Information Sciences and Technological Innovations

www.isti.reapress.com

Inf. Sci. Technol. Innov. Vol. 2, No. 4 (2026) 288–302.

Paper Type: Original Article

Detecting Phishing Websites from Character-Level URL Sequences: A Bidirectional LSTM Approach with Class-Imbalance Mitigation

Ilobekemen Perpetual Oladoja[1,*], Chukwuemeka C. Ugwu[1], Anuoluwapo P. Ajibade[1], Tolulope A. Ugwu[1], Olugbenga Ayomide Madamidola[1]

[1] The Federal University of Technology, Akure, Nigeria; ipoladoja@futa.edu.ng; ccugwu@futa.edu.ng; ajibadepacsc2019@futa.edu.ng; taugwu@futa.edu.ng; oamadamidola@futa.edu.ng.

Citation:

Received: 19 March 2026 *Revised: 12 June 2026* *Accepted: 01 August 2026*	Oladoja, I. P., Ugwu, Ch. C., Ajibade, A. P., Ugwu, T. A., & Madamidola, O. A. (2026). Detecting Phishing websites from character-level URL sequences: A bidirectional LSTM approach with class-imbalance mitigation. *Information sciences and technological innovations,* 2(4), 288-302.

Abstract

Phishing websites remain a significant cybersecurity threat, exploiting deceptive Uniform Resource Locator (URLs) to trick users into revealing sensitive information. Existing detection approaches, particularly blacklist-based and traditional Machine Learning (ML) methods, struggle to generalize to newly crafted Phishing URLs and often fail to capture the sequential patterns inherent in URL structures. To address these limitations, this study proposes a Deep Learning (DL)–based Phishing website detection model using a Bidirectional Long Short-Term Memory (Bi-LSTM) network that operates on character-level URL representations. A large-scale dataset comprising 450,176 URLs (Phishing and legitimate) sourced from the Mendeley repository was employed. The URLs were preprocessed through normalization, character-level tokenization, and embedding, while class imbalance was mitigated using the Synthetic Minority Over-Sampling Technique (SMOTE). The proposed Bi-LSTM model was trained and evaluated using standard performance metrics, including accuracy, precision, recall, F1-score, and Receiver Operating Characteristic (ROC)–Area Under the Curve (AUC), and was compared against a baseline Convolutional Neural Network (CNN). Experimental results demonstrate that the Bi-LSTM model achieves an accuracy of 97.81%, precision of 94.75%, recall of 95.97%, and an F1-score of 95.36%, outperforming the CNN baseline in terms of accuracy and precision while maintaining comparable recall.

Keywords: Deep learning, Phishing uniform resource locators, Legitimate uniform resource locators, Detection model, Machine learning, Bidirectional long short-term memory, Convolutional neural network.

1 | Introduction

Phishing remains one of the most prevalent and damaging cybersecurity threats, exploiting deceptive techniques to trick users into disclosing sensitive information such as usernames, passwords, financial details,

Corresponding Author: ipoladoja@futa.edu.ng

https://doi.org/10.48314/isti.vi.55

and personal data [1]. These attacks typically involve fraudulent websites or communications that impersonate trusted entities, thereby deceiving users into interacting with malicious platforms that closely resemble legitimate services [1], [2]. As internet usage continues to expand across e-commerce, online banking, and digital communication platforms, Phishing attacks have become increasingly frequent, sophisticated, and impactful, resulting in financial losses, identity theft, data breaches, and erosion of trust in online systems [3].

Phishing attacks take several forms, including email Phishing, spear Phishing, and website Phishing, as well as smishing and vishing via SMS messages and voice calls [4]. Among these, website Phishing remains particularly dangerous, as attackers create malicious Uniform Resource Locator (URLs) and webpages that visually and structurally mimic legitimate sites to deceive users. This study focuses specifically on website Phishing, where malicious URLs are analyzed to distinguish Phishing websites from legitimate ones.

Various methods have been proposed to detect Phishing websites. Early detection methods were blacklist-based and heuristic-based. Blacklist-based methods rely on repositories of known Phishing URLs, making them ineffective against newly generated or zero-day Phishing attacks. Heuristic-based methods depend on handcrafted rules and expert knowledge, which limits their adaptability to evolving Phishing strategies [3], [4]. Consequently, these approaches are less effective in real-world environments where Phishing URLs are rapidly created and modified to evade detection.

However, Machine Learning (ML) techniques have been introduced for Phishing website detection to address the limitations of early detection methods. ML-based approaches apply statistical patterns and engineered features extracted from URLs, domain information, or webpage characteristics, using classifiers such as Support Vector Machines (SVMs), Random Forests (RFs), and Naïve Bayes (NBs) [5], [6]. While these methods have demonstrated improved performance over rule-based systems, their effectiveness often declines when applied to large-scale datasets, highly imbalanced class distributions, or URLs that employ complex obfuscation techniques. Moreover, the reliance on extensive feature engineering makes many ML-based solutions less flexible and less effective against novel Phishing patterns [6], [7].

Recent advances in Deep Learning (DL) have shifted Phishing detection research toward models that can automatically learn discriminative representations directly from raw data. Researchers have explored DL methods such as Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and hybrid models to capture structural and sequential characteristics of Phishing URLs and webpages [7–10]. However, many existing DL models still face notable challenges, including shallow sequential modeling, insufficient capture of long-range dependencies in URL character sequences, vulnerability to severe class imbalance, and increased architectural complexity that can hinder scalability and practical deployment.

Bidirectional Long Short-Term Memory (Bi-LSTM) networks offer a promising solution by modeling sequential data in both forward and backward directions, enabling the capture of contextual dependencies across entire URL strings. This bidirectional modeling is particularly well suited to phishing URL detection, where meaningful patterns may appear at different positions within the URL. Despite their potential, Bi-LSTM-based Phishing detection models remain underexplored at scale, especially in scenarios involving large, real-world URL datasets and explicit strategies for handling class imbalance.

Motivated by these gaps, this study proposes a Phishing website detection model based on a Bi-LSTM network integrated with the Synthetic Minority Over-Sampling Technique (SMOTE) to effectively learn discriminative patterns from character-level URL sequences while addressing class imbalance in large-scale Phishing datasets, thereby improving detection accuracy, robustness, and generalization to previously unseen Phishing websites.

The rest of the study is partitioned as follows: Section 2 reviews related work, Section 3 describes the proposed methodology, Section 4 presents and discusses the experimental results, and Section 5 concludes the study.

2 | Related Works

Research on Phishing website detection has evolved significantly over the past decade, progressing from rule-based systems to advanced DL architectures.

Barik et al. [8] proposed an optimized DL framework for Phishing URL detection to improve accuracy while reducing reliance on manual feature engineering. The study introduces a newly constructed Phishing dataset aggregated from multiple sources. It employs TF-IDF for feature extraction, Variational Autoencoders (VAEs) for feature enhancement, and a CNN as the core classifier. Model performance is further improved through Enhanced Grid Search Optimization (EGSO) for hyperparameter tuning. The experimental results show that the proposed model outperformed the standard CNN, LSTM, and GRU architectures, achieving 99.44% accuracy, with precision, recall, and F1-score exceeding 99%. However, the approach is limited by its focus on text-based URL features.

