IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Ingénierie des Systèmes d'Information
Vol. 30, No. 5, May, 2025, pp. 1219-1227
Journal homepage: http://iieta.org/journals/isi

# Phishing URL Detection Using Deep Learning: A Resilient Approach to Mitigating Emerging Cybersecurity Threats

Muhannad Almohaimeed[1], Faisal Albalwy[2*], Leinah Algulaiti[1], Hanan Althubyani[1]

[1] Department of Information Systems, College of Computer Science and Engineering, Taibah University, Madinah 42353, Saudi Arabia
[2] Department of Cybersecurity, College of Computer Science and Engineering, Taibah University, Madinah 42353, Saudi Arabia

Corresponding Author Email: fbalwy@taibahu.edu.sa



https://doi.org/10.18280/isi.300510

**Received:** 4 March 2025
**Revised:** 12 May 2025
**Accepted:** 27 May 2025
**Available online:** 31 May 2025

***Keywords:***
*bidirectional gated recurrent unit (BiGRU), character-level embedding, convolutional neural network (CNN), cybersecurity, deep learning, machine learning, phishing detection, URL analysis*

**ABSTRACT**

Phishing via malicious URLs remains a significant cybersecurity threat, exacerbated by the increasing dependence on digital platforms for communication, transactions, and data exchange. The ability to accurately distinguish between legitimate and phishing URLs is critical for safeguarding sensitive information and mitigating cyber threats. This study proposes a deep learning-based phishing URL detection model that processes raw URL input without requiring manual feature engineering. The model integrates char-acter-level embeddings with a hybrid parallel CNN-BiGRU architecture, leveraging Parallel CNN layers for local pattern extraction and BiGRU for capturing sequential dependencies in URL structures. The experimental results demonstrate that the proposed model achieves 98.46% accuracy, an AUC curve score of 99.62%, precision 98.45%, and recall 98.46%, along with a F1-score 98.45%. The hybrid architecture outperforms the utilization of individual CNNs since it combines parallel convolutional layers for local features as well as BiGRU for sequential relationships to offer more balanced and global performance on all of the measure metrics surpassing the performance of existing phishing detection frameworks. These findings underscore the effectiveness of combining convolutional and recurrent neural networks to enhance phishing detection capabilities. The study contributes to advancing cybersecurity defenses by providing an efficient, reliable, and scalable deep learning-based phishing detection framework capable of adapting to evolving phishing tactics.

## 1. INTRODUCTION

In the digital landscape, phishing has become one of the most pervasive and rapidly evolving cyber threats, targeting individuals and organizations worldwide. According to the APWG Phishing Activity Trends Report, over 1,077,501 phishing attacks were recorded in Q4 2023, marking a continuous surge in these malicious activities and underscoring the urgency for robust detection mechanisms [1, 2]. As attackers develop increasingly deceptive tactics, traditional security measures often fail to keep pace, necessitating advanced, AI-driven solutions [3-5].

Phishing is a social engineering attack in which cybercriminals use fraudulent communication channels, such as emails, SMS (smishing), or phone calls (vishing), to deceive victims into revealing sensitive credentials or installing malware. Among these, phishing via URLs has become particularly concerning, as attackers craft deceptive web links that mimic legitimate domains, exploiting human cognitive biases and URL mis-interpretation vulnerabilities [6-8]. These attacks leverage typo squatting, homoglyph substitutions, and domain spoofing techniques to evade detection, making them increasingly difficult to identify with conventional methods.

Existing phishing detection approaches fall into two primary categories: heuristic based techniques and machine learning (ML)-based models [9]. Heuristic approaches rely on manually defined rules, such as blacklist databases and URL pattern matching, but they struggle to detect newly crafted phishing URLs [10]. Machine learning-based methods, on the other hand, require extensive feature engineering, where human experts define characteristics such as domain age, URL length, and lexical patterns to train classification models [11-13]. But these methods are sometimes con-strained by their reliance on domain-specific knowledge and their incapacity to dynamically change assault tactics [14-16].

To address these limitations, this research suggests using deep learning to handle raw URLs without the need for manually created characteristics. Particularly, we pro-vide a hybrid architecture that integrates character-level embeddings with a parallel Convolutional Neural Network (CNN) and a Bidirectional Gated Recurrent Unit (BiGRU) for improved sequential learning and feature representation.

Character-level embedding is a method whereby each character is considered as an atomic unit of information, hence

transforming raw URLs into dense numerical representations [17]. Otherwise, word-based embeddings, which struggle to detect subtle obfuscation techniques (e.g., replacing 'o' with '0' in 'g00gle.com'), character embeddings allowed the model to learn fine-grained textual patterns, making them highly effective for phishing detection [18, 19]. Additionally, this approach eliminates the need for domain expertise in feature selection, thereby enhancing scalability and adaptability to new phishing tactics.

CNNs are well-suited for extracting local n-gram patterns within URLs, identifying common substrings frequently used in phishing domains [20, 21]. Meanwhile, the BiGRU component efficiently captures long-range dependencies and contextual relationships within URL sequences [22, 23]. Unlike traditional recurrent neural networks (RNNs), BiGRUs utilize gating mechanisms that mitigate vanishing gradient issues and improve training efficiency [24, 25]. Furthermore, the bidirectional processing capability of BiGRU ensures that critical phishing-indicative patterns are detected regardless of their position in the URL [26, 27].

