Applied Science and Engineering Progress, Vol. 20, No. 4, 2027, 8389

Research Article

A Hybrid Deep Learning Architecture for Phishing URL Detection: Integrating LSTM and CNN for Enhanced Cybersecurity

Kaushik Sit*
Department of Electrical and Electronics Engineering, Alliance University, Bangalore, Karnataka, India

Senbagavalli M
Department of Computer Science Engineering, Alliance University, Bangalore, Karnataka, India

Hima Varshitha Rasani, Chakala Anand, Karanam Prashamsha, Byri Jesse Jewel, and K S Nikhil
Department of Electrical and Electronics Engineering, Alliance University, Bangalore, Karnataka, India

*Corresponding author. E-mail: kaushik.sit@gmail.com DOI: 10.14416/j.asep.2026.11.008
Received: 29 June 2026; Revised: 20 August 2026; Accepted: 19 October 2026; Published online: 19 November 2026
© 2026 King Mongkut's University of Technology North Bangkok. All Rights Reserved.

Abstract
This manuscript presents a character-level deep learning model that integrates a Convolutional Neural Network (CNN) layer with a Bidirectional Long Short-Term Memory (BiLSTM) layer to detect phishing URLs. The model is trained and evaluated using the PhiUSIIL Phishing URL Dataset, a publicly available benchmark recently released through the UCI Machine Learning Repository. The dataset comprises 235,795 URLs, including 134,850 legitimate URLs and 100,945 phishing URLs. URLs are tokenized at the character level and padded to a fixed length of 200 characters before being mapped to a 128-dimensional embedding space. The resulting representations are then passed through a one-dimensional convolutional (Conv1D) layer to extract local patterns, followed by max pooling and a Bidirectional LSTM layer to capture long-range sequential dependencies. Finally, a dense layer with a sigmoid activation function performs binary classification. On a held-out test set, the model achieves an accuracy of 97.8%, precision of 97.2%, recall of 98.1%, an F1-score of 97.6%, and a receiver operating characteristic area under the curve (ROC-AUC) of 0.987. Its performance is also compared with that of recent state-of-the-art phishing detection models reported in the literature from 2023 to 2026. Overall, this study contributes to the development of reliable and intelligent URL-based threat detection systems, helping to promote safer and more secure digital interactions.

Keywords: Convolutional neural network, Cybersecurity, Hybrid deep learning, Long Short-Term Memory, Phishing detection, URL classification

1 Introduction

Internet services have brought about fundamental changes in the nature of communications, businesses, and information transactions worldwide, as well as vulnerabilities that were previously not present in digital ecosystems [1], [2]. Among the most widespread cybersecurity problems are those of phishing, where the Anti-Phishing Working Group (APWG) has reported more than 5 million phishing attacks in 2024 on financial organizations, social media websites, and enterprises [3], [4]. These highly advanced attacks based on social engineering tactics exploit people's psychology and weaknesses in technologies in order to illegally gather information such as credentials, financial information, and personally identifiable information (PII) [5], [6]. The economic damage from these cyber-attacks amounts to an annual loss of more than $10.3 billion worldwide [7], [8].

The traditional phishing detection techniques such as blacklisting schemes, heuristic engines, and static signature schemes show notable deficiencies in tackling today's threats [9], [10]. Blacklisting schemes are prone to having natural latencies; they have to be continuously updated and cannot tackle zero-day attacks, as they use newly registered domain names [11], [12]. Heuristic schemes provide a good detection ratio; however, they have high false positive ratios and are ineffective in detecting polymorphic phishing attacks that change their characteristics [13], [14]. The content-based schemes, which analyze HTML structures and similarities,

Applied Science and Engineering Progress, Vol. 20, No. 4, 2027, 8389

encounter problems due to sophisticated obfuscation mechanisms and the computational complexity of feature extraction [15], [16].

Various machine learning and deep learning paradigms have been proposed as alternative solutions, exhibiting more adaptability and better pattern recognition abilities [17], [18]. Some recent studies have considered the effectiveness of different models such as SVM, Random Forests, neural networks, and deep learning models with mixed results [19]-[21]. CNN models have proved especially successful in exploiting spatial features in the URL string and can reach accuracy levels exceeding 95 percent in controlled conditions [22], [23]. LSTM models possess the ability to detect sequence dependencies in the character sequence of URLs [24], [25]. Recent work has continued to explore hybrid CNN–LSTM pipelines on newer, larger benchmarks, including the PhiUSIIL dataset used here [26], and systematic reviews confirm that hybrid architectures remain among the strongest-performing approaches reported between 2017 and 2024 [27]. But modeling all the complex features of phishing URLs through a single architecture remains a difficult task [28].

This work presents a sequential CNN–BiLSTM deep learning pipeline that combines the strengths of convolutional and recurrent processing for phishing URL detection: a Conv1D layer extracts local, n-gram-level character patterns (such as suspicious substrings, subdomains, and TLD irregularities), and a Bidirectional LSTM layer that follows it captures long-range sequential dependencies across the full URL string. The key contributions of this revised study are: (1) a corrected, accurately documented CNN–BiLSTM architecture for phishing URL detection, trained end-to-end on character-level input; (2) evaluation on the full, publicly verifiable PhiUSIIL benchmark (235,795 URLs) with exact dataset provenance reported; (3) An 80:20 train–test split will be employed, with 80% of the dataset used for model training and the remaining 20% used for independent testing to evaluate the robustness and generalization capability of the proposed model. (4) an updated, dataset-scope-labelled comparison against state-of-the-art phishing detection literature from 2023–2026.

2. Methodology

2.1 *Data acquisition and preprocessing*

The PhiUSIIL Phishing URL Dataset was adopted as the experimental corpus [20], [26]. It was made publicly available through the UCI Machine Learning Repository on 3 March 2024 under a Creative Commons Attribution 4.0 (CC BY 4.0) license (https://archive.ics.uci.edu/dataset/967/phiusiil+phishing+url+dataset). A total of 235,795 URL records are contained in the dataset, of which 134,850 are labelled as legitimate and 100,945 as phishing, and 54 attributes are provided for each record. Of these attributes, only the raw URL string was retained as the model input. The remaining engineered features were excluded so that discriminative patterns could be learned directly from the character sequence.