The author in [9] explored DL techniques for Phishing detection, with particular emphasis on a hybrid CNN–BiLSTM architecture designed to improve robustness against sophisticated attacks such as URL obfuscation and domain impersonation. We compiled a dataset of 50,000 URLs from sources such as PhishTank, OpenPhish, and Common Crawl. Lexical, structural, and domain-based features were extracted and refined using recursive feature elimination. Several DL models CNN, BiLSTM, Transformer, and Deep Belief Networks (DBNs) were implemented and evaluated. The results show that the hybrid CNN–BiLSTM model performed best with 81% accuracy. However, there is a tendency toward high computational complexity, which might necessitate further optimization.

Alsaedi et al. [4] presented a Cyber Threat Intelligence (CTI)–based malicious URL detection model that integrates external intelligence sources to improve detection accuracy. The proposed framework combined traditional URL features with Google-based and Whois-based CTI features. These heterogeneous features are processed using TF-IDF and information gain, and classified through a two-stage ensemble consisting of multiple RF models whose outputs are fused using a Multilayer Perceptron. Experimental results show that the CTI-based ensemble achieves an accuracy of approximately 96.8% and significantly reduces false-positive rates compared to conventional ML-only models. However, the multi-stage ensemble architecture increases system complexity and may limit scalability.

In another study, Raja et al. [3] focused on detecting malicious domains using only the textual information contained in URLs, with the aim of reducing reliance on handcrafted features and webpage content analysis. The methodology involved preprocessing URLs to extract domain names, followed by feature representation using Count Vectorizer, TF-IDF, Hash Vectorizer, and word embeddings. The extracted features were passed to a range of ML and DL models and were evaluated on two benchmark datasets. Results showed that traditional ML models combined with TF-IDF features outperform the DL models considered, achieving up to 99.5% accuracy on Phishing-focused datasets. However, conventional ML models combined with TF-IDF incur increased computational and memory complexity due to high-dimensional sparse feature representations, which may limit training efficiency.

Aljofey et al. [10] developed a client-side Phishing website detection approach that combines URL- and HTML-based features to address the limitations of blocklist and third-party–dependent methods. The features extracted include URL character patterns, textual content, and hyperlink information. The study trained multiple ML models, with XGBoost achieving the best performance. However, the developed approach's training time is relatively long due to the high-dimensional vector generated by textual content features.

Murhej and Nallasivan [11] developed a Phishing detection system that uses visual, semantic, and structural webpage features. They rendered webpages into HTML and DOM trees, then extracted features like images, forms, and links. They processed visual features with EfficientNet and represented text features using FH BERT embeddings. The team optimized the selected features with an Entropy-based Chameleon Swarm

Algorithm and classified them using a SELU CRNN network. Their model achieved 98.42% accuracy, outperforming standard CNN, RNN, and DBN models. However, this high performance comes at the cost of significant computational and architectural complexity. The reliance on multiple DL models and pretrained networks increases resource requirements.

Fang et al. [12] introduced CMNet, which combines CNN and MetaFormer (Transformer). This model uses contrastive learning on multimodal website data by integrating text and structural features. CMNet achieved 96.3% accuracy by learning a detailed representation that highlights the subtle differences between phishing and legitimate sites.

Xie et al. [13] created a scalable model that employs a dual-branch Temporal Convolutional Network (TCN) with channel and spatial attention mechanisms. This model reached 97.66% accuracy on a newly collected Phishing dataset and outperformed other techniques.

Abid and Abid [14] presented an AI-based model focused on emails and URLs that uses BiLSTM networks. Their system reached an overall accuracy of 92.4%, showing that deep recurrent models can be effective for Phishing detection.

While existing Phishing detection approaches achieve high accuracy through optimization-heavy, ensemble, or multimodal architectures, their computational complexity and reliance on extensive feature engineering limit practical deployment. In contrast, BiLSTM-based models balance performance and efficiency by directly learning bidirectional sequential patterns from URLs, offering a lightweight, scalable, and deployable solution for real-world Phishing detection.

3 | Methodology

The proposed system comprises four main stages: data collection, data preprocessing, model training, and performance evaluation. As illustrated in *Fig. 1*, the dataset used in this study was obtained from the Mendeley repository and consists of two primary classes of website URLs: 104,438 Phishing URLs and 345,738 legitimate URLs. During preprocessing, raw URLs are transformed into a structured representation suitable for ML. This process involves removing special characters, tokenizing URLs into meaningful components, and mapping each token into a dense vector space using an embedding layer. The resulting token sequences then train a Bi-LSTM network. The Bi-LSTM architecture is selected due to its ability to capture long-term dependencies in both forward and backward directions, making it particularly well suited for modeling sequential data such as URLs. Each LSTM unit computes hidden states that collectively encode contextual information across the entire URL sequence. After training, the model is evaluated using standard classification metrics, including accuracy, precision, recall, and F1-score. These metrics provide a comprehensive assessment of the model's effectiveness in distinguishing phishing from legitimate URLs.