The main contributions of this research are as follows:

- A comprehensive evaluation of recent deep learning-based phishing detection models, identifying their strengths and limitations.
- Development of a data preprocessing pipeline to enhance the quality and balance of phishing URL datasets.
- Proposal of a novel hybrid Parallel CNN-BiGRU architecture that improves phishing detection accuracy by effectively capturing both local and sequential URL features.

These contributions advance the state-of-the-art in phishing detection, offering a scalable, efficient, and end-to-end deep learning solution that enhances cybersecurity defenses against sophisticated phishing attacks.

The remainder of this paper is structured as follows: Section 2 presents a com-prehensive literature review, examining recent advancements in deep learning for phishing detection. Section 3 outlines the proposed methodology, detailing the dataset, model architecture, and training procedures. Section 4 discusses experimental results and comparative analysis, and Section 5 concludes the study by summarizing key findings and future research directions.

## 2. LITERATURE REVIEW

### 2.1 Overview of phishing detection techniques

Phishing detection has become a significant research focus due to the increasing sophistication of cyberattacks and their severe impact on individuals and organizations. Traditional phishing detection approaches, such as blacklist-based methods and heuristic rule-based systems, suffer from limited adaptability to emerging phishing techniques, as attackers frequently alter URLs, website structures, and obfuscation methods to evade detection.

Researchers have thoroughly investigated machine learning (ML) and deep learning (DL) techniques to address the issues, which allow for automated feature extraction and enhanced DL models like Long Short-Term Memory Networks (LSTMs), Convolutional Neural Networks (CNNs), Gated Recurrent Units (GRUs), and hybrid architectures have proven it is effective in identifying phishing patterns with high accuracy.

This section describes a comprehensive review of advanced DL-based phishing detection studies, categorized based on their model architecture and extraction techniques. Table 1 provides a comparative summary of the reviewed DL-based phishing detection studies, highlighting datasets, architectures, feature extraction methods, and key performance metrics.

**Table 1.** Summary of deep learning-based phishing detection studies

| Study | Dataset | DL Method | URL Handling Technique | Results |
|---|---|---|---|---|
| Korkmaz et al. [28] | High-Risk URL and Content-Based Dataset (87,489 samples: 51,316 legitimate, 36,173 phishing) | CNN, GAN And DNN | URLs Character embedding, URLs Features analysis | 98.37% |
| Aldakheel et al. [29] | 20,000 URLs (10,000 phishing from Phish Tank, 10,000 legitimate from the open datasets of the University of New Brunswick) | CNN | URLs Character embedding | 98.01% |
| Alshingiti et al. [30] | ISCX-URL2016 dataset: 20,000 URLs (9,800 phishing, 10,200 legitimate) | CNN LSTM And Hybrid LSTM-CNN | URLs Features analysis | CNN: 99.2% LSTM-CNN:97.6% LSTM: 96.8% |
| Elberri et al. [4] | PhishTank, UCI, and Tan datasets | CNNs-LSTMs | URLs Features analysis | Tan: 98.79% PhishTank: 99.37% UCI: 98.87% |
| Driss and Zougagh [31] | Dataset contains 11,430 instances of both phishing and legitimate URLs, also including 87 features | DNN | URLs Features analysis | 99.43% |
| Alsubaei et al. [32] | Dataset from Kaggle contains 10,000 URLs (phishing and legitimate) | ResNeXt-embedded Gated Recurrent Unit (RNT) with Jaya optimization | URLs Features analysis | 98% |
| Linh et al. [33] | Dataset collected from 5 datasets. The total of malicious URLs is 651191 including 428,103 as benign, 96,457 as defacement, 94,111 as phishing, and 32,520 as malware | LR, DT, RF, SVM, CNN, and CNN-LSTM | URLs Character embedding | CNN: 98.42% |
| Korkmaz et al. [28] | High-Risk URL and Content-Based Dataset (87,489 samples: 51,316 legitimate, 36,173 phishing) | CNN, GAN And DNN | URLs Character embedding, URLs Features analysis | 98.37% |

### 2.2 Deep learning approaches for phishing detection

2.2.1 CNN-based phishing detection models

CNNs are widely used in phishing detection due to their ability to extract spatial patterns from URLs and web-related features. Several studies have demonstrated CNNs' effectiveness in phishing detection:

- Korkmaz et al. [28] proposed a hybrid CNN-GAN phishing detection system, combining URL-based and content-based feature analysis. The approach achieved 98.37% accuracy, demonstrating the advantages of auto-mating feature extraction while integrating traditional feature engineering.
- Aldakheel et al. [29] developed a CNN-based model trained on 20,000 URLs (10,000 phishing and 10,000 legitimate), utilizing character embedding to represent URL patterns. Their approach achieved 98.01% accuracy, out-performing ML classifiers such as k-Nearest Neighbors (87%) and Random Forest (94.26%).
- Alshingiti et al. [30] compared CNN, LSTM, and a hybrid CNN-LSTM model using the ISCX-URL2016 dataset. Their CNN-based model achieved 99.2% accuracy, outperforming LSTMs and showing CNNs' effectiveness in extracting localized phishing patterns.
- Despite their high accuracy, CNNs are limited in capturing sequential relationships in URL structures, which affects their ability to adapt to complex phishing attack variations.

