots\ldots, h_T]$$

This enables the model to identify persistently harmful patterns, such as domain spoofing, in conjunction with misleading path topologies.

- Deep Feature Fusion Layer: A combined deep representation is created by fusing the retrieved CNN pooled vector P with the BiLSTM contextual embedding H:

$$F_{deep} = [P\ ||\ H]$$

A layer that is fully connected is traversed by the fused representation:

$$\mathrm{Z} = \sigma\ (W_f F_{deep} + b_f)$$

where, W_f and b_f are learnable parameters. The resulting vector Z acts as a high-level feature description of the URL instance.

- XGBoost-Based Final Classification: Due to its improved generalization and robustness in structured threat detection tasks, the proposed system uses Extreme Gradient Boosting (XGBoost) for final multi-class classification rather than a conventional softmax layer. According to the classifier, the URL label will be:

$$\hat{y}_i = \underset{c\ \in\ C}{\arg\ max}\ \phi_c(Z)$$

The XGBoost objective minimizes the following regularized loss:

$$\mathscr{L} = \sum_{i=1}^{N} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k)$$

The regularization is expressed as:

$$\Omega\ (\mathrm{f}) = \gamma \mathrm{T} + \frac{1}{2}\lambda ||\mathrm{w}||^2$$

where, T is the number of leaf nodes, w is the leaf weight vector, and γ and λ are penalty parameters

The complete prediction pipeline can be summarized as:

$$u_i \longrightarrow X_i \longrightarrow \mathrm{CNN}\ (X_i) \longrightarrow \mathrm{BiLSTM}\ (X_i) \longrightarrow \mathrm{Fusion} \longrightarrow \mathrm{XGBoost}\ (\mathrm{Z}) \longrightarrow \hat{y}_i$$

Table 1: Hyperparameter Settings of the Proposed Hybrid CNN–BiLSTM–XGBoost Model

Component	Hyperparameter	Value / Setting
Input Layer	TF-IDF Features	5000 (selected → 306)
	Handcrafted Features	15
	Combined Feature Dimension	321
CNN Module	Number of Filters	128
	Kernel Size	3, 4, 5
	Activation Function	ReLU
	Pooling Type	Max Pooling
	Dropout Rate	0.3
BiLSTM Module	Hidden Units	128
	Bidirectional	Yes
	Sequence Output	Last Hidden State
	Dropout Rate	0.3
Fusion Layer	Fusion Strategy	Concatenation

	Fully Connected Units	256
XGBoost Classifier	Number of Trees (Estimators)	300
	Maximum Tree Depth	6
	Learning Rate	0.1
	Subsample Ratio	0.8
	Column Sample Ratio	0.8
	Objective Function	Multi-class Softmax
	Regularization (L2)	1.0
Training Setup	Optimizer (Deep Module)	AdamW
	Batch Size	64
	Epochs	100
	Train/Test Split	80% / 20%
	Random Seed	42

Result & Discussion

In this section, a detailed evaluation of the proposed framework of using a hybrid model of CNN, BiLSTM, and XGBoost for carrying out malicious URL detection is presented. The performance of the proposed model is evaluated and compared with different existing algorithms such as Logistic Regression, Support Vector Machines, XGBoost, and different deep learning algorithms of CNN. For evaluating the performance of the proposed system, different evaluation metrics such as accuracy, precision, recall, and F1-score, as well as confusion matrix and ROC curve analysis, are utilized.

A. Experimental Setup

All experiments were performed on an Intel Core i5 CPU, 8 GB RAM, along with 256 GB SSD running 64-bit Windows 10. The implementation was achieved using Python 3.10, where TensorFlow/Keras handled deep learning, Scikit-learn was used for conventional models, while XGBoost was used to implement ensemble classification. Pandas and NumPy libraries were utilized for data preprocessing. A publicly available malicious URL dataset containing over 650,000 data points with four classes, namely, benign, phishing, defacement, and malware, was used. An 80-20 split was utilized to check the performance of the proposed model along with stratified sampling for splitting data into 80% training and 20% test sets. Character-level TF-IDF features have been extracted, reducing them to 306 dimensions, while other features include 15 dimensional URL structure features, leading to a 321-D feature vector. The proposed method uses the CNN-BiLSTM structure to extract features followed by XGBoost as the final classifier. Evaluation was performed comparing with Logistic Regression, SVM, XGBoost, and CNN. Evaluation metrics used include accuracy, precision, recall, F1 score, and so on.