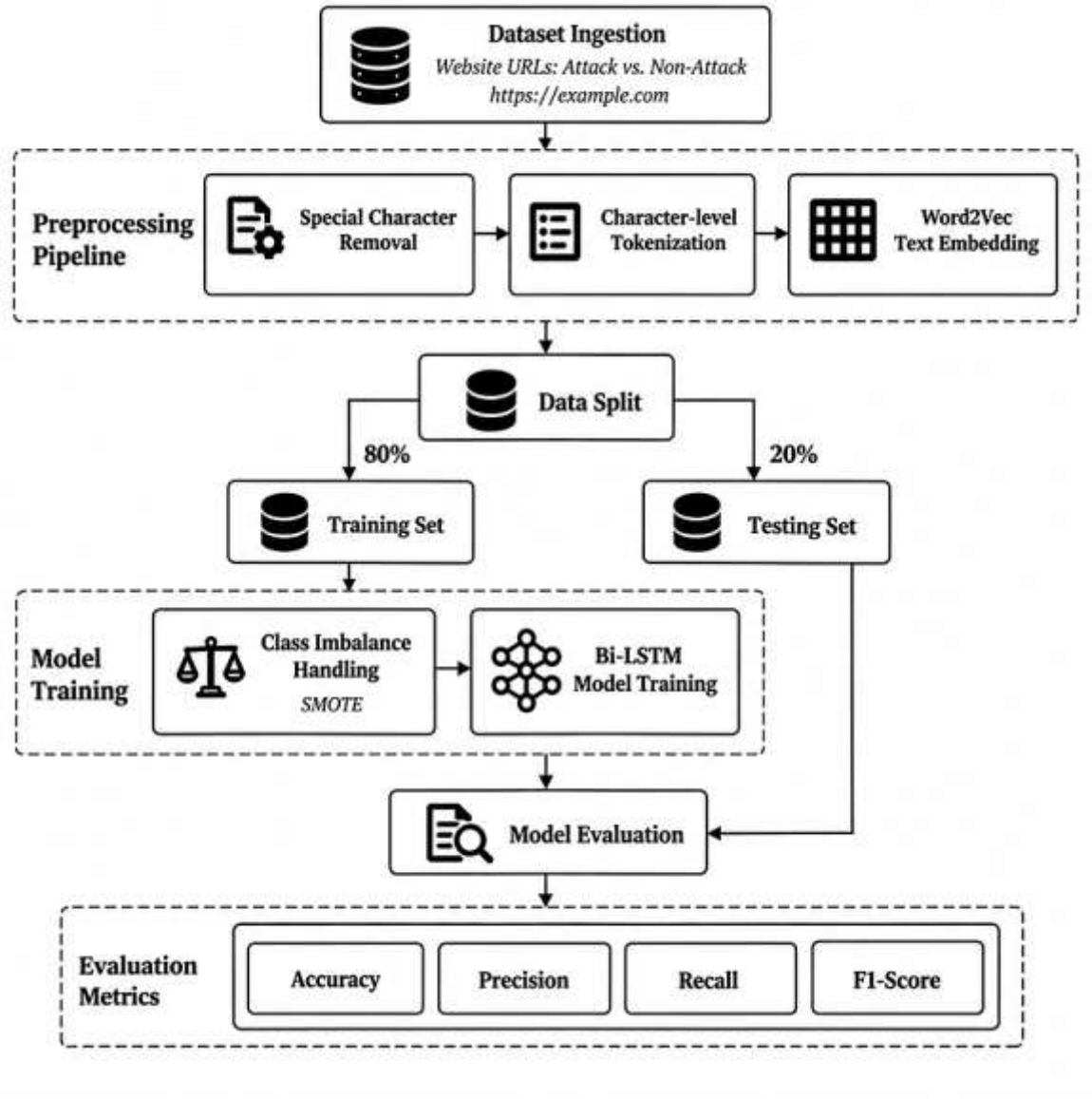


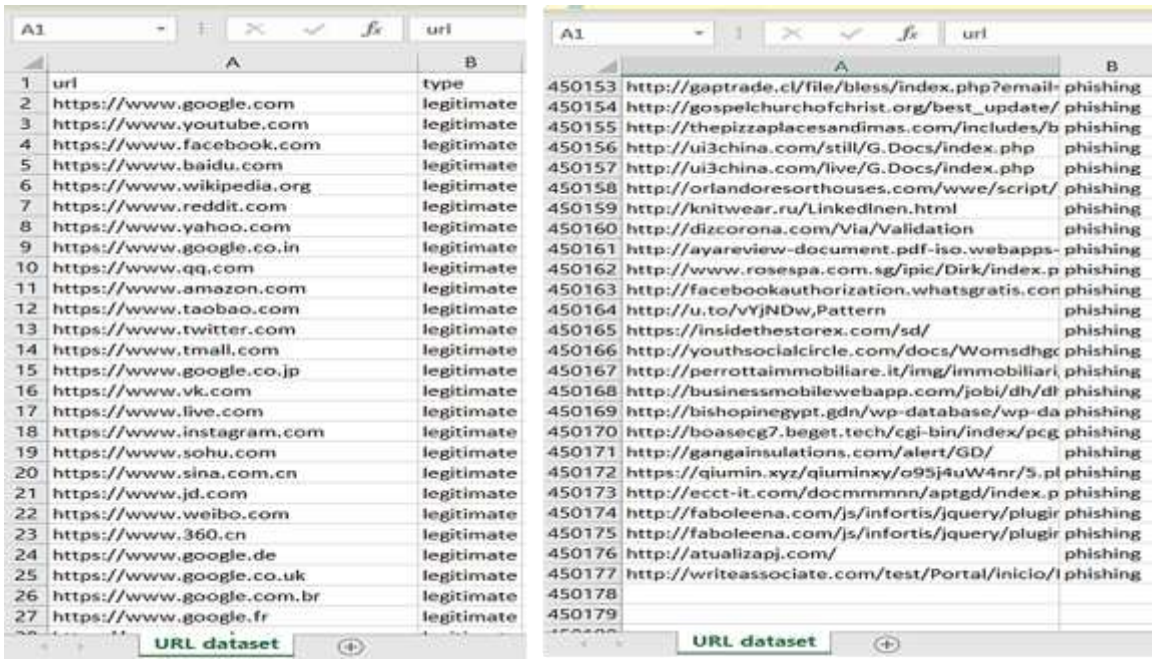
Fig. 1. System architecture.

3.1 | Data Acquisition and Description

The dataset used in this study was obtained from the Phishing URL Dataset published on 2 April 2024, curated by Kaitholikkal and Arthi [15]. The dataset was designed to support research on Phishing detection by providing a large and diverse collection of labeled URLs representing both malicious and benign web activities.

The dataset comprises 450,176 URLs collected from multiple reputable sources, including PhishTank, the Majestic Million, and other relevant platforms. Each URL is explicitly labeled as either phishing or legitimate, enabling supervised learning and rigorous evaluation of Phishing detection models. Of the total URLs, 104,438 are identified as phishing, reflecting malicious intent, while 345,738 URLs are classified as legitimate, representing non-malicious web traffic. This distribution mirrors real-world conditions, where legitimate websites significantly outnumber Phishing sites.

In addition to the combined dataset, the resource includes a Phishing-only subset containing 54,807 URLs, providing a focused benchmark for analyzing Phishing-specific patterns and behaviors. The inclusion of both a large mixed dataset and a Phishing-exclusive subset allows researchers to conduct flexible experiments, such as binary classification, imbalance analysis, and Phishing pattern exploration. A sample of the dataset is shown in *Fig. 2*.



	A	B
1	url	type
2	https://www.google.com	legitimate
3	https://www.youtube.com	legitimate
4	https://www.facebook.com	legitimate
5	https://www.baidu.com	legitimate
6	https://www.wikipedia.org	legitimate
7	https://www.reddit.com	legitimate
8	https://www.yahoo.com	legitimate
9	https://www.google.co.in	legitimate
10	https://www.qq.com	legitimate
11	https://www.amazon.com	legitimate
12	https://www.taobao.com	legitimate
13	https://www.twitter.com	legitimate
14	https://www.tmall.com	legitimate
15	https://www.google.co.jp	legitimate
16	https://www.vk.com	legitimate
17	https://www.live.com	legitimate
18	https://www.instagram.com	legitimate
19	https://www.sohu.com	legitimate
20	https://www.sina.com.cn	legitimate
21	https://www.jd.com	legitimate
22	https://www.weibo.com	legitimate
23	https://www.360.cn	legitimate
24	https://www.google.de	legitimate
25	https://www.google.co.uk	legitimate
26	https://www.google.com.br	legitimate
27	https://www.google.fr	legitimate