2.2.2 LSTM- and GRU-based phishing detection models

Recurrent Neural Networks (RNNs), particularly LSTMs and GRUs, are widely used for analyzing sequential URL patterns and long-range dependencies in phishing detection.

- Elberri et al. [4] proposed a CNN-LSTM phishing detection model, enhanced by the African Vulture Optimization Algorithm (AVOA) for feature selection. Their N-Gram-based URL representation achieved 99.37% accuracy on the PhishTank dataset, demonstrating LSTMs' ability to model sequential dependencies.
- Driss and Zougagh [31] developed a deep neural network (DNN) model utilizing a dataset compiled from Alexa, Yandex, PhishTank, and Open-Phish. Their feature selection approach, using chi-squared tests, enhanced model interpretability while achieving 99.43% accuracy.

While LSTMs and GRUs effectively capture sequential dependencies, they often rely on extensive feature selection techniques, which introduce manual preprocessing overhead and limit adaptability to evolving phishing strategies.

2.2.3 Hybrid deep learning models

Several studies have combined CNNs with RNN-based architecture to leverage both local feature extraction (CNNs) and sequential pattern recognition (LSTMs/GRUs):

- Alsubaei et al. [32] developed a hybrid ResNeXt-GRU model, optimized using the Jaya algorithm. Their study emphasized hyperparameter tuning's role in performance enhancement, achieving 98% accuracy on a Kaggle dataset.
- Linh et al. [33] proposed a real-time phishing detection system integrated into web browsers. Their multi-model approach (CNN, LSTM, SVM, RF, DT, CNN-LSTM) dynamically selects the best-performing model for re-al-time phishing detection. The CNN-based model achieved 98.42% accuracy, showcasing DL's potential for real-time cybersecurity applications.

Hybrid models combine multiple feature extraction techniques, offering high classification accuracy, but their computational cost remains a challenge for real-time applications.

2.2.4 Comparative analysis of deep learning, machine learning, and ensemble learning approaches

Phishing detection models vary in effectiveness, with machine learning (ML), ensemble learning (EL), and deep learning (DL) techniques offering distinct trade-offs. Zara et al. [34] compared these methods using a Kaggle dataset of 11,055 websites with 32 attributes, applying feature selection techniques (IG, PCA, GR) for optimization. Among the models:

- Random Forest (RF) achieved the highest F1 score (99%), excelling in accuracy and efficiency.
- LSTM performed best among DL models (F1 score: 97.7%), capturing sequential URL patterns effectively.
- Decision Tree (DT) was the top ML model (F1 score: 97.7%), offering fast inference with lower computational cost.

These results highlight RF's efficiency, DL's pattern recognition capability, and ML's suitability for real-time detection. This study extends these findings by proposing a hybrid DL model that balances accuracy and computational efficiency for real-world applications.

### 2.3 Research gaps and limitations in existing studies

Despite significant advancements in DL-based phishing detection, several limitations remain, creating opportunities for further improvement:

- Limited dataset sizes: Many studies use small, balanced datasets, which may not capture the full diversity of real-world phishing attacks.
- Dependence on feature selection: Several approaches rely on manual feature engineering (e.g., SelectKBest, PCA, FSOR, chi-squared tests), limiting adaptability to evolving phishing strategies
- Computational overhead: Hybrid models require high processing power, making them less feasible for real-time deployment.
- Lack of real-time adaptability: While some studies propose browser-integrated detection systems, most DL models are not optimized for real-time phishing prevention

### 2.4 Contribution of this study

To address these challenges, this study proposes a deep learning model that directly processes raw URLs using character-level embedding, eliminating the need for feature selection or manual preprocessing. The proposed Parallel CNN-BiGRU archi-tecture captures both local feature patterns (via CNNs) and sequential dependencies (via BiGRUs), enhancing phishing detection accuracy while maintaining computational efficiency, making it suitable for real-time applications.

## 3. MATERIALS AND METHODS

This section outlines the methodology adopted to develop, train, and evaluate the deep learning (DL) model for phishing URL detection. The process involves dataset selection, preprocessing, model architecture design, training strategy, and performance evaluation. The proposed hybrid architecture integrates CNNs and a BiGRU to enhance feature

representation and detection accuracy. Figure 1 provides an overview of the methodological workflow.

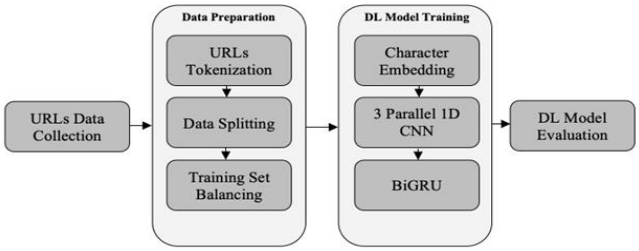


**Figure 1.** Proposed method for phishing detection

### 3.1 Dataset

The Malicious URLs Dataset [35], obtained from Kaggle, was used for training and evaluating the phishing detection model. This dataset comprises 651,191 URLs, categorized into 32,520 malware URLs, 94,111 phishing URLs, 96,457 defacement URLs, and 428,103 benign URLs. The data was sourced from five repositories to ensure a diverse representation of phishing and benign URLs. The ISCX-URL-2016 dataset provided the primary data, supplemented with phishing and malware samples from the Malware Domain Blacklist. Additional benign URLs were obtained from Faizan's GitHub repository, while the Phishtank and PhishStorm datasets contributed further phishing URLs.