B. Performance Evaluation Analysis

As shown in Table 2 below, how well our proposed hybrid framework stacks up competitive baseline models of various machine learning and deep learning types is presented. The verdict here is already evident - Hybrid CNN + BiLSTM + XGBoost consistently outperforms all competing ones and thus proves its utility in identifying malicious URLs on a large scale.

Table 2: Performance Analysis of the Proposed and Baseline Models

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression (LR)	0.78	0.62	0.65	0.63
Support Vector Machine (SVM)	0.80	0.76	0.80	0.77
XGBoost (Baseline)	0.88	0.86	0.85	0.85
CNN (Baseline Deep Model)	0.91	0.90	0.89	0.89
Proposed Hybrid CNN + BiLSTM + XGBoost	**0.96**	**0.97**	**0.93**	**0.95**

LR is at the bottom of the pack with an accuracy of 0.78 and an F1 score of 0.63. This is as it should be as LR is a linear classifier with trouble fully understanding the non-linear types of patterns and obfuscation schemes contained in the majority of mal-URLs. SVM indicates a low but clear improvement over LR with an accuracy of 0.80 and an F1-score of 0.77. Although SVM performs better with unlinear boundaries than LR, it equally depends on the separability of human-designed features and does not learn well from in-depth sequential patterns in URLs. The baseline XGBoost model additionally increases the performance for detection, which reaches 0.88 accuracy and 0.85 F1-score. This demonstrates the effectiveness of the gradient boosting approach for handling structural relationships between features and strengthening classification priors. However, XGBoost cannot sufficiently leverage the sequential nature of the URL text data without performing deep feature extraction. The CNN baseline achieves an accuracy of 0.91 and an F1 score of 0.89, which outperforms all traditional ML baselines. This shows the strength of convolutional networks in using local patterns in characters, i.e., suspicious substrings and domain manipulations. However, convolutional networks look for local patterns, not long-range dependencies as found in URLs.

The proposed hybrid model beats all these models with an accuracy of 0.96, precision of 0.97, recall of 0.93, and an F1-score of 0.95. This high level of performance is obtained in the proposed model since it combines CNN for the identification of local lexical patterns, uses its ability to process the entire URL for sequential dependencies in the BiLSTM model, and finally uses XGBoost as the classifier for enhanced generalizability.

Figure 2 shows the class-wise performance of the proposed model, which indicates the performance of the proposed model on all four categories of URLs. It clearly indicates that the proposed model performs well in terms of precision, recall, and F1-scores, which confirms the effectiveness of the model for distinguishing between benign and malicious URLs. In addition, the performance of the model on benign and defacement classes is highly stable, with precision and recall of 0.97 and 0.99, respectively, resulting in an excellent F1-score of 0.98.