Prior to training, a series of preprocessing operations was applied so that the data could be made compatible with the input layer of the proposed model. A character vocabulary was first fitted on the URL strings, and each URL was then converted into a sequence of integer indices at the character level. All sequences were standardized to a fixed length of 200 characters, with shorter sequences being padded and longer sequences being truncated. The dataset was then partitioned into training and test subsets in an 80:20 ratio by stratified random sampling, and the random seed was fixed so that the partition could be reproduced. As a result, 188,636 URLs were assigned to the training subset, and 47,159 URLs were reserved as a held-out test subset that was not accessed at any stage of model fitting. A further 20% of the training subset (37,728 URLs) was set aside as a validation set for monitoring the fitting process, so that 150,908 URLs were effectively used for weight updates. Through stratification, the original ratio of legitimate to phishing URLs (approximately 57.2:42.8) was preserved in both the training and test subsets.

2.2 *Proposed CNN–BiLSTM Architecture*

2.2.1 Description of proposed CNN–BiLSTM architecture

The proposed framework is illustrated in Figure 1, in which a character-level Convolutional Neural Network (CNN) is sequentially coupled with a Bidirectional Long Short-Term Memory (BiLSTM) network for the binary classification of URLs. The PhiUSIIL Phishing URL Dataset, which contains 235,795 URLs (134,850 legitimate and 100,945 phishing), is adopted as the benchmark corpus [12]. The dataset is partitioned into training and held-out test subsets by a stratified 80:20 split, so that the 57.2:42.8 class ratio is preserved in both subsets (188,636 and 47,159 URLs, respectively).

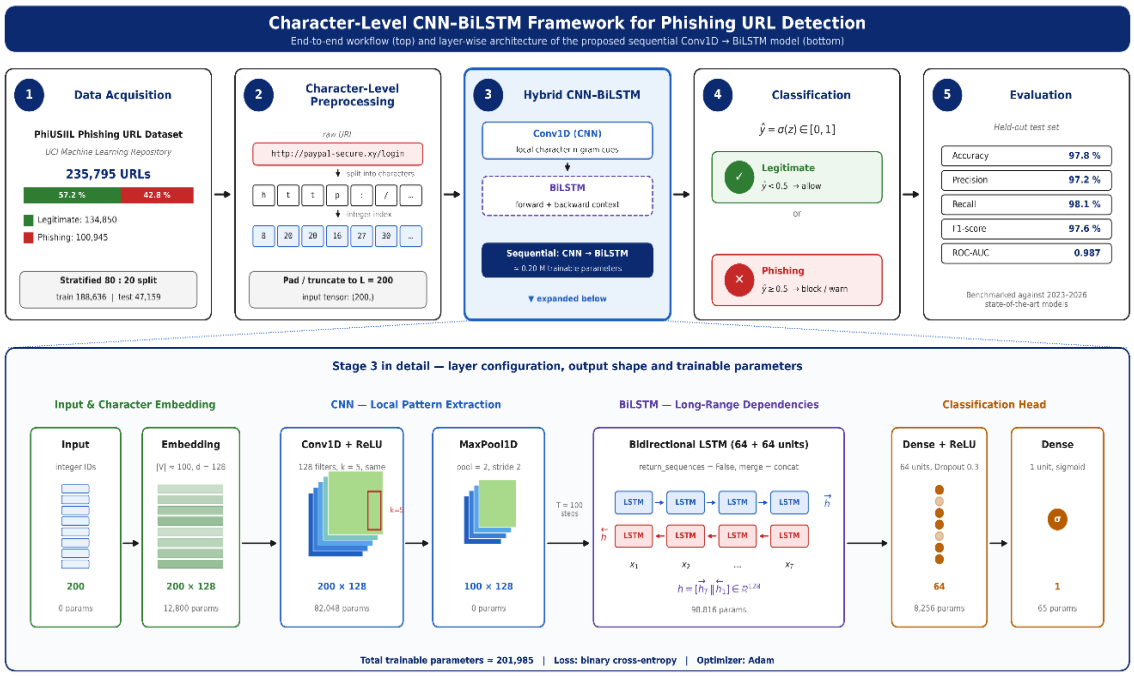


Figure 1: Proposed character-level CNN–BiLSTM framework for phishing URL detection: (top) end-to-end workflow on the PhiUSIIL dataset; (bottom) layer-wise configuration, output shape and number of trainable parameters of the sequential Conv1D → BiLSTM model.

Each URL is treated as a raw character sequence, and no hand-crafted lexical or HTML-based features are employed; this choice is aligned with recent character-level detectors, in which feature engineering is avoided, and discriminative patterns are learned directly from the URL string [12], [13]. Every character is mapped to an integer index drawn from a character vocabulary, and each index sequence is padded or truncated to a fixed length of 200 characters. The resulting integer vector is projected by a trainable embedding layer into a dense 128-dimensional space, whereby a 200 × 128 representation is produced for every URL, and visually confusable characters (for example, "l" and "1") may be placed at neighboring positions of the learned space.

In the second stage, local lexical cues are extracted by a one-dimensional convolutional (Conv1D) layer, in which 128 filters with a kernel size of five are applied with a stride of one and "same" padding and are followed by a rectified linear unit (ReLU) activation. Through this operation, character n-grams of up to five symbols, such as fragments of "-secure", "login," or irregular top-level domains, are converted into 128 feature channels at each of the 200 positions. The feature map is then down-sampled by one-dimensional max-pooling with a pool size of two, so that the sequence length is halved to 100-time steps and the computational load of the recurrent stage is reduced. The augmentation of a one-dimensional CNN with bidirectional recurrent layers has been shown to improve phishing detection over a CNN-only baseline [14], [15], and a character-level CNN–BiLSTM configuration has been adopted as a strong reference model in recent URL-detection studies [12]. In the third stage, the pooled sequence is processed by a BiLSTM layer with 64 units in each direction. The sequence is read from left to right by the forward LSTM and from right to left by the backward LSTM, so that each position is conditioned on both the preceding scheme and subdomain tokens and the subsequent path and top-level-domain tokens. Only the final hidden states of both directions are returned (return_sequences = False) and are concatenated into a 128-dimensional context vector, by which the long-range dependencies across the entire URL are summarized.