	A	B
450153	http://gaptrade.cl/file/bless/index.php?email=	phishing
450154	http://gospelchurchofchrist.org/best_update/	phishing
450155	http://thepizzaplacesandimas.com/includes/b	phishing
450156	http://ui3china.com/still/G.Docs/index.php	phishing
450157	http://ui3china.com/live/G.Docs/index.php	phishing
450158	http://orlandoresorthouses.com/wwe/script/	phishing
450159	http://knitwear.ru/LinkedInen.html	phishing
450160	http://dizcorona.com/Via/Validation	phishing
450161	http://ayareview-document.pdf-iso.webapps-	phishing
450162	http://www.rosespa.com.sg/ipic/Dirk/index.p	phishing
450163	http://facebookauthorization.whatsgratis.con	phishing
450164	http://u.to/vYjNDw,Pattern	phishing
450165	https://insidethestorex.com/sd/	phishing
450166	http://youthsocialcircle.com/docs/Womsdhgc	phishing
450167	http://perrottaimmobiliare.it/img/immobiliari	phishing
450168	http://businessmobilewebapp.com/jobi/dh/dl	phishing
450169	http://bishopinegypt.gdn/wp-database/wp-da	phishing
450170	http://boasecg7.beget.tech/cgi-bin/index/pcg	phishing
450171	http://gangainsulations.com/alert/GD/	phishing
450172	https://qiumin.xyz/qiuminxy/o95j4uW4nr/5.pl	phishing
450173	http://ecct-it.com/docmmmnn/aptgd/index.p	phishing
450174	http://faboleena.com/js/infortis/jquery/plugir	phishing
450175	http://faboleena.com/js/infortis/jquery/plugir	phishing
450176	http://atualizapj.com/	phishing
450177	http://writeassociate.com/test/Portal/inicio/l	phishing
450178		
450179		

Fig. 2. Sample of URL dataset for Phishing detection.

3.2|Data Preprocessing

Prior to model training, raw URLs undergo a series of standalone preprocessing steps aimed at cleaning, standardizing, and structuring the data before it is passed to the DL model. These steps are deliberately separated from model-internal operations to ensure reproducibility, reduce input noise, and provide a consistent URL representation across experiments.

3.2.1|Uniform resource locator normalization and cleaning

We first normalize all URLs to remove redundant and non-discriminative components. Protocol prefixes such as http:// and https:// are stripped, as they do not contribute meaningful information for Phishing detection and may introduce unnecessary variance. Uniform elements and irrelevant special characters are removed or standardized while preserving the structural integrity of the URL. This step reduces noise and ensures consistency across samples.

3.2.2. | Character-level tokenization

Each normalized URL is treated as a sequence of individual characters. Character-level tokenization is adopted because Phishing URLs frequently exploit subtle character manipulations, obfuscation techniques, and deceptive patterns that are not adequately captured by word-level or handcrafted features. Each unique character is mapped to a discrete integer index, producing a numerical sequence representation. To enable batch processing and uniform input dimensionality, sequences are padded or truncated to a fixed maximum length. This standalone tokenization step produces structured integer sequences that serve as direct input to the DL model

3.2.3|Class imbalance handling

An important characteristic of the dataset used in this study is its significant class imbalance, which reflects real-world Phishing detection scenarios. As shown in *Table 1*, the number of legitimate URLs substantially

exceeds the number of Phishing URLs in both the training and test sets. Specifically, the training set contains 276,590 legitimate URLs compared to 83,550 Phishing URLs, while the test set includes 69,148 legitimate URLs and 20,888 Phishing URLs. This argument corresponds to an approximate class ratio of 3.3:1 in favor of legitimate URLs.

Such imbalance poses a critical challenge for supervised learning models, as classifiers trained on skewed data distributions tend to favor the majority class. In Phishing detection, this bias can lead to deceptively high overall accuracy while failing to correctly identify Phishing URLs, thereby increasing the rate of False Negatives (FNs). Given the security-sensitive nature of Phishing detection, misclassifying Phishing URLs as legitimate is particularly costly, as it exposes users to potential financial loss, identity theft, and data breaches. Therefore, addressing class imbalance is essential to ensure that the model learns representative patterns from both classes and maintains high recall for Phishing URLs.

Table 1. Class distribution of the training and test set.

Class	No. of Training Set	No. of Test Set
Legitimate	276,590	69,148
Phishing	83,550	20,888
Total	360,140	90,036

To mitigate the pronounced class imbalance observed in the dataset, SMOTE is applied exclusively to the training set. SMOTE generates synthetic Phishing samples by interpolating between existing minority-class instances, thereby increasing the representation and diversity of Phishing URLs without simply duplicating samples. Applying SMOTE only to the training data ensures that the model is exposed to a more balanced class distribution during learning, improving its ability to recognize Phishing patterns while avoiding bias toward the majority class. The test set is intentionally left unchanged to preserve its original data distribution and ensure an unbiased, realistic evaluation of model performance. *Table 2* presents the class distribution of the training dataset after applying SMOTE.

Table 2. Balanced distribution on the training set.

Class	No. of Training Set
Legitimate	276,590
Phishing	276,590
Total	553,180

3.3 | Bidirectional Long Short-Term Memory

The proposed system's core classification component is a Bi-LSTM network. LSTM networks are a specialized form of RNNs designed to effectively model long-range dependencies in sequential data by mitigating the vanishing gradient problem through gated memory units. This capability makes LSTM particularly suitable for Phishing URL detection, where discriminative patterns may appear at any position within a URL sequence. A standard LSTM cell regulates information flow using three gates (forget, input, and output) and a cell state that preserves contextual information across time steps. Given an input vector x_i at time step t and the previous hidden state h_{t-1}, the LSTM cell operations are defined as follows:

$$f_i = \sigma(W_f x_t + U_f h_{t-1} + b_f), \quad (1)$$

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i), \quad (2)$$

$$\hat{c}_t = \tanh(W_c x_t + U_c h_{t-1} + b_c), \quad (3)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \hat{c}_t, \quad (4)$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o), \quad (5)$$

$$h_t = o_t \tanh \odot (c_t), \quad (6)$$

where f_i, i_t, and o_t denote the forget, input, and output gates, respectively; $\hat{c}_t$ is the candidate cell state; c_t is the updated cell state; and h_t represents the hidden state. The matrices W, U, and bias vectors b are learnable parameters; $\sigma(.)$ and $\tanh(.)$ denote the activation functions. However, unlike the unidirectional LSTM, the

Bi-LSTM processes the input sequence in both forward and backward directions, enabling the model to capture dependencies from past and future contexts simultaneously. At each time step t, the hidden representations from the forward and backward LSTM layers are concatenated as:

$$h_t = [\overrightarrow{h_t}; \overleftarrow{h_t}], \quad (7)$$

where $\overrightarrow{h_t}$ and $\overleftarrow{h_t}$ denote the hidden states from the forward and backward LSTM passes, respectively. This concatenation h_t yields a richer representation of the URL sequence, allowing the model to effectively learn complex Phishing patterns that depend on both preceding and succeeding characters. As illustrated in *Fig. 3*, the bidirectional structure enables the model to incorporate full contextual information when classifying URL sequences, making it particularly effective for detecting obfuscated and deceptive Phishing URLs.