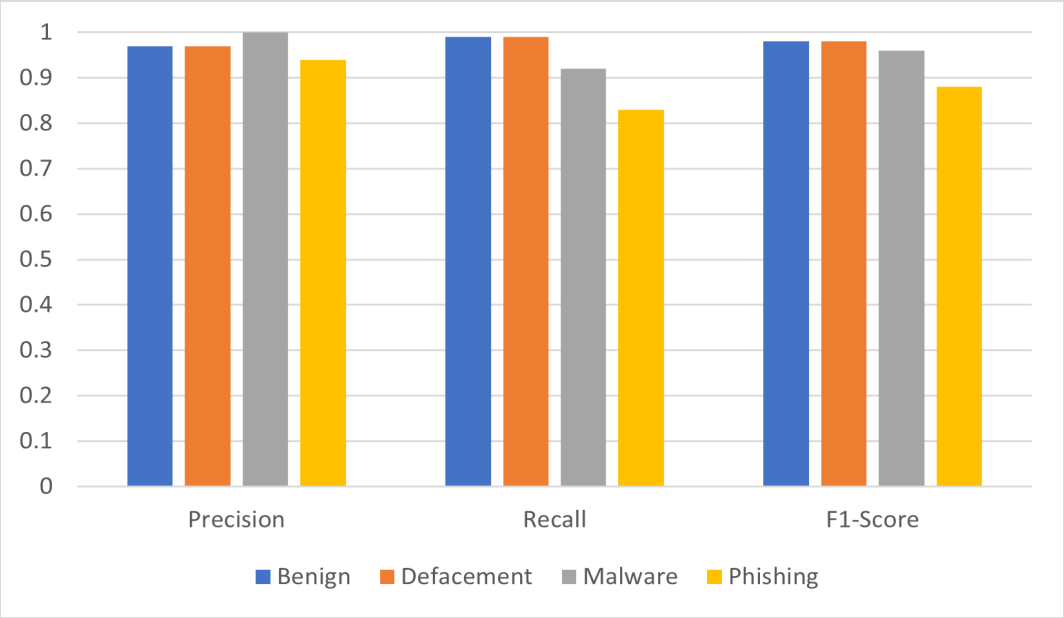


Figure 2: Proposed Hybrid Model Class-wise Performance

The malware category also shows an excellent precision of 1.00, which confirms the effectiveness of the model in avoiding false malware predictions. However, there is some decline in the recall of 0.92, which results in an F1-score of 0.96, indicating that the model successfully detects most of the malware URLs. In addition, the low F1-score of 0.88 for the phishing category indicates low recall of 0.83, which may be due to the complexity of the problem, as most of the phishing URLs resemble legitimate URLs and use complex techniques for evading detection.

C. Confusion Matrix Analysis

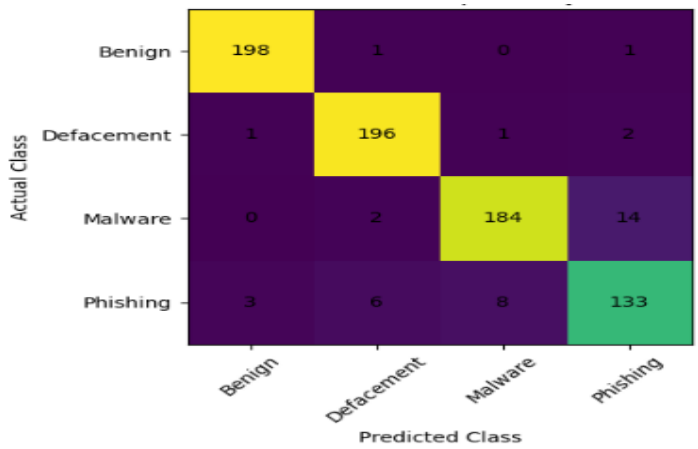


Figure 3: Confusion matrix analysis of the proposed model

Figure 3 presents the confusion matrix of our hybrid model for the multiclass malicious URL classification, including categories such as Benign, Defacement, Malware, and Phishing. One can easily observe that the matrix is diagonally dominant, which means that the model rightly classifies most examples and a few cross-category errors.

The benign one can be identified by its consistent results: 198 of the 200 URLs were appropriately classified. The two misclassifications concern the harmless URLs that were incorrectly classified as phishing and defacement, respectively. This demonstrates the model's ability to distinguish between harmful and non-malicious URLs, which is an important aspect of lowering false alarms in practical applications. Defacement is also very reliable, as 196 predictions were correct. The few errors are towards phishing and malware, indicating some overlap in features used among the web-based attack patterns. The small off-diagonal values however indicate high recall and precision for this class. In the Malware category, 184 samples are correctly identified, with a few miscategorized as phishing. This is to be expected, since many of the malware-hosting URLs share traits with phishing URLs. Yet, there are very few false positives, underlining the model's good discriminatory power and the perfect precision that has been reported for malware detection.

Phishing enjoys relatively more misclassifications, sometimes being classified as malware or defacement. This accounts for the lower recall for phishing URLs. The overlap is a reflection of the crafty and evolving nature of phishing, which often imitates either legitimate or malware-driven web activity. Nevertheless, the majority of phishing cases are correctly detected, showing the strength of the model against socially engineered attacks.

D. ROC Curve Analysis

Figure 4 illustrates ROC curves of our proposed system in multiclass malicious URL classification using a one-vs-rest mechanism. Note that all the classes are plotted tightly around the upper left region of the graph, indicating a good balance of true and false positive rates.