Since the study focuses on phishing detection, only the phishing and benign classes were retained, reducing the dataset to 522,214 URLs. The dataset was further preprocessed to enhance model performance.

### 3.2 Data preprocessing

Several preprocessing techniques were applied to prepare the dataset for deep learning-based classification. First, non-relevant categories (malware and defacement) were removed, and phishing and benign labels were numerically encoded as 1 and 0, respectively. The primary feature used for model training was the raw URL itself.

To convert URLs into a machine-readable format, character-level tokenization was employed. This method treats each character in the URL as a unique token, ensuring that the model captures fine-grained structural patterns indicative of phishing attempts. A tokenizer was used to map characters to unique integer values, and sequences were padded to a uniform length of 200 characters to standardize input dimensions.

The dataset was divided into three subsets: 70% for training, 15% for validation, and 15% for testing. This 70-15-15 split was selected specifically because of the large size of the dataset used in this study. Allocating 70% of the data to training provided the model with sufficient data to learn meaningful and robust patterns. Meanwhile, setting aside 15% each for validation and testing enabled more comprehensive evaluation and hyperparameter tuning. This broader allocation for evaluation helped reduce overfitting and improved the model's ability to generalize, which is crucial for real-world phishing detection tasks. However, the improved performance observed in this study cannot be attributed solely to the choice of data split. Linh et al. [33], who used the same dataset, experimented with various split ratios such as 70:30 and 80:20. Despite trying these configurations, their models did not achieve the same level of performance as the one proposed in this study. This demonstrates that the improved results stem from the model architecture and training methodology, not simply from altering the data partition.

To maintain class distribution across all subsets, stratified sampling was used to split the dataset. However, an imbalance was observed, with phishing URLs significantly underrepresented. To address this, the Synthetic Minority Oversampling Technique (SMOTE) [36] was applied. SMOTE was preferred over other balancing methods such as random oversampling of minority class or undersampling of majority class because it generates new, synthetic samples rather than duplicating existing ones or discarding valuable data. This approach helped enrich the minority class, reduce overfitting, and improve the model's ability to learn more generalizable patterns without sacrificing data diversity or size.

### 3.3 Deep learning model architecture

The proposed phishing detection model, illustrated in Figure 2, employs a hybrid Parallel CNN-BiGRU architecture designed to capture both local and sequential patterns in phishing URLs.

(1) Character Embedding Layer: The input URLs, after tokenization, are passed through an embedding layer of dimension 64, transforming characters into dense vector representations that preserve semantic relationships.

(2) Parallel 1D CNN Layers: Three parallel 1D CNN layers with different kernel sizes of 3, 5, and 7 capture n-gram patterns of diverse lengths that are commonly found in phishing URLs. These varying kernel sizes allow for capturing a richer range of features.

a. 3-character kernel: captures short character patterns, such as prefixes or suspicious short tokens, which often appear at the beginning or end of phishing URLs and are usually indicators of phishing attempts.

b. 5-character kernel: captures mid-length substrings, like parts of legitimate brand names (e.g., "apple", "Gmail") being used deceptively or short word combinations designed to mislead users.

c. 7-character kernel: captures longer, high-level structural phishing-specific patterns.

Each CNN layer consists of 128 filters, followed by a global max-pooling layer, reducing feature maps to a compact vector representation.

(3) BiGRU Layer: The concatenated outputs from the CNN layers are passed into a Bidirectional GRU (BiGRU) layer with 128 neurons (64 per direction). The BiGRU enhances the model's ability to learn contextual dependencies within URL sequences.

(4) Fully Connected Layers: A 256-neuron dense layer, followed by dropout regularization (0.2), refines the extracted features before classification.

(5) Output Layer: A sigmoid activation function generates a probability score between 0 and 1, representing the likelihood of a URL being phished. A threshold of 0.5 is used for classification.