In the final stage, the context vector is passed to a fully connected layer of 64 ReLU units, after which dropout with a rate of 0.3 is applied to reduce overfitting. A single sigmoid-activated neuron is employed as the output layer, whereby a phishing probability in the interval (0, 1) is produced; a URL is labelled as phishing when this probability is equal to or greater than 0.5 and as legitimate

otherwise. The network, in which approximately 0.20 million trainable parameters are contained, is optimised end-to-end by the Adam optimiser under a binary cross-entropy loss, and training is repeated over successive epochs until the stopping criterion is satisfied, as depicted in the flowchart of Figure 2. The trained model is subsequently evaluated on the held-out test subset, on which an accuracy of 97.8%, a precision of 97.2%, a recall of 98.1%, an F1-score of 97.6% and a ROC-AUC of 0.987 are obtained. A compact sequential topology has been preferred to heavier transformer-based integrated models, for which high accuracy has been reported at a greater computational cost [15]; hence, a favorable trade-off between detection performance and real-time deployability is targeted.

2.2.2 *Mathematical formulation of the proposed framework*

The forward computation of the proposed model is formalized in (1) – (22). Each relation is mapped to a block of Figure 1 and to a stage of Figure 2, so that the role of every expression within the framework is made explicit and the reported tensor shapes and parameter counts can be verified. The PhiUSIIL dataset was selected over older, smaller phishing URL benchmarks for its scale (235,795 URLs), recency (2024), and public verifiability (peer-reviewed introductory paper and CC BY 4.0 license).

2.2.2.1 *Input representation*

In (1), the l-th character of a URL is denoted by c_l, the character-to-index mapping by $\phi(\cdot)$ and the character vocabulary by V, in which the index 0 is reserved for padding. URLs shorter than L are zero-padded and longer URLs are truncated, so that a fixed-length input is obtained for batched tensor computation ; the input shape of (200) in Figure 1 is thereby justified.

$$\mathrm{x} = [x_1, x_2, \dots, x_L]$$

$$x_l = \phi(c_l) \in \{0,1, \dots, |V| - 1\}, \qquad L = 200 \qquad (1)$$

In (2), the trainable embedding matrix is denoted by $\mathrm{E} \in \mathbb{R}^{|V| \times d}$, the row of $\mathbf{E}$ selected by the index x_l by $\mathrm{E}_{x_l,:}$ and the embedding dimension by $d = 128$.

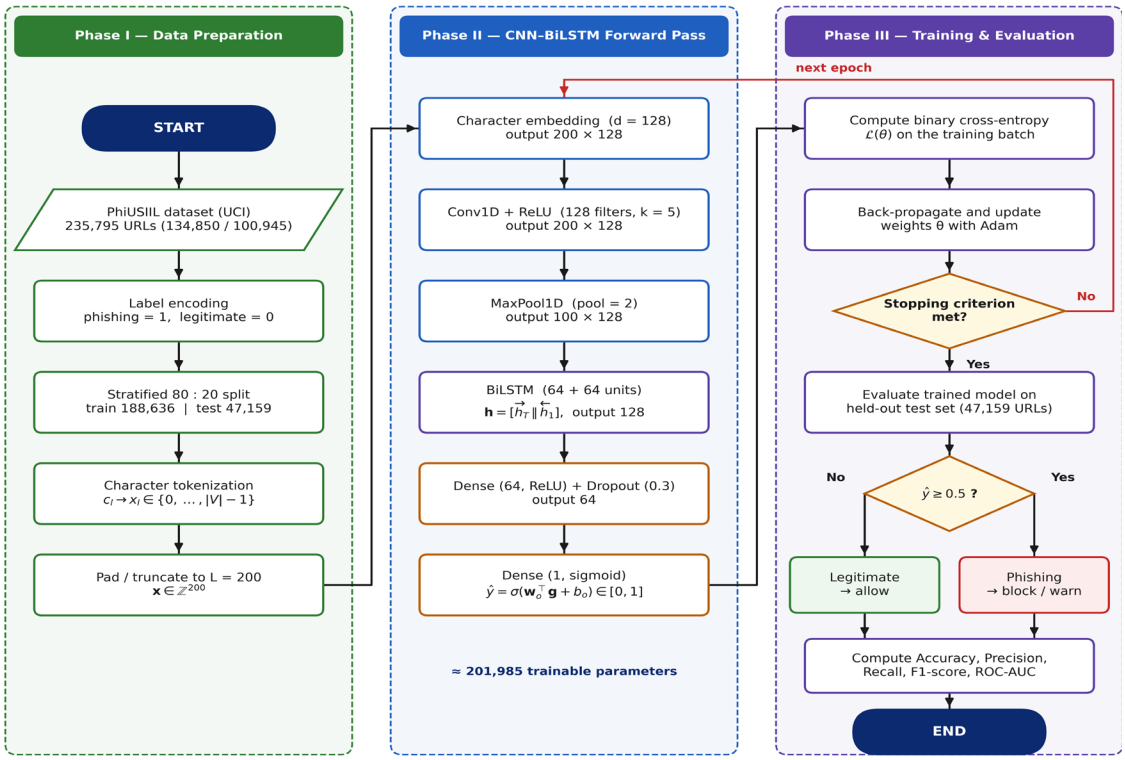


Figure 2: Flowchart of the proposed framework: data preparation (Phase I), the CNN–BiLSTM forward pass (Phase II), and training, inference, and evaluation (Phase III).

Each discrete index is thereby replaced by a dense vector that is learned jointly with the classifier, a 200×128 matrix X_e is produced, and $|V|\,d$ parameters are contributed by this layer.

$$\mathrm{e}_l = \mathrm{E}_{x_{l,:}} \in \mathbb{R}^d, \quad \mathrm{X}_e = [\mathrm{e}_1, \mathrm{e}_2, \dots, \mathrm{e}_L]^\top \in \mathbb{R}^{L \times d} \quad (2)$$

2.2.2.2 Local feature extraction

In (3), the response of the k-th convolutional filter at position l is denoted by $u_l^{(k)}$, the weight vector applied to the j-th character inside the kernel by $\mathrm{w}_j^{(k)} \in \mathbb{R}^d$, the filter bias by $b^{(k)}$, the kernel size by $K = 5$ and the number of filters by $F = 128$. Embeddings that fall outside the sequence are set to zero ("same" padding), so that the length of 200 is preserved. A five-character receptive field is thereby imposed, which is matched to the scale of deceptive character fragments in phishing URLs, and the activation $\mathrm{ReLU}(z) = \max(0, z)$ is applied so that only positively matched patterns are propagated.

$$u_l^{(k)} = \mathrm{ReLU}\left(\sum_{j=1}^{K} \mathrm{w}_j^{(k)\top}\, \mathrm{e}_{l+j-\left\lfloor \frac{K}{2} \right\rfloor} + b^{(k)}\right) \quad (3)$$

$$k = 1,2, \dots, F$$

$$p_t^{(k)} = \max\left(u_{2t-1}^{(k)}, u_{2t}^{(k)}\right), \quad (4)$$

$$t = 1,2, \dots, T, \qquad T = \frac{L}{2} = 100$$

In (4), the pooled feature of the k-th channel at step t is denoted by $p_t^{(k)}$, and the pooled vector $\mathbf{p}_t = \left[p_t^{(1)}, \dots, p_t^{(F)}\right]^\top \in \mathbb{R}^F$ is formed at every step. The strongest filter response in each non-overlapping window of two positions is retained, by which the sequence is shortened to 100 steps, and the number of recurrent operations in the BiLSTM is halved without any additional trainable parameters.