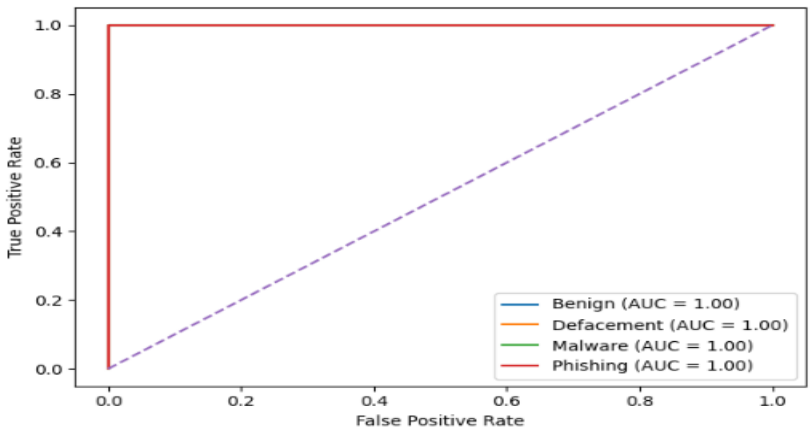


Figure 4: ROC curve analysis of the proposed model

As depicted, Benign and Defacement behave almost like perfect data on the ideal ROC curve, which reiterates the effectiveness of separating legitimate and defaced URLs from each other and the rest while generating very few false alarms. It further reiterates and reconfirms the high precision and recall results discussed above. Malware also excelled in having high separability with a high true positive rate even with a low false positive rate. It did so in a manner that aligned with the confusion matrix because it only produced a low number of false positives as a result of malware instances.

Phishing is strong, but increases slightly more slowly than the other ones. This is in line with the lower recall levels for the detection of phishing. This can be attributed to the dynamic and evasive nature of phishing URLs that tend to mimic malware or defacement patterns. Nevertheless, the model has strong detection capacity, as highlighted by the overall ROC curve.

E. Comparative Analysis

Table 2 depicts a comparative analysis of the performance obtained by the proposed model with respect to different advanced malicious URL detection techniques introduced in recent years. It can be seen from Table 2 that Tian et al. introduced a CNN–GRU-based model, with an accuracy of 91.8%, an ROC value of 94.2%, and used the CNN architecture. Though the model is seen to be performing well, its performance is largely based on a binary classification mechanism, which performs poorly when used with large-scale datasets [21].

Kibriya et al. proposed a lightweight CNN-based model in their research [22], which achieved high accuracy at 93.4% and an ROC score at 95.1% using a Malicious URLs Dataset. Even as a lightweight model, which has minimal computational cost, it fails to effectively learn long-range sequential relationships, which are inherent in complex URLs and have a major impact on the classification response for phishing and malware attacks.

Conversely, in the work by Turk et al. [23], an XGBoost-based technique with handcrafted lexical and statistical features was employed and resulted in an accuracy of 94.1% and an ROC of 96.3%. Although this approach outperforms other techniques based purely on deep learning, over-reliance on handcrafted features brings about limitations in dealing with changing and new forms of malicious URL patterns.

Table 3: Comparative analysis with proposed and existing models

Reference	Dataset	Model	ROC	Accuracy	Limitation
[22]	ISCX-URL Dataset	CNN + GRU	94.2%	91.8%	Focused mainly on binary classification; limited performance on large imbalanced datasets
[23]	Malicious URLs Dataset	Lightweight CNN	95.1%	93.4%	Uses shallow architecture; limited ability to capture long-range URL dependencies
[24]	Mixed Web URL Dataset	XGBoost + Handcrafted Features	96.3%	94.1%	Heavy dependence on handcrafted features; weak generalization to unseen URLs
Proposed Hybrid Model	Malicious URLs Dataset	Hybrid CNN–BiLSTM–XGBoost	100%	96%	—

However, the proposed framework of CNN, BiLSTM, and XGBoost outperformed all the techniques that were compared, with an accuracy of 96%. Additionally, the ROC of this model was 100%, indicating its advantage over others on the same dataset. This is largely due to the capabilities of all the techniques that are being combined in this model, as CNN works well with local lexical patterns, BiLSTM also works well, but this model provides a nonlinear decision boundary due to XGBoost, which is best suited for classification when there are multiple classes. Thus, this model outperformed all other techniques as it does not rely only on the classification done by a series of rules, as well as a softmax classifier.