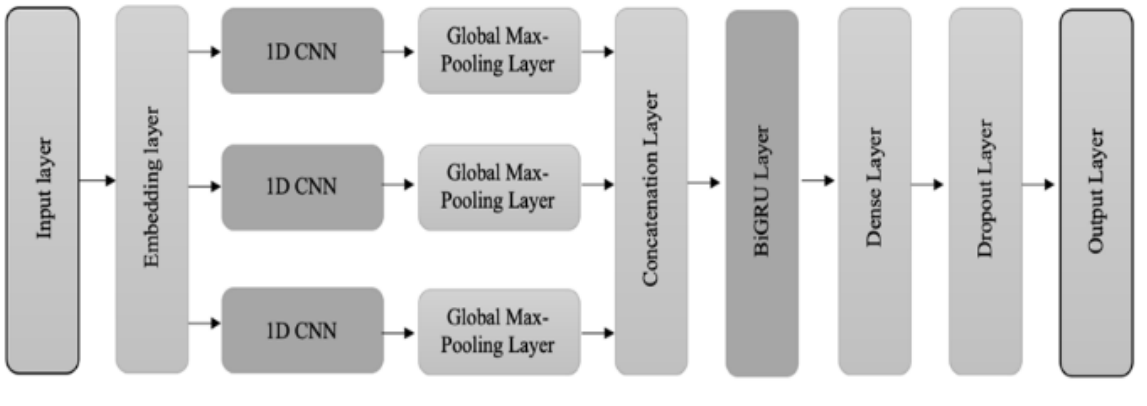


**Figure 2.** Deep learning model architecture

### 3.4 Model training strategy

The training of the phishing detection model was performed using Google Colab with V2-8 TPUs, ensuring efficient computation. The Adam optimizer was employed due to its adaptive learning rate properties, facilitating faster convergence. The initial learning rate was set to 0.001, with the ReduceLROnPlateau technique applied to decrease the learning rate dynamically if validation loss stagnated. This approach prevents overshooting the optimal parameters and stabilizes training. The binary cross-entropy loss function was used, as it is well-suited for binary classification tasks. The training was conducted using a batch size of 32 over a maximum of 20 epochs, with early stopping implemented to terminate training if no improvement in validation loss was observed for five consecutive epochs. The model checkpoint mechanism is used to enhance training efficiency, ensuring that the model maintains high accuracy.

### 3.5 Evaluation metrics

In this research, multiple metrics were used to evaluate the effectiveness and capability of the model, including accuracy, precision, recall, F1 test, and area under the curve (AUC CURVE). Accuracy is a general metric but may not be appropriate in the case of imbalanced data. Therefore, precision and recall were combined to evaluate the ability of the model to correctly identify fraudulent URLs while minimizing false positives and false negatives. The F1 score, which is the balance between precision and recall, is particularly critical in determining the reliability of the model in real-world applications where both types of errors have security implications.

The AUC CURVE score was also calculated to measure the ability of the model to distinguish between fraudulent and benign URLs across different classification thresholds.

The evaluation metrics were computed using the following formulas:

(1) Accuracy (ACC): Measures overall classification correctness.

$$Accuracy = \frac{TP+TN}{TP+TN+FN+FP} \quad (1)$$

(2) Precision (P): Indicates the proportion of correct phishing predictions.

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

(3) Recall (R): Evaluates how well the model captures phishing instances.

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

(4) F1-Score: A harmonic mean of Precision and Recall, balancing false positives and false negatives.

$$\text{F1} - \text{Score} = \frac{2 \times (Precision \times Recall)}{(Precision + Recall)} \quad (4)$$

(5) AUC CURVE (Area Under the Receiver Operating Characteristic Curve): Measures the model's ability to distinguish between phishing and benign URLs across different classification thresholds.

The evaluation metrics they were used in this research provide a comprehensive assessment of the model's strengths and weaknesses, proving it is reliable in re-al-world deployment scenarios.

## 4. RESULTS

In this section, we present the results of the proposed deep learning model. Assessing the model performance on a balanced dataset and using a combination of parallel CNNs and BiGRUs for feature extraction and sequence modeling, key performance metrics, like accuracy, precision, recall, F1-score, and the Area Under the Receiver Operating Characteristic (AUC CURVE) curve, are used to assess the model's efficacy.

### 4.1 Dataset distribution and preprocessing

The data set in this research comprised 365,550 URLs; 299,672 were benign (82%) and 65,878 were phishing URLs (18%). The Synthetic Minority Oversampling Technique (SMOTE) was applied to balance the dataset, resulting in 299,672 phishing URLs and an overall training set of 599,344 samples. Figure 3 and Figure 4 illustrate the class distributions before and after applying SMOTE, demonstrating the oversampling process's effectiveness in mitigating class imbalance.