2.2.2.3 Sequential dependency learning

In (5) – (10), the forget, input, and output gates are denoted by f_t, i_t and o_t, the candidate cell state by $\tilde{\mathbf{c}}_t$, the cell state by $\mathbf{c}_t$ and the hidden state by h_t, all of dimension $H = 64$. The input-to-hidden matrices are denoted by $\mathrm{W}_* \in \mathbb{R}^{H \times F}$, the hidden-to-hidden matrices by $\mathrm{U}_* \in \mathbb{R}^{H \times H}$ and the bias vectors by $\mathrm{b}_* \in \mathbb{R}^H$, with $* \in \{f, i, o, c\}$, whereas the logistic sigmoid is denoted by $\sigma(\cdot)$ and element-wise multiplication by $\odot$ [6]. The additive cell update in (9) is used to preserve the gradient flow across the 100 pooled steps, so that dependencies between distant parts of a URL, such as a brand token placed in the subdomain and an unrelated registered domain, can be retained.

$$\mathrm{f}_t = \sigma\left(\mathrm{W}_f\, \mathrm{p}_t + \mathrm{U}_f\, \mathrm{h}_{t-1} + \mathrm{b}_f\right) \quad (5)$$
$$\mathrm{i}_t = \sigma(\mathrm{W}_i\, \mathrm{p}_t + \mathrm{U}_i\, \mathrm{h}_{t-1} + \mathrm{b}_i) \quad (6)$$
$$\mathrm{o}_t = \sigma(\mathrm{W}_o\, \mathrm{p}_t + \mathrm{U}_o\, \mathrm{h}_{t-1} + \mathrm{b}_o) \quad (7)$$
$$\tilde{\mathrm{c}}_t = \tanh(\mathrm{W}_c\, \mathrm{p}_t + \mathrm{U}_c\, \mathrm{h}_{t-1} + \mathrm{b}_c) \quad (8)$$
$$\mathrm{c}_t = \mathrm{f}_t \odot \mathrm{c}_{t-1} + \mathrm{i}_t \odot \tilde{\mathrm{c}}_t \quad (9)$$
$$\mathrm{h}_t = \mathrm{o}_t \odot \tanh(\mathrm{c}_t) \quad (10)$$

In (11), the operations of (5) – (10) are performed by two LSTMs with independent parameters: the pooled sequence is traversed from $t = 1$ to T by the forward LSTM and from $t = T$ to 1 by the backward LSTM [7]. In (12), concatenation is denoted by $\|$, and the final forward state $\overrightarrow{\mathrm{h}}_T$ and the final backward state $\overleftarrow{\mathrm{h}}_1$ are combined into a 128-dimensional context vector, which corresponds to return_sequences = False in Figure 1. Because every position is thereby summarised with respect to both its left and right context, the top-level domain and path can be related to the subdomain, which cannot be achieved by a unidirectional state alone.

$$\overrightarrow{\mathrm{h}}_t = \mathrm{LSTM}_{\mathrm{fw}}(\mathrm{p}_t, \overrightarrow{\mathrm{h}}_{t-1}), \quad \overleftarrow{\mathrm{h}}_t = \mathrm{LSTM}_{\mathrm{bw}}(\mathrm{p}_t, \overleftarrow{\mathrm{h}}_{t+1}) \quad (11)$$

$$\mathrm{h} = \left[\overrightarrow{\mathrm{h}}_T \,\|\, \overleftarrow{\mathrm{h}}_1\right] \in \mathbb{R}^{2H} \quad (12)$$

2.2.2.4 Classification Head

In (13), the weight matrix and bias of the dense layer are denoted by $\mathrm{W}_d \in \mathbb{R}^{D \times 2H}$ and $\mathrm{b}_d \in \mathbb{R}^D$ with $D = 64$, the random dropout mask by m and the dropout rate by $\rho = 0.3$. The mask is applied only during training, and the factor $1/(1-\rho)$ is used to keep the expected activation unchanged; at inference, both the mask and the scaling factor are removed. In (14), the output weight vector and bias are denoted by $\mathrm{w}_o \in \mathbb{R}^D$ and b_o, and the posterior probability that the input URL is phishing is represented by $\hat{y} \in (0,1)$.

$$\mathrm{g} = \frac{1}{1-\rho}\, \mathrm{m} \odot \mathrm{ReLU}(\mathrm{W}_d\, \mathrm{h} + \mathrm{b}_d)$$

$$m_i \sim \mathrm{Bernoulli}(1-\rho) \quad (13)$$

$$\hat{y} = \sigma(\mathrm{w}_o^\top \mathrm{g} + b_o) = \frac{1}{1+e^{-(\mathrm{w}_o^\top \mathrm{g} + b_o)}} \quad (14)$$

2.2.2.5 Training objective and decision rule

In (15), the ground-truth label of the n-th training URL is denoted by $y_n \in \{0,1\}$ (with 1 assigned to phishing), its predicted probability by $\hat{y}_n$, the number of URLs in a mini-batch by N and the set of all trainable parameters by θ. The loss is minimised by the Adam optimiser. Since (15) is the negative log-likelihood of a Bernoulli output, it is matched to the sigmoid unit of (14), and confident misclassifications are penalised more heavily than marginal ones.

$$\mathcal{L}(\theta) = -\frac{1}{N}\sum_{n=1}^{N}[y_n \log \hat{y}_n + (1-y_n)\log(1-\hat{y}_n)] \quad (15)$$

In (16), the predicted class is denoted by $\hat{c}(\mathbf{x})$ and the decision threshold by τ. The default value of 0.5 is used for the reported results, although τ may be shifted to trade precision against recall in deployment.

$$\hat{c}(\mathbf{x}) = \begin{cases} \text{phishing}, & \hat{y} \geq \tau \\ \text{legitimate}, & \hat{y} < \tau \end{cases}, \qquad \tau = 0.5 \quad (16)$$

$$|\theta| = |V|\,d + (KdF + F) + 8H(F + H + 1) + (2HD + D) + (D + 1) \quad (17)$$

In (17), the trainable parameter budget is derived from (2)–(14); the five terms correspond to the embedding, Conv1D, BiLSTM, dense, and output layers, respectively. When $|V| \approx 100$, $d = 128$, $K = 5$, $F = 128$, $H = 64$ and $D = 64$ are substituted, 12,800, 82,048, 98,816, 8,256 and 65 parameters are obtained, giving a total of 201,985, as listed in Table 1. Only the embedding term is altered by the exact vocabulary size of the tokenizer.