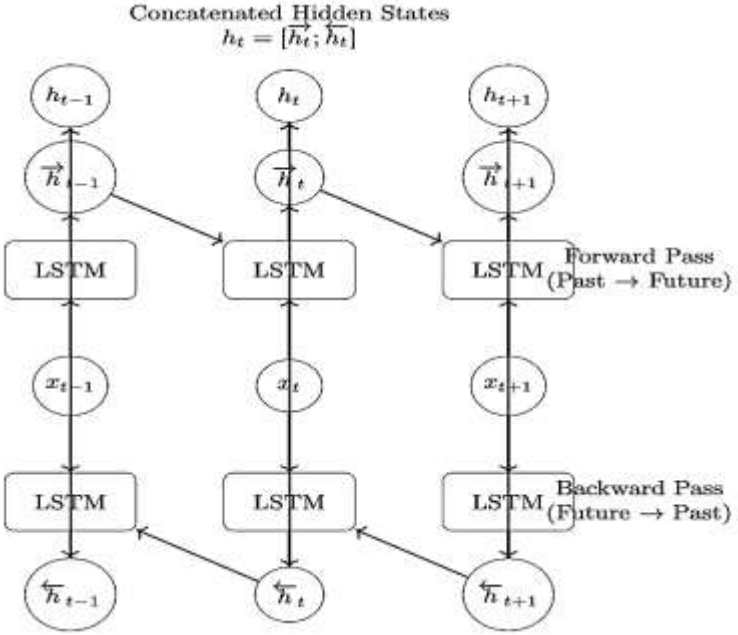


Fig. 3. Unrolled BiLSTM structure.

The forward LSTM processes the URL sequence from past to future, while the backward LSTM processes it from future to past. At each time step, the hidden states from both directions are concatenated to form a context-aware representation used for Phishing URL classification. *Fig. 4* shows the BiLSTM-based Phishing URL detection architecture.

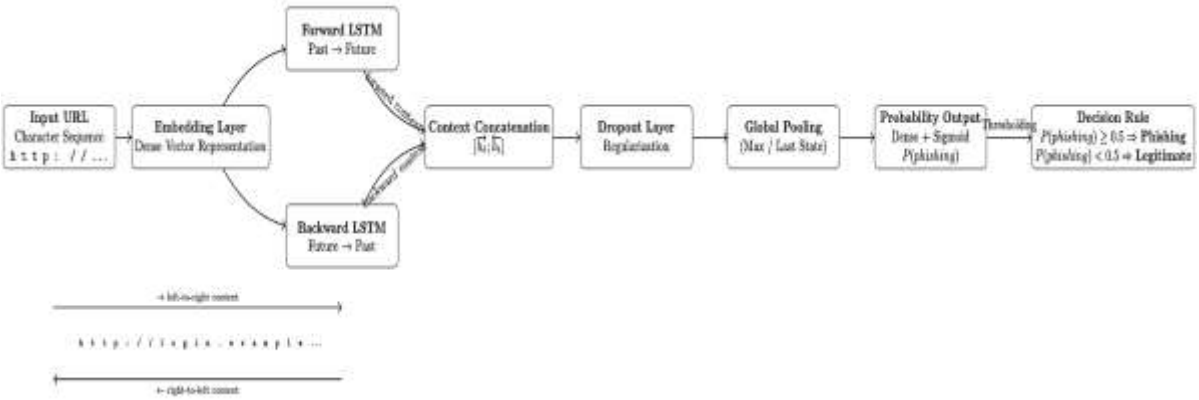


Fig. 4. BiLSTM-based Phishing URL detection architecture.

The bidirectional LSTM captures full contextual dependencies across URL sequences, while dropout and pooling improve generalization. A sigmoid-based binary classifier produces a probability score that is thresholded to classify URLs as Phishing or legitimate.

3.4 | Evaluation Metrics

To estimate the percentage in which the proposed Phishing website link detection model correctly distinguishes Phishing links from legitimate links, we use the following metrics: Acc, Pre, Recall, F1, defined in terms of True Positives (TP), True Negatives (TN), False Positives (FP), and FN. Also Receiver Operating Characteristic (ROC) and Area Under the Curve (AUC).

$$\text{Acc} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} * 100, \quad (8)$$

$$\text{Prec} = \frac{\text{TP}}{\text{TP} + \text{FP}} * 100, \quad (9)$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} * 100, \quad (10)$$

$$\text{F1} = 2 * \frac{\text{Precision x Recall}}{\text{Precision} + \text{Recall}} * 100, \quad (11)$$

Each metric provides a different perspective. Accuracy gives an overall correctness but can be inflated if the dataset is imbalanced (e.g., if legitimate URLs are much more common). Precision and recall focus on the Phishing class: precision penalizes FP (important to avoid false alarms), while recall penalizes FNs (important to catch all phishing). The F_1-score combines these two. In Phishing detection, a high recall is often critical to minimize missed attacks, but precision must also be high to avoid blocking legitimate sites. Thus, we report all these metrics to get a complete evaluation.

4 | Experimental Setup

All experiments were conducted on a computer system equipped with 8 GB RAM and an Intel® Core™ i5-4200U CPU operating at 2.30 GHz. The implementation was carried out using Python 3.7.6, leveraging widely adopted scientific and ML libraries. NumPy and Pandas handled numerical computation and data manipulation, while Matplotlib and Seaborn supported data visualization, including performance plots and confusion matrices. Data preprocessing and evaluation utilities were provided by scikit-learn, and DL models were developed using the TensorFlow/Keras framework. To accelerate training and improve computational efficiency, experiments were executed using Google Colab, which provides access to NVIDIA GPU resources. This setup ensured reproducibility while enabling efficient training of DL models on large-scale URL datasets. In addition, a CNN was implemented as a baseline model due to its proven effectiveness in capturing local character-level patterns in Phishing URLs. Both the CNN and the proposed BiLSTM were trained and evaluated under identical experimental conditions, including the same dataset splits, preprocessing steps, hyperparameters, and optimization strategy, to ensure a fair comparison. *Table 3* presents the models' parameterization

Table 3. Models parameterization.

Parameter	BiLSTM	CNN (Baseline)
Input shape	(100,5000)	(100,5000)
Number of layers	1 BiLSTM layer	1 Conv layer
Units/Filters	128 units (per direction)	128 filters
Pooling layers	Global max pooling	Max pooling (1D)
Dropout rate	0.2	0.2
Activation function	Tanh (LSTM), sigmoid (output)	ReLU(conv), sigmoid
Epochs	30	30
Batch size	128	128
Loss function	Binary cross-Entropy	Binary cross-entropy
Optimizer	Adam	Adam

5 | Results and Discussion

This subsection provides the experimental results of this study. This section explains the performance of all models in terms of accuracy, precision, recall, and F1-score.