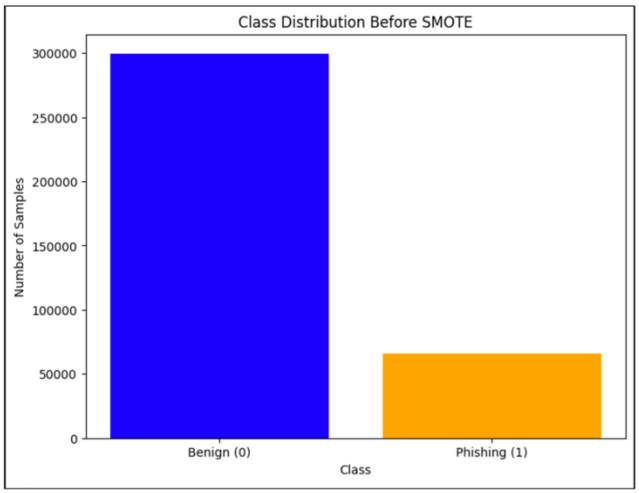


**Figure 3.** Training set before SMOTE

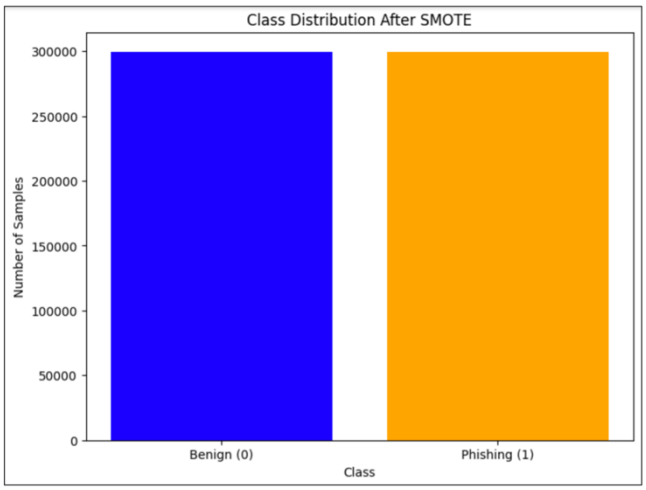


**Figure 4.** Training set after SMOTE

### 4.2 Model performance evaluation

The evaluation metrics of the proposed model for detecting phishing showed a high capability in classifying most URLs correctly. The testing set produced the following results:

- Accuracy: 98.46%
- Precision: 98.45%
- Recall: 98.46%
- F1-score: 98.45%
- AUC Curve Score: 0.9962

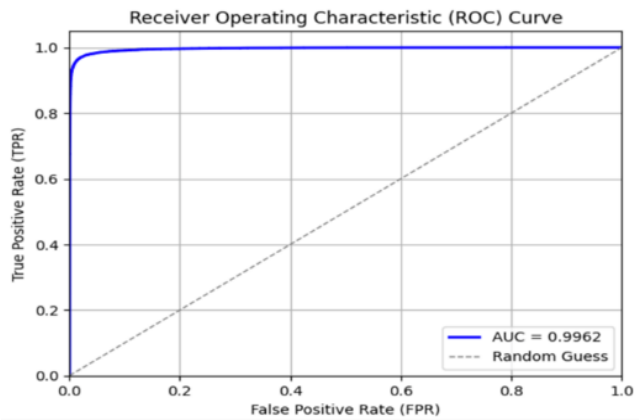


**Figure 5.** ROC curve

Figure 5 presents the ROC curve, which indicates the model's high discriminative power between phishing and benign URLs, with almost perfect classification accuracy (99.62%). The high performance is an indicator of low errors, with scarcely any false positives (valid URLs incorrectly labeled as phishing) or false negatives (actual phishing sites being ignored). High discriminability makes this model highly reliable for real-world applications in corporate security tools or browser anti-phishing plugins.

### 4.3 Confusion matrix analysis

The confusion matrix (Figure 6) gives insight into the model's classification efficiency:

- True Positives (TP): 13,408 phishing URLs correctly identified as phishing.
- True Negatives (TN): 63,716 benign URLs correctly classified.
- False Positives (FP): 500 benign URLs misclassified as phishing.
- False Negatives (FN): 709 phishing URLs misclassified as benign.

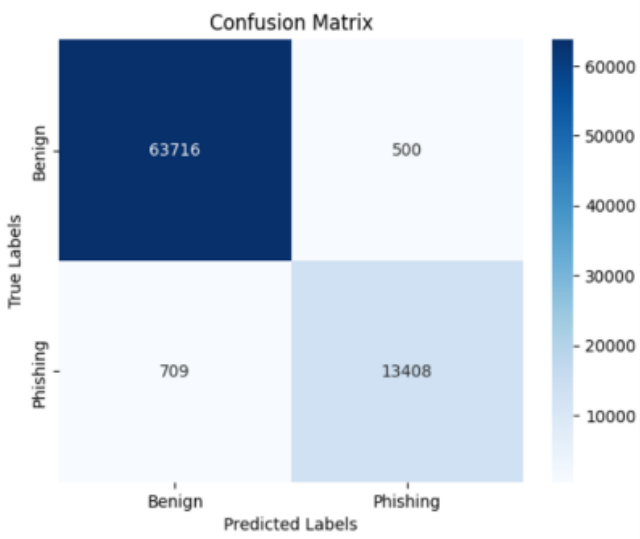


**Figure 6.** Confusion matrix

Although the false positive rate is low, reducing false negatives is important because undetected phishing URLs are a significant security risk

## 5. DISCUSSION

The performance results of this research justify the effectiveness of the proposed parallel CNN-BiGRU model in phishing URL detection with 98.46% accuracy, surpassing other state-of-the-art deep learning-based techniques. Through the fusion of character-level embeddings, parallel CNN layers, and BiGRU, the model can learn local textual patterns and long-range dependencies of URLs, respectively, hence being more robust to obfuscation techniques used by attackers. The ability of the proposed model to process raw URLs without requiring hand-engineered features makes it more flexible to adapt to new and future phishing techniques. To fully appreciate the applicability of these findings, this discussion reflects on the relative performance of the model compared to

prior studies, the impact of data balancing, error analysis, and its computational feasibility for real-world applications. Additionally, key limitations and future directions are outlined to highlight areas for improvement.