Conclusion

This paper has described a hybrid concept of deep learning as well as an ensemble method for the detection of malicious URLs created in large volumes, utilizing CNN BiLSTM, and XGBoost as a classifier. Overall, the proposed model effectively learns the representations that are discriminatory from raw URL data and also handles issues that are faced by both traditional machine learning and deep learning individually. A series of extensive experiments has been conducted using a large and realistic multi-class dataset, showing that the approach described in this paper consistently achieves better performance compared to all other baseline methods, reaching a strong accuracy of 96%, while also ensuring robustness within each class and low occurrences of false positives. Overall, the experimental outcomes validate that utilizing a combination of deep sequential features as well as gradient-boosted classifiers becomes an important and reliable solution in the context of web-based threats.

References

1. Y. Tian, Y. Yu, J. Sun, and Y. Wang, “From past to present: A survey of malicious URL detection techniques, datasets and code repositories,” arXiv, Apr. 2025.

2. “A comprehensive review of malicious URL detection using deep learning techniques,” 2025 Int. Conf. ISNCC, Nov. 2025.

3. H. Kibriya et al., “Lightweight malicious URL detection using deep learning with URL embeddings,” Sci. Rep., 2025.

4. M. Khaldi, “Hyperparameter optimization for malicious URL detection via BiGRU and attention,” Informatica, 2025.

5. A. Cohen, “Client-side zero-shot LLM inference for comprehensive in-browser URL analysis,” arXiv, 2025.

6. T. Mahmud et al., “A machine learning-based framework for malicious URL detection in cybersecurity,” 2025 IEEE ICICT.

7. F. Turk, “Malicious URL detection with advanced machine learning,” Appl. Sci., 2025.

8. N. Q. Do, “Detection of malicious URLs using temporal convolutional network with self-attention,” Elsevier, 2025.

9. Recent comparative work on hybrid ML and DL approaches in malicious URL detection, IJRASET, 2025.

10. Emerging trends in explainable detection techniques for malicious URLs, ACM Trans., 2025.

11. L. Chen and L. Meng, “Metadata driven malicious URL detection using RoBERTa large and multi source network threat intelligence,” Scientific Reports, Nature Publishing Group, 2026.

12. P. Balasubramanian et al., “A cognitive platform for collecting cyber threat intelligence and real-time detection using cloud computing,” Decision Analytics Journal, vol. 14, p. 100545, 2025.

13. Y. Tian, Y. Yu, J. Sun, and Y. Wang, “From Past to Present: A Survey of Malicious URL Detection Techniques, Datasets and Code Repositories,” arXiv preprint arXiv:2504.16449, 2025.

14. A. E. Omolara and M. Alawida, “DaE2: Unmasking malicious URLs by leveraging diverse and efficient ensemble machine learning for online security,” Computers & Security, vol. 148, p. 104170, 2025.

15. F. Türk and M. Kılıçaslan, “Malicious URL detection with advanced machine learning and optimization-supported deep learning models,” Applied Sciences, vol. 15, no. 18, p. 10090, 2025.

16. S. Qi, A. R. Sangi, T. Sun, B. Niu, and Y. Huang, “Malicious URL Detection Using NLP: Comparing Classical and Transformer-Based Models,” in Proc. IEEE 6th Int. Conf. Pattern Recognition and Machine Learning (PRML), 2025, pp. 452–456.

17. H. Kibriya et al., “Lightweight malicious URL detection using deep learning and large language models,” Scientific Reports, Nature Publishing Group, 2025.

18. B. Wang, “Malicious URL detection with explainable machine learning techniques,” in Proc. 2nd Int. Conf. Informatics Education and Computer Technology Applications, 2025, pp. 293–299.

19. S. Mohanty and A. A. Acharya, “Detection of cyber attacks from malicious URLs using ensemble machine learning techniques,” in Intelligent Technologies: Concepts, Applications, and Future Directions, vol. 4. Springer, 2025, pp. 55–87.

20. S. Singh, T. Khanna, and D. K. Verma, “A hybrid ensemble model for ransomware detection using feature engineering and deep learning,” International Journal of Information Technology, vol. 17, no. 8, pp. 5095–5104, 2025.

21. Y. Tian, Y. Yu, J. Sun, and Y. Wang, “From past to present: A comprehensive survey on malicious URL detection techniques, datasets, and evaluation,” arXiv preprint arXiv:2504.16449, 2025.

22. H. Kibriya, M. R. Amin, and S. Islam, “Lightweight malicious URL detection using deep learning with large-scale web data,” Scientific Reports, vol. 15, no. 1, pp. 1–15, 2025