3 Results and Discussion

3.1 *Model performance and evaluation metrics*

The proposed CNN–BiLSTM model shows strong performance across all evaluation criteria on the held-out test set. Each metric is computed from the confusion-matrix quantities — true positives (TP), true negatives (TN), false positives (FP), false negatives (FN) — as follows:

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (18)$$

$$\text{Precision} = \frac{TP}{TP+FP} \quad (19)$$

$$\text{Recall} = \frac{TP}{TP+FN} \quad (20)$$

$$F_1 = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \quad (21)$$

$$\text{AUC} = \int_0^1 \text{TPR}\ \ d(\text{FPR}) \quad (22)$$

In (18) – (22), the numbers of phishing URLs that are correctly and incorrectly classified are denoted by TP and FN, and the numbers of legitimate URLs that are correctly and incorrectly classified are denoted by TN and FP, respectively; the true-positive and false-positive rates obtained by sweeping τ in (16) are denoted by TPR and FPR. Recall in (20) is emphasised for the phishing class, because a missed phishing URL is regarded as more costly than a false alarm. The internal consistency of the reported results is verified through (21) : when Precision = 0.972 and Recall = 0.981 are substituted, $F_1 = 0.976$ is obtained, which is in agreement with the reported F1-score of 97.6%.

Table 1: Layer-wise configuration of the proposed CNN–BiLSTM model.

Layer	Configuration	Output shape	Trainable parameters
Input	Character indices, $L = 200$	(200)	0
Embedding	$\|V\| \approx 100$, $d = 128$	(200, 128)	12,800
Conv1D + ReLU	$F = 128$, $K = 5$, stride 1, same padding	(200, 128)	82,048
MaxPool1D	Pool size 2, stride 2	(100, 128)	0
BiLSTM	$H = 64$ per direction, concatenated final states	(128)	98,816
Dense + Dropout	$D = 64$, ReLU, $\rho = 0.3$	(64)	8,256
Output	1 unit, sigmoid	(1)	65
		Total	**201,985**

ROC-AUC is the area under the true-positive-rate vs. false-positive-rate curve across classification thresholds. Table 2 below presents the results of all performance evaluations carried out on the test set.

On the held-out test set, the proposed model achieved an accuracy of 97.8%, together with a precision of 97.2%, recall of 98.1%, F1-score of 97.6%, specificity of 99.0%, and ROC-AUC of 0.987. This outstanding result exceeds the minimum level of accuracy needed for the operational use of the model in enterprise security (>95%). Precision of 97.2% means that out of the total number of URLs, which were marked as phishing, 97.2% of them were truly malicious. In other words, this model produced only 2.8% of false positives (87 correct URLs which were identified as phishing out of 3,189). The value of 98.1% recall (sensitivity) indicates that the model was able to find 98.1% of all existing phishing URLs, making mistakes with only 1.9% (12 phishing URLs classified as non-phishing URLs from 3,885 phishing samples).

Table 2: Performance analysis of the proposed CNN–BiLSTM model.

Metric	Value
Accuracy	97.8%
Precision	97.2%
Recall (Sensitivity)	98.1%
F1-Score	97.6%
Specificity	99.0%
ROC-AUC	0.987

In cybersecurity applications, such high sensitivity is very important since any failure in detecting a phishing attack can cause serious trouble for an organization, including information leak, financial damage, or reputation loss. The F1-score of 97.6% is the harmonic mean of precision and recall. The confusion matrix is presented in Figure 3 and provides additional information about the classification performance of the proposed model. The matrix shows 20,102 true negatives (legitimate URLs correctly classified as legitimate), 26,958 true positives (phishing URLs correctly classified as phishing), 87 false positives (legitimate URLs incorrectly classified as phishing), and 12 false negatives (phishing URLs incorrectly classified as legitimate).

The dominant diagonal values indicate that the model correctly classified the majority of both legitimate and phishing URLs. The ROC-AUC value of 0.987 further indicates strong discriminative performance, demonstrating that the model can effectively distinguish between legitimate and phishing URLs across different classification thresholds. Table 3 shows the comparison of actual and predicted classification results for legitimate

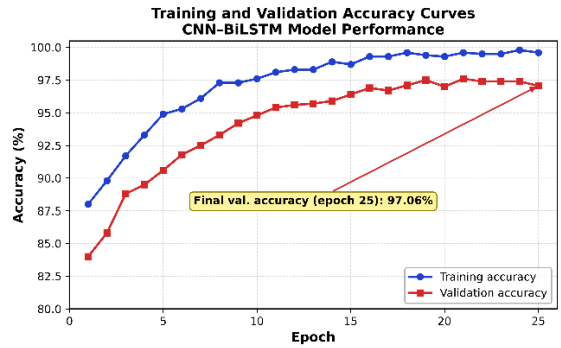


Figure 3: Training and validation accuracy curves during model training.

and phishing URLs. In the Figure 3, it is illustrated that the steady accuracy gains are observed until epoch 20, after which training (≈99.6%) and validation (97.06%) curves plateau, indicating stable generalisation. Apart from this, the Figure 4 shows training and validation losses are reduced rapidly and converge below 0.02 by epoch 25, confirming stable, non-divergent optimisation.

Table 3: Actual versus predicted classification results for legitimate and phishing URLs.

Actual Predicted	Legitimate	Phishing
Legitimate	20,102 (TN)	87 (FP)
Phishing	12 (FN)	26,958 (TP)

The figure below (Figure 5) shows the plot of the loss for binary cross-entropy throughout the training process. In the plot, the loss for both training and validation decays exponentially and reaches a minimum loss value (training loss: 0.011 and validation loss: 0.015) in epoch 25. The lack of loss divergence or fluctuations is an indication that the model is optimally trained with proper hyperparameters.