5.1 | Training and Validation Loss

The training and validation performance of the BiLSTM and CNN models over the five epochs is presented in *Fig. 5*. The results indicate that the BiLSTM model demonstrated superior learning performance and generalization compared with the CNN model. For the BiLSTM, training accuracy increased consistently from approximately 79.5% in the first epoch to 83.8% by the fifth epoch, while validation accuracy increased from approximately 80.9% to 84.7% over the same period. This steady improvement in both training and validation accuracy indicates that the model progressively learned relevant patterns from the data. Similarly, the training loss decreased from approximately 0.478 to 0.382, while validation loss declined from approximately 0.444 to 0.363. The simultaneous increase in accuracy and reduction in loss suggest stable convergence and effective generalization, with no apparent evidence of overfitting within the observed training period.

In contrast, the CNN model exhibited comparatively slower learning and lower predictive performance. Its training accuracy increased from approximately 76.1% at the beginning of training to 77.8% by the fifth epoch. In comparison, validation accuracy remained around 76.8% during the initial epochs before increasing to approximately 79.0% at the final epoch. The CNN training loss experienced a substantial reduction from approximately 1.21 to 0.54 during the first epoch, after which the reduction became relatively gradual, reaching approximately 0.51 by the fifth epoch. Validation loss similarly decreased from approximately 0.54 to 0.45. Although the loss reduction indicates that the CNN was learning, the limited improvement in accuracy across the intermediate epochs suggests slower convergence and possible underfitting relative to the BiLSTM model.

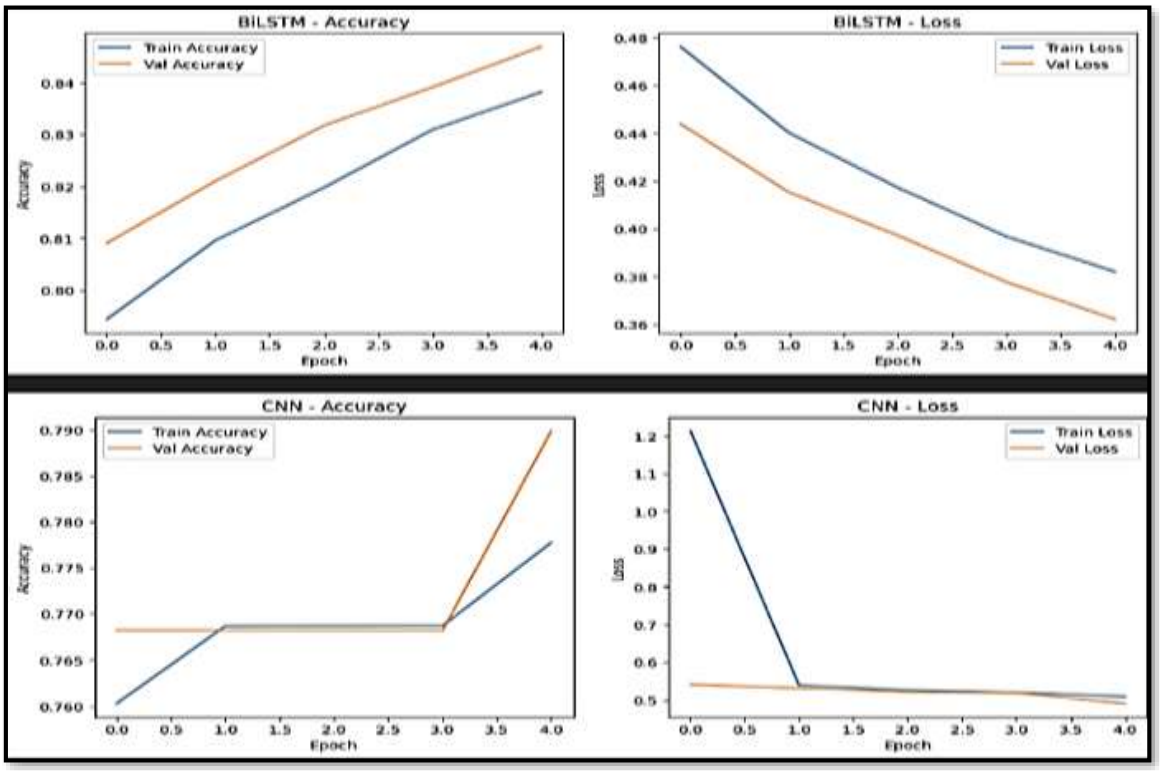
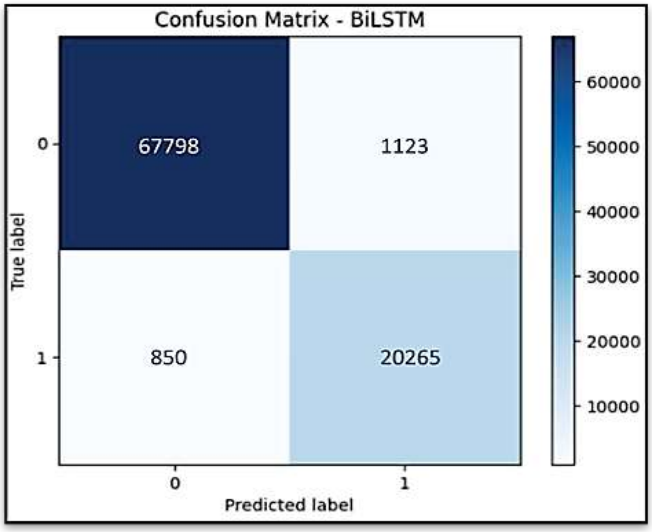


Fig. 5. Training and validation accuracy and loss plots.

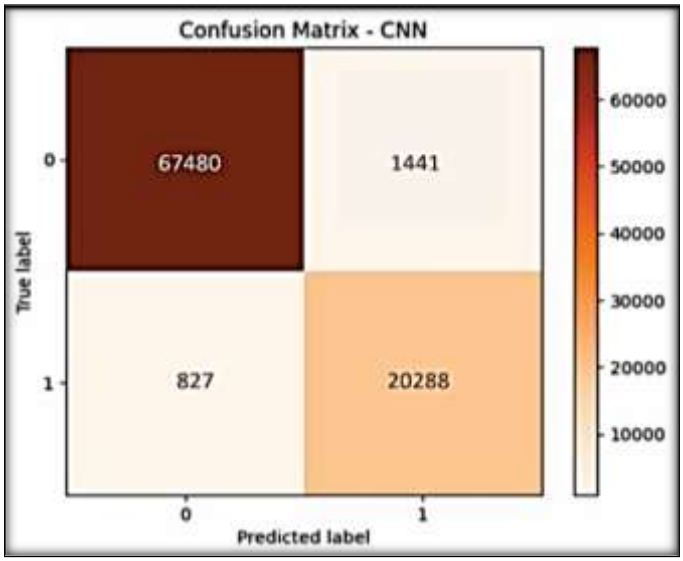
5.2 | Models Confusion Matrix

The confusion matrix analysis further substantiates the effectiveness of the BiLSTM model in Phishing URL detection. As illustrated in *Fig. 6*, out of approximately 90,000 test URLs, the BiLSTM correctly identified 20,265 Phishing URLs (TP) and 67,798 legitimate URLs (TN). The number of misclassifications was relatively small, with only 1,123 FP and 850 FN recorded.