### 5.1 Comparative analysis with existing studies

To assess the effectiveness of the proposed approach, we compare its performance with prior phishing detection models. Table 2 provides an overview of existing deep learning-based phishing detection methods and their reported accuracy. Traditional CNN-based approaches, such as those presented in [29, 33], achieved 98.01% and 98.42% accuracy, respectively, whereas recurrent models like LSTM and BiGRU-based methods have demonstrated improved sequential learning capabilities but often struggle with feature extraction.

The superior accuracy (98.46%) of our model can be attributed to its hybrid architecture, which combines:

- Parallel CNNs for extracting multi-scale features (character n-grams, sub-strings, domain structures).
- BiGRU, which enables bidirectional sequence learning to detect phishing-indicative patterns regardless of their position in the URL.
- Character-level embeddings, which eliminate the need for handcrafted feature engineering, enhancing scalability and adaptability:

**Table 2.** Comparison with similar studies

| Study | Approach | ACC | Precision | Recall | F1 | AUC |
|---|---|---|---|---|---|---|
| Korkmaz et al. [28] | Character Embedding + CNN | 97.17% | NA | NA | NA | NA |
| Aldakheel et al. [29] | Character Embedding + CNN | 98.01% | NA | NA | NA | 100% |
| Linh et al. [33] | Character Embedding + CNN | 98.42% | 98.40% | 98.43% | NA | NA |
| Proposed Model | Character Embedding + Parallel CNN + BiGRU | 98.46% | 98.45% | 98.46% | 98.45% | 99.62% |

### 5.2 Impact of data balancing on model performance

Phishing URL detection presents a significant class imbalance challenge, as benign URLs vastly outnumber phishing URLs in real-world datasets. This study addressed the issue using SMOTE, an oversampling technique that generates synthetic phishing samples, thereby improving class distribution. As illustrated in Figures 3 and 4, applying SMOTE resulted in a more balanced dataset, which in turn enhanced the model's ability to learn phishing patterns effectively.

The effect of SMOTE is evident in the model's precision-recall trade-off. Without balancing, the model exhibited a higher false negative rate, where phishing URLs were misclassified as benign. After balancing, the model maintained a high recall (98.46%), ensuring that a greater proportion of phishing URLs were correctly detected. The ability to detect phishing URLs without significantly increasing false positives is crucial for real-world deployment, as excessive false alarms can reduce user trust in cybersecurity systems.

### 5.3 Error analysis and model limitations

Despite its strong performance, the model is not without limitations. A detailed error analysis highlights two key challenges:

- False negatives (misclassified phishing URLs): The model failed to detect 709 phishing URLs, which could pose a security risk in practical scenarios.
- False positives (misclassified benign URLs): While the false positive rate was relatively low (0.78%), the incorrect classification of 500 legitimate URLs as phishing could lead to unnecessary blocking of harmless websites.

The misclassification errors suggest potential areas for refinement. One approach to reduce false negatives could involve attention mechanisms that prioritize high-risk URL segments. Similarly, reducing false positives may require integrating external contextual features, such as domain registration data or WHOIS lookup information.

### 5.4 Computational efficiency and real-world applicability

While deep learning models provide high accuracy, their practical deployment depends on computational efficiency. The proposed Parallel CNN-BiGRU model achieves an effective balance between performance and resource usage, containing approximately 343,297 trainable parameters and requiring only 1.31 MB of memory. This relatively small model size reflects a lightweight architecture that is computationally efficient and well-suited for real-world applications, including environments with limited hardware resources.

Nevertheless, certain components of the architecture, such as multiple convolutional layers and bidirectional GRU units, do introduce additional processing overhead and longer training times, particularly when dealing with large-scale datasets. While the model performs well under current settings, future optimization strategies such as model pruning, quantization, or knowledge distillation could further reduce latency and memory usage without compromising accuracy.

Furthermore, alternative transformer-based architectures such as DistilBERT, ALBERT, TinyBERT, or MobileBERT may offer superior phishing URL classification while improving inference speed.

### 5.5 Future research directions

To further improve phishing URL detection, several directions for future research are proposed:

- Dynamic learning models: Implementing continuous learning approaches that adapt to new phishing attack trends over time.
- Hybrid feature extraction: Combining deep learning-based models with rule-based heuristics or graph-based URL analysis.
- Real-time phishing prevention systems: Deploying lightweight models in web browsers, email filters, and cybersecurity platforms for real-time URL analysis.
- Dataset expansion and generalization: Utilizing larger and more diverse datasets from multiple sources to ensure robustness against adversarial phishing techniques.

## 6. CONCLUSIONS

In this study, a deep learning-based phishing URL detection model was developed to address the limitations of traditional detection methods and existing machine learning approaches. The proposed model leverages character-level embeddings, Parallel CNN layers, and a BiGRU to process raw URLs directly, eliminating the need for handcrafted feature extraction. The Parallel CNN layers effectively capture local character n-grams and structural patterns, while the BiGRU component models long-range dependencies, enabling a more comprehensive analysis of phishing URLs.