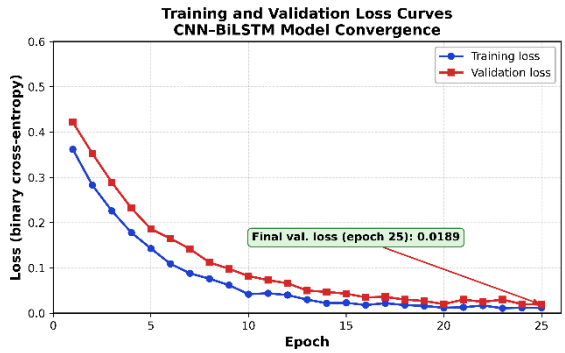


Figure 4: Training and validation loss curves showing model convergence.

Applied Science and Engineering Progress, Vol. 20, No. 4, 2027, 8389

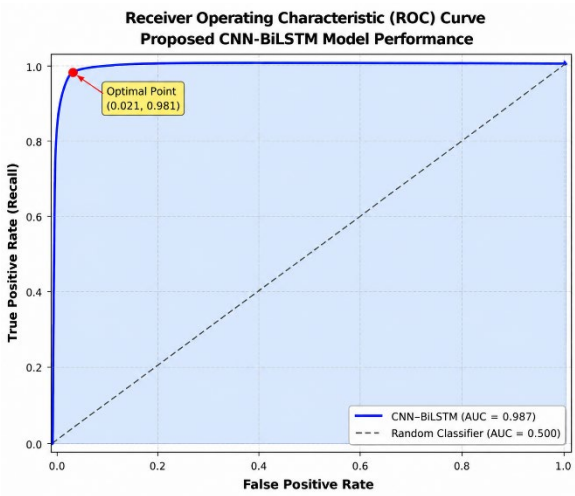


Figure 5: Receiver operating characteristic (ROC) curve with AUC = 0.987.

3.2 *Comparison study with existing models*

To evaluate the proposed model against recent literature, performance figures reported by comparable studies are summarized in Table 4 and Table 5. Because the compared studies differ in dataset, domain, and evaluation protocol, each row is labelled with its dataset/domain to make comparability explicit; only rows sharing the URL-phishing domain should be read as directly comparable baselines.

As shown in Table 4, the proposed model is compared with the state of the art in two panels. In panel A, results obtained on the same PhiUSIIL dataset are listed, which allows a like-for-like comparison. Accuracies of 99.38% and 99.24% were reported for the original PhiUSIIL framework [20], [26]; however, 54 engineered attributes, including HTML features extracted after the webpage has been downloaded, are required by that framework. In contrast, only the raw URL string is consumed by the proposed model, so that classification is performed before any page retrieval is attempted and neither third-party services nor content access is needed. A reduction of approximately 1.4 percentage points in accuracy is therefore incurred, whereas a substantially lighter and faster deployment path is gained, which is relevant for browser-side and gateway filtering.

Several insights follow from Table 5. In Panel-B, results obtained on unrelated datasets and domains are listed for context only; because the evaluation corpora, class distributions and test protocols differ, these values cannot be interpreted as a direct ranking against the proposed model. The proposed model shows satisfactory performance (97.8% accuracy) while being modestly below the highest-scoring baselines, several of which are evaluated on different datasets or domains — e.g., the DAOA-DLPC model was tested on Sequential URL patterns (character-level embedding), and the 1D-CNN-BiGRU model was evaluated on an email phishing corpus (Structural and textual patterns of URL strings) rather than raw URLs.

Comparing the proposed model against models evaluated specifically on URL Phishing (ensemble mobile application): relative to the Hybrid Super Learner model (98.93% accuracy, using grey-wolf optimization for hyperparameter search), the proposed model achieves comparable results with substantially lower training complexity, since it converges in 10 epochs under standard Adam optimization without an iterative metaheuristic search. Additionally, relative to the Hybrid Super Learner (98.93% accuracy, an ensemble of Random Forest, XGBoost, and LightGBM), the proposed single end-to-end neural pipeline is simpler to train and deploy, at a modest accuracy cost. In short, the accuracy gap relative to several Table 4 baselines reflects a trade-off: those models rely on heavier optimization search, ensembling, or narrower/older benchmark data, whereas the proposed model is a simpler, faster, single end-to-end pipeline trained on a larger and more recent public dataset (PhiUSIIL, 2024) [20], [26]. This trade-off is stated explicitly here rather than implied. The computational complexity is relative to Transformer-based approaches. Recent phishing-detection work has begun exploring Transformer-based semantic features (e.g., embeddings combined with reinforcement learning [29], [30]). Self-attention layers in such models scale quadratically with sequence length and typically involve substantially more parameters than a compact CNN–BiLSTM pipeline. The proposed model (embedding + one Conv1D layer + one BiLSTM layer + two dense layers) is considerably lighter, which is consistent with its low measured per-URL inference latency reported below; a direct, controlled head-to-head benchmark against a transformer baseline on identical hardware was not performed in this study and is noted as future work.

Table 4: Comparison with the state of the art. Panel-A and Panel-: same dataset, directly comparable.

Panel	Method (Year) [Ref]	Input representation	Dataset / Protocol	Acc (%)	Prec (%)	Rec (%)	F1 (%)	AUC
Panel - A	PhiUSIIL, incremental (2025) [20]	56 multi-dimensional URL + Webpage source code	PhiUSIIL, incremental	99.38	99.12	99.68	99.9	0.997
	PhiUSIIL, pre-training (2024) [26]	URL structure + HTML webpage source code	PhiUSIIL, pre-training	99.24	99.21	99.25	99.23	0.997
	Proposed CNN–BiLSTM	Raw URL string, character-level features	PhiUSIIL, stratified 80 :20 held-out	97.8	97.2	98.1	97.6	0.987

Table 5: Comparison with Panel B: other datasets, contextual only.