These figures translate to a false-positive rate of 1.23%, where legitimate URLs were incorrectly flagged as phishing, and a false-negative rate of 0.94%, where actual Phishing URLs were missed. Such low error rates indicate that the BiLSTM model achieves a strong balance between sensitivity and specificity, effectively detecting Phishing attempts while minimizing unnecessary false alarms. In contrast, the CNN model shows slightly more FPs, despite achieving a comparable number of TPs. It indicates that while the CNN is highly sensitive to Phishing-related patterns, it is more prone to misclassifying legitimate URLs that contain locally suspicious substrings. This outcome is consistent with the CNN's reliance on local feature extraction, which may overlook broader structural context within URLs.



a.



b.

Fig. 6. Confusion matrices of the Phishing URL detection models: a. BiLSTM model and b. CNN model.

5.3 | Models Performance

Table 4 presents the comparative performance of the BiLSTM and CNN models on the Phishing URL test set, evaluated using accuracy, precision, recall, and F1-score. Both models demonstrate strong detection capability, achieving accuracies above 97%, which indicates that DL approaches are effective for identifying Phishing attacks from URL character sequences.

The BiLSTM model achieves the highest overall accuracy (97.81%) and F1-score (95.36%), reflecting a better balance between precision and recall. Its higher precision (94.75%) suggests that the BiLSTM is more effective at reducing FPs, meaning fewer legitimate websites are incorrectly flagged as phishing. It is particularly important in Phishing detection systems, as excessive false alarms can reduce user trust and hinder real-world adoption. The strong recall (95.97%) further indicates that the BiLSTM successfully captures a wide range of Phishing patterns, including obfuscated and deceptive URLs that rely on long-range contextual dependencies.

In contrast, the CNN model records a slightly higher recall (96.08%), indicating a marginally better ability to detect Phishing URLs. However, this comes at the cost of lower precision (93.37%), implying a higher false-positive rate compared to the BiLSTM. This behavior is consistent with CNNs' focus on local pattern extraction, which may overemphasize suspicious substrings without fully considering global URL context. Insightfully, the results show that while both models are suitable for Phishing detection, the BiLSTM provides a more robust and balanced performance, making it better suited for practical deployment where minimizing FP is as critical as detecting Phishing attacks.

Table 4. Model performance on the test set.

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
BiLSTM	97.81	94.75	95.97	95.36
CNN	97.48	93.37	96.08	94.71

5.4 | Receiver Operating Characteristic Analysis

The ROC curve provides a comprehensive assessment of a model's discriminative capability by illustrating the trade-off between the true positive rate and the false positive rate across varying decision thresholds. Unlike accuracy, which depends on a single threshold, the ROC curve evaluates performance across the full spectrum of operating points, making it particularly valuable for security-critical tasks such as Phishing detection. The AUC summarizes this behavior as a single scalar value: 1.0 indicates perfect class separation, while 0.5 corresponds to random guessing.

As shown in *Fig.7*, both the BiLSTM and CNN models exhibit strong discriminative power, with ROC curves that remain well above the diagonal reference line. it indicates that both models are capable of effectively distinguishing between phishing and legitimate URLs across a wide range of thresholds. However, the BiLSTM model achieves a higher AUC of 0.97, compared to 0.94 for the CNN, demonstrating a superior ability to separate the two classes consistently. This performance gap, although numerically modest, is significant in the context of Phishing detection, where even small improvements in discrimination can substantially reduce false alarms or missed attacks at scale.

The higher AUC of the BiLSTM suggests that its predictions are more stable and reliable when the decision threshold is adjusted to meet different operational requirements. For instance, in high-security environments where minimizing FNs is critical, the BiLSTM can maintain strong detection rates without a proportional increase in FPs. This robustness can be attributed to the BiLSTM's bidirectional sequential modeling, which captures both preceding and succeeding contextual patterns within URL character sequences, an advantage when dealing with obfuscated or deceptively crafted Phishing URLs.

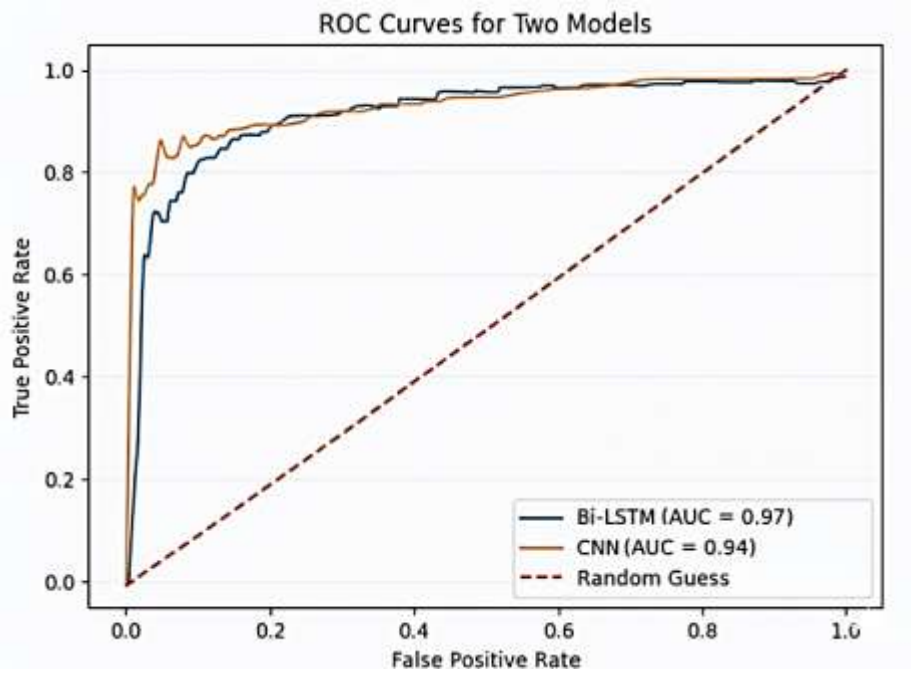


Fig. 7. BiLSTM and CNN ROC plot.

5.5 | Performance Comparison with Existing Studies

To better understand the relative effectiveness of the proposed model, its performance is systematically compared with representative Phishing detection approaches reported in the literature. This comparison considers not only classification accuracy but also the underlying modeling complexity and feature dependencies of existing methods. As shown in *Table 5*, the proposed BiLSTM demonstrates competitive performance relative to state-of-the-art techniques, while avoiding the computational overhead associated with extensive feature engineering, ensemble learning strategies, or multimodal data processing. By directly learning bidirectional sequential patterns from raw URL character sequences, the proposed approach achieves an effective balance between detection performance and computational efficiency, making it more suitable for scalable and practical Phishing detection deployments.

Table 5. Performance comparison with existing studies.

Study	Model	Feature Type	Accuracy (%)	Precision/ Recall/F1 (%)	Remarks
Barik et al. [8]	CNN+VAE+EGSO	TF-IDF text features	99.44	>99/>99/>99	Very high accuracy but optimization-heavy
Alsabri and Al-Hadi [9]	CNN+BiLSTM	Lexical, structural, domain features	81.00	Not reported	Lower accuracy; high computational complexity
Alsaedi et al. [4]	CTI-based RF+MLP Ensemble	URL+ Google + Whois features	96.80	Not Reported	Slightly lower accuracy than the proposed model
Raja et al. [3]	TF-IDF +ML	TF-IDF, Hash vectors	99.50	Not reported	High accuracy but high memory and computational cost
Fang et al. [12]	CNN-Transformer	Multimodal website features	96.30	Not reported	Slightly lower accuracy than the proposed model
Xie et al. [13]	TCN+ Attention	URL features	97.66	Not reported	Scalable, but attention adds complexity
Abid and Abid [14]	BiLSTM	Email and URL features	92.40	Not reported	Lower accuracy than the proposed model
Proposed model	BiLSTM + SMOTE	Character-level URL sequences	97.81	94.75 / 95.97 / 95.36	Balanced performance with low FP; lightweight and scalable

6 | Conclusion

This study successfully developed and evaluated a BiLSTM–based Phishing website detection model that effectively addresses the objectives outlined in this work. A large-scale benchmark dataset comprising 450,176 Phishing and legitimate URLs sourced from the Mendeley repository was employed to ensure a realistic and comprehensive evaluation. The URLs were systematically preprocessed through normalization and character-level tokenization, while class imbalance was effectively addressed using SMOTE. We evaluated the proposed model using standard performance metrics, including accuracy, precision, recall, F1-score, and ROC–AUC. We benchmarked it against a CNN baseline under identical experimental conditions to ensure a fair comparison. Experimental results demonstrate that the proposed BiLSTM model achieves 97.81% accuracy, 94.75% precision, 95.97% recall, and an F1-score of 95.36%, outperforming the CNN baseline in terms of accuracy and precision while maintaining comparable recall. These results highlight the effectiveness of bidirectional sequential modeling in capturing long-range contextual dependencies within URL character sequences, which obfuscated Phishing attacks commonly exploit. Overall, the proposed BiLSTM model offers a strong balance between detection performance and computational efficiency, making it a practical and scalable solution for real-world Phishing website detection.

Author Contributaion

Conceptualization, I. P. O. and Ch. C. U.; methodology, A. P. A., Ch. C. U., I. P. O.; software, A. P. A.; validation, T. A. U., O. A. M. and Ch. C. U. All authors have read and agreed to the published version of the manuscript.

Funding

This study received no external funding

Data Availability

All data supporting the reported findings in this study are provided within the manuscript

Conflicts of Interest

The authors declared no conflict of interest

Consent for Publication

The author confirms consent for the publication of this work.

Ethics Approval and Consent to Participate

This article does not contain any studies with human participants performed by the author.

References

[1] Al-Ahmadi, S., & Alharbi, Y. (2020). A deep learning technique for web Phishing detection combined URL features and visual similarity. *International journal of computer networks and communications*, *12*(5), 41–54. https://doi.org/10.5121/ijcnc.2020.12503

[2] Devan, K. P. K., Nagaraj, R., Kamaleshwaran, P., & Karthick, S. (2023). Detection of Phishing websites using machine learning. *Proceedings of the 2022 international conference on intelligent communication, control and devices (ICCCD)* (pp. 1–4). IEEE. https://doi.org/10.2139/ssrn.4366981

[3] Raja, A. S., Pradeepa, G., Mahalakshmi, S., & Jayakumar, M. S. (2023). Natural language based malicious domain detection using machine learning and deep learning. *Scientific and technical journal of information technologies, mechanics and optics*, *32*(2), 304–312. https://doi.org/10.17586/2226-1494-2023-23-2-304-312

[4] Alsaedi, M., Ghaleb, F. A., Saeed, F., Ahmad, J., & Alasli, M. (2022). Cyber threat intelligence-based malicious URL detection model using ensemble learning. *Sensors*, *22*(9), 3373. https://doi.org/10.3390/s22093373

[5] Kadiyala, P., KV, S. S., Shashank, B. S., & Kumar, K. A. (2021). Phishing website detection using emerging machine learning models. *Proceedings of the international conference on smart data intelligence (ICSMDI)* (pp. 1–5). IEEE. https://doi.org/10.2139/ssrn.3853016

[6] Sahingoz, O. K., Buber, E., Demir, O., & Diri, B. (2019). Machine learning based Phishing detection from URLs. *Expert systems with applications*, *117*, 345-357. https://doi.org/10.1016/j.eswa.2018.09.029

[7] Rayalla, P. S., Vivek, S. V., Golla, H., Katakam, G., & AN, M. Z. (2023). Detecting Phishing websites using deep learning. *1st-international conference on recent innovations in computing, science & technology* (pp. 1–4). IEEE. https://doi.org/10.2139/ssrn.4527759

[8] Barik, K., Misra, S., & Mohan, R. (2025). Web-based Phishing URL detection model using deep learning optimization techniques. *International journal of data science and analytics*, *20*(5), 4449–4471. https://doi.org/10.1007/s41060-025-00728-9

[9] Alsabri, A. A., & Al-Hadi, M. A. (2025). A hybrid CNN-BLSTM model for Phishing attack detection using deep learning to strengthen internet security. *Sana'a university journal of applied sciences and technology*, *3*(4), 964–972. https://doi.org/10.59628/jast.v3i4.1822

[10] Aljofey, A., Jiang, Q., Rasool, A., Chen, H., Liu, W., Qu, Q., & Wang, Y. (2022). An effective detection approach for Phishing websites using URL and HTML features. *Scientific reports*, *12*(1), 8842. https://doi.org/10.1038/s41598-022-10841-5

[11] Murhej, M., & Nallasivan, G. (2025). Component features based enhanced Phishing website detection system using EfficientNet, FH-BERT, and SELU-CRNN methods. *Frontiers in computer science*, *7*, 1582206. https://doi.org/10.3389/fcomp.2025.1582206

[12] Fang, C., Yin, X., Teng, S., Zhao, C., & Huang, D. (2025). Contrastive multimodal network for Phishing detection via enhanced website authenticity analysis. *International journal of digital crime and forensics, 17*(1), 1–23. https://dx.doi.org/10.2139/ssrn.5036126

[13] Xie, L., Zhang, H., Yang, H., Hu, Z., & Cheng, X. (2025). A scalable Phishing website detection model based on dual-branch TCN and mask attention. *Computer networks*, *263*, 111230. https://doi.org/10.1016/j.comnet.2025.111230

[14] Abid, I., & Abid, M. (2025). *AI-based Phishing detection for emails and URLs*. https://dx.doi.org/10.2139/ssrn.5336075

[15] Kaitholikkal, J. K. S., & Arthi, B. (2024). *Phishing URL dataset*. Mendeley data. https://doi.org/10.17632/vfszbj9b36.1