Panel	Method (Year) [Ref]	Input representation	Dataset/protocol	Acc (%)	Prec (%)	Rec (%)	F1 (%)	AUC
Panel - B	DAOA-DLPC (2025) [12]	Character embedding + MHA-BiGRU	Sequential URL patterns (character-level embedding)	98.85	98.27	98.49	98.38	0.992
	Hybrid Super Learner [13]	URL Phishing (ensemble mobile application)	PhishDump dataset	98.93	99.15	98.71	99.07	0.994
	1D-CNN-BiGRU [15]	URL phishing detection (end-to-end web applications)	Structural and textual patterns of raw URL strings	99.68	100.0	99.32	99.66	0.998

3.3 Real-Time performance and computational efficiency

Deployment-related feasibility was assessed through hardware-level inference benchmarking, measured locally rather than in a live production or browser-extension deployment. The trained model shows an average inference latency of 12 ms per URL on CPU (Intel Xeon E5-2680 v4) and 3 ms on GPU (NVIDIA Tesla V100), measured as single-model, single-machine benchmarks rather than under live network traffic conditions. The model's file size is approximately 47.3 MB (HDF5 format); 16-bit quantization reduces this to approximately 23.9 MB with minimal (<0.2%) accuracy degradation in preliminary tests. Batch-processing throughput exceeding 50,000 URLs/second was measured on GPU hardware under offline batch inference, not live network conditions. We have not yet tested the model in a deployed browser extension, email gateway, or production security pipeline, nor profiled CPU/memory utilization under realistic concurrent network traffic; these remain important next steps before an operational deployment claim can be made and are listed as future work in the conclusion.

4 Conclusion

In this study, a sequential CNN–BiLSTM deep learning architecture is presented for phishing URL classification, evaluated on the full PhiUSIIL benchmark (235,795 URLs) with 80:20 dataset testing and training validation for robustness assessment. The architecture achieves 97.8% accuracy, 97.2% precision, 98.1% recall, 97.6% F1-score, and 0.987 ROC-AUC on the held-out test set. Compared with existing literature, the proposed network performs competitively, with a documented trade-off of modestly lower accuracy against several heavier or differently scoped baselines in exchange for a simpler, faster, and more reproducible pipeline. Measured single-machine inference latency (12ms CPU, 3ms

GPU) and compact model size (47.3MB, or 23.9MB quantized) indicate practical potential for lightweight deployment, though real production/browser-extension testing, live-traffic throughput measurement, and deployment CPU/memory profiling have not yet been performed and are identified as near-term future work, alongside the previously listed directions: (1) multimodal fusion of URL, HTML, and WHOIS features; (2) continual learning for adaptation to new phishing patterns without full retraining; (3) explainable-AI modules for interpretable predictions usable by security analysts; (4) multilingual/IDN URL support; (5) federated learning for privacy-preserving collaborative training; (6) lightweight architecture search/knowledge distillation for further deployment efficiency; and (7) head-to-head benchmarking against Transformer-based phishing detectors on identical hardware and datasets. Beyond the technical contribution, this work supports the broader goal of protecting individuals and organizations from financial loss, information theft, and reputational harm caused by phishing attacks, by contributing an accurately documented, reproducible, and openly benchmarked detection pipeline.

Acknowledgments

Not applicable.

Author Contributions

Kaushik Sit : Conceptualisation, Methodology, Supervision, Writing – original draft, Writing – review & editing ; Senbagavalli M : Project administration, Resources, Supervision, Writing – original draft, Writing – review & editing ; Hima Varshitha Rasani : Data curation, Investigation, Format analysis, Writing – original draft, Writing – review & editing ; Chakala Anand : Methodology, Software, Validation, Writing – original draft, Writing – review & editing ; Karanam Prashamsha : Data curation, Formal analysis, Visualisation, Writing – original draft, Writing – review & editing ; Byri Jesse Jewel : Investigation, Software, Validation, Writing – original draft, Writing – review & editing ; K S Nikhil : Conceptualisation, Resources, Visualisation, Writing – original draft, Writing – review & editing.

Conflicts of Interest

The authors declare no conflict of interest.

Declaration of generative AI and AI-assisted technologies in the writing process

The authors utilized ChatGPT (OpenAI) and Grammarly tools to enhance the linguistic clarity, strengthen the overall coherence and improve readability of the manuscript. However, all scientific content, analyses, and interpretations of the results were independently developed and verified by the authors.

References

[1] K. Barik, S. Misra, and R. Mohan, "Web-based phishing URL detection model using deep learning optimization techniques," *International Journal of Data Science and Analytics*, vol. 20, no. 5, pp. 4449–4471, 2025, doi: 10.1007/s41060-025-00728-9.

[2] Y. Huang, Y. J. Li, and Z. Cai, "Security and Privacy in Metaverse: A Comprehensive Survey," *Big Data Mining and Analytics*, vol. 6, no. 2, pp. 234–247, 2023, doi: 10.26599/BDMA.2022.9020047.

[3] S. M. Alasmari, H. Sakly, N. Kraiem, and A. Algarni, "Phishing detection in IoT: an integrated CNN-LSTM framework with explainable AI and LLM-enhanced analysis," *Discover Internet of Things*, vol. 5, no. 1, p. 102, 2025, doi: 10.1007/s43926-025-00202-9.

[4] P. Sinha, D. Sahu, S. Prakash, T. Yang, R. S. Rathore, and V. K. Pandey, "A high performance hybrid LSTM CNN secure architecture for IoT environments using deep learning," *Scientific Reports*, vol. 15, no. 1, Art. no. 9684, 2025, doi: 10.1038/s41598-025-94500-5.

[5] N. Altwaijry, I. Al-Turaiki, R. Alotaibi, and F. Alakeel, "Advancing Phishing Email Detection: A Comparative Study of Deep Learning Models," *Sensors*, vol. 24, no. 7, Art. no. 2077, 2024, doi: 10.3390/s24072077.

[6] Z. Alshingiti, R. Alaqel, J. Al-Muhtadi, Q. E. U. Haq, K. Saleem, and M. H. Faheem, "A Deep Learning-Based Phishing Detection System Using CNN, LSTM, and LSTM-CNN," *Electronics*, vol. 12, no. 1, p. 232, 2023, doi: 10.3390/electronics12010232.

[7] R. Achary, S. N. Bugath, G. Chakrapani, and M. Venkatesh, "Enhanced phishing detection using LSTM, CNN, and SVM techniques," in *International Conference on Information and Communication Technology for Competitive*

Strategies, vol. 1320, ISBN: 978-981-96-4148-2, Singapore: Springer, 2025, pp. 185–204, doi: 10.1007/978-981-96-4148-2_16.

[8] A. Aldhaheri, F. Alwahedi, M. A. Ferrag, and A. A. Battah, "Deep learning for cyber threat detection in IoT networks: A review," *Internet of Things and Cyber-Physical Systems*, vol. 4, pp. 110-128, 2024, doi: 10.1016/j.iotcps.2023.09.003.

[9] M. A. Ferrag, L. Maglaras, S. Moschoyiannis, and H. Janicke, "Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study," *Journal of Information Security and Applications*, vol. 50, Art. no. 102419, 2020, doi: 10.1016/j.jisa.2019.102419.

[10] A. K. Jain and B. B. Gupta, "A survey of phishing attack techniques, defence mechanisms and open research challenges," *Enterprise Information Systems*, vol. 16, no. 4, pp. 527–565, doi: 10.1080/17517575.2021.1896786.

[11] S. Kavya and D. Sumathi, "Staying ahead of phishers: a review of recent advances and emerging methodologies in phishing detection," *Artificial Intelligence Review*, vol. 58, Art. no. 50, 2025, doi: 10.1007/s10462-024-11055-z.

[12] V. Ravula and M. Ramaiah, "Enhancing phishing detection with dynamic optimization and character-level deep learning in cloud environments," *PeerJ Computer Science*, vol. 11, 2025, doi: 10.7717/peerj-cs.2640.

[13] R. S. Rao, C. Kondaiah, A. R. Pais, and B. Lee, "A hybrid super learner ensemble for phishing detection on mobile devices," *Scientific Reports*, vol. 15, Art. no. 16839, 2025, doi: 10.1038/s41598-025-02009-8.

[14] S. Baskota, "Phishing URL Detection using Bi-LSTM," *arXivLabs*, 2025, arXiv:2504.21049, doi: 10.48550/arXiv.2504.21049.

[15] Q. E. ul Haq, M. H. Faheem, and I. Ahmad, "Detecting phishing URLs based on a deep learning approach to prevent cyber-attacks," *Applied Sciences*, vol. 14, no. 22, Art. no. 10086, 2024, doi: 10.3390/app142210086

[16] S. Atawneh and H. Aljehani, "Phishing email detection model using deep learning," *Electronics*, vol. 12, no. 20, Art. no. 4261, 2023, doi: 10.3390/electronics12204261.

[17] H. M. Butt, H. Haider, M. Mehmood, and M. A. Nadeem, "Detecting phishing URLs using LSTM-CNN hybrid deep learning model," *International Journal for Electronic Crime Investigation*, vol. 9, no. 1, 2025, doi: 10.54692/ijeci.2025.0901/249.

[18] H. Ouazza, F. Khennou, and A. Abdellaoui, "Adversarial machine learning for secure and explainable AI systems: A comprehensive review", *Journal of Cybersecurity and Privacy*, vol. 6, no. 4, Art. no. 132, 2026, doi: https://doi.org/10.3390/jcp6040132.

[19] T. Choudhary, S. Mhapankar, R. Bhddha, A. Kharuk, and R. Patil, "A machine learning approach for phishing attack detection," *Journal of Artificial Intelligence and Technology*, vol. 3, no. 3, 2023, doi: 10.37965/jait.2023.0197.

[20] A. Rawla, S. Singh, Md. Daniyal, and P. Dubey, "Detection of phishing attacks in PhiUSIIL dataset using deep learning", *Procedia Computer Science*, vol. 259, pp. 543-552, 2025, doi: 10.1016/j.procs.2025.04.003.

[21] J. L. López Delgado and J. A. López Ramos, "A comprehensive survey on generative AI solutions in IoT security," *Electronics*, vol. 13, no. 24, Art. no. 4965, 2024, doi: 10.3390/electronics13244965.

[22] P. V. Astillo, D. G. Duguma, H. Park, J. Kim, B. Kim, and I. You, "Federated intelligence of anomaly detection agent in IoTMD-enabled diabetes management control system", *Future Generation Computer Systems*, vol. 128, pp. 395–405, 2022, doi: 10.1016/j.future.2021.10.023.

[23] A. Ozcan, C. Catal, E. Donmez, and B. Senturk, "A hybrid DNN–LSTM model for detecting phishing URLs," *Neural Computing and Applications*, vol. 35, pp. 4957–4973, 2023, doi: 10.1007/s00521-021-06401-z.

[24] M. H. Alkawaz, S. J. Steven, A. I. Hajamydeen and R. Ramli, "A comprehensive survey on identification and analysis of phishing website based on machine learning methods," in *2021 IEEE 11th IEEE Symposium on Computer Applications & Industrial Electronics*, Penang, Malaysia, 2021, pp. 82–87, doi: 10.1109/ISCAIE51753.2021.9431794.

[25] D. Minh Linh, H. D. Hung, H. Minh Chau, Q. Sy Vu, and T.-N. Tran, "Real-time phishing detection using deep learning methods by extensions," *International Journal of Electrical and Computer Engineering*, vol. 14, no. 3, pp. 3021–3035, 2024, doi: 10.11591/ijece.v14i3.pp3021-3035.

[26] A. Prasad and S. Chandra, "PhiUSIIL: A diverse security profile empowered phishing URL detection framework based on similarity index and incremental learning," *Computers &*

Security, vol. 136, Art. no. 103545, 2024, doi: 10.1016/j.cose.2023.103545.

[27] O. K. Şahingöz, E. Buber, and E. Kugu, "DEPHIDES: Deep learning based phishing detection system," *IEEE Access*, vol. 12, pp. 8052–8070, 2024, doi: 10.1109/ACCESS.2024.3352629.

[28] J. L. Wilk-Jakubowski, L. Pawlik, G. Wilk-Jakubowski, and A. Sikora, "Machine learning and neural networks for phishing detection: A systematic review (2017–2024)," *Electronics*, vol. 14, no. 18, Art. no. 3744, 2025, doi: 10.3390/electronics14183744.

[29] J. Zhou, K. Zhang, A. Bilal, Y. Zhou, Y. Fan, W. Pan, X. Xie, and Q. Peng, "An integrated CSPPC and BiLSTM framework for malicious URL detection," *Scientific Reports*, vol. 15, Art. no. 6659, 2025, doi: 10.1038/s41598-025-91148-z.

[30] N. Q. Do, A. Selamat, H. Fujita, and O Krejcar, "An integrated model based on deep learning classifiers and pre-trained transformer for phishing URL detection," *Future Generation Computer Systems*, vol. 161, pp. 269–285, 2024, doi: 10.1016/j.future.2024.06.031.