For enhanced model robustness, character tokenization was employed, and Synthetic Minority Oversampling Technique (SMOTE) was employed to address class imbalance in the dataset. The model was trained on 599,344 URLs and evaluated on a test set, with an accuracy of 98.46%, precision of 98.45%, recall of 98.46%, and a 98.45% F1-score, outperforming several other current phishing detection frameworks. The results indicate that the hybrid parallel CNN-BiGRU architecture is effective in accurately distinguishing phishing URLs from genuine ones.

Even though the model shows high performance, future research must focus on hyperparameter optimization to improve detection accuracy and efficiency further. Besides, considering even more sophisticated deep models such as transformer-based models (e.g., BERT or GPT) would likely enable greater contextual reasoning and learning capacity in adaptive phishing attacks. Further research into the application of real-time deployment as well as in optimization of computational resources will be also crucial in integrating this strategy into a large-scale cyber-security application.

This research is a valuable contribution to phishing detection because it introduces an end-to-end, scalable deep learning system that strengthens cybersecurity defense against constantly evolving phishing attacks.

## REFERENCES


[1] APWG. (2024). Phishing Activity Trends Report: 4th Quarter 2023. https://docs.apwg.org/reports/apwg_trends_report_q4_2023.pdf.

[2] APWG. APWG Q4 Report Finds 2023 Was Record Year for Phishing. https://apwg.org/apwg-q4-report-finds-2023-was-record-year-for-phishing/.

[3] Pourmohamad, R., Wirsz, S., Oest, A., Bao, T., Shoshitaishvili, Y., Wang, R., Bazzi, R.A. (2024). Deep dive into client-side anti-phishing: A longitudinal study bridging academia and industry. In Proceedings of the 19th ACM Asia Conference on Computer and Communications Security, pp. 638-653. https://doi.org/10.1145/3634737.3657027

[4] Elberri, M.A., Tokeşer, Ü., Rahebi, J., Lopez-Guede, J. M. (2024). A cyber defense system against phishing attacks with deep learning game theory and LSTM-CNN with African vulture optimization algorithm (AVOA). International Journal of Information Security, 23(4): 2583-2606. https://doi.org/10.1007/s10207-024-00851-x

[5] Mwavali, A. (2024). Combating phishing in Kenya: A supervised learning model for enhanced email security in Kenyan financial institutions. International Journal of Technology and Systems, 9(4): 23-36.

[6] Hong, J. (2012). The state of phishing attacks. Communications of the ACM, 55(1): 74-81. https://doi.org/10.1145/2063176.2063197

[7] Nikitha, A, Rakshitha, S.P., Akhila, P., Kavyasree, K. (2025). Asatyajaal anveshak. International Journal for Research in Applied Science and Engineering Technology (IJRASET), 133(1): 1709-1717. https://doi.org/10.22214/ijraset.2025.66610

[8] Gomathi, K. (2013). A cloud-based AI way to deal with phishing URL location. International Journal for Research in Applied Science and Engineering Technology. https://doi.org/10.22214/ijraset.2023.50041

[9] Jadhav, A., Chandre, P.R. (2025). Survey and comparative analysis of phishing detection techniques: current trends, challenges, and future directions. IAES International Journal of Artificial Intelligence, 14(2): 853-866. https://doi.org/10.11591/ijai.v14.i2.pp853-866

[10] Abdolrazzagh-Nezhad, M., Langarib, N. (2025). Phishing detection techniques: A review. Data Science: Journal of Computing and Applied Informatics, 9(1): 32-46.

[11] Ogunleye, G., Olukoya, B.M., Olusesi, A.T., Olabisi, P., Sodipo, Q.B., Adekunle, O. (2023). Heterogeneous ensemble feature selection and multilevel ensemble approach to machine learning phishing attack detection. FUOYE Journal of Engineering and Technology, 8(4): 438-447. http://doi.org/10.46792/fuoyejet.v8i4.1105

[12] Alazaidah, R., Al-Shaikh, A., Al-Mousa, M.R., Khafajah, H., Samara, G., Alzyoud, M., Almatarneh, S. (2024). Website phishing detection using machine learning techniques. Journal of Statistics Applications & Probability, 13(1): 119-129. http://doi.org/10.18576/jsap/130108

[13] Daniel, M.A., Chong, S.C., Chong, L.Y., Wee, K.K. (2025). Optimising phishing detection: A comparative analysis of machine learning methods with feature selection. Journal of Informatics and Web Engineering, 4(1): 200-212. https://doi.org/10.33093/jiwe.2025.4.1.15

[14] Murad, S.A., Rahimi, N., Muzahid, A.J.M. (2023). PhishGuard: Machine learning-powered phishing URL detection. In 2023 Congress in Computer Science, Computer Engineering, & Applied Computing (CSCE), Las Vegas, NV, USA, pp. 2279-2284. https://doi.org/10.1109/CSCE60160.2023.00371

[15] Kumar, G., S, K. (2024). Comprehensive review on an advanced machine learning approach for enhancing phishing website detection. International Journal for Research in Applied Science and Engineering Technology.

[16] Suresh, N., Kumar, U.S. (2024). PHISHSNAP-A chrome extension tool used for detection of phishing applying machine learning. Journal of Artificial Intelligence and Capsule Networks, 6(1): 105-121. https://doi.org/10.36548/jaicn.2024.1.008

[17] Jin, Y., Yu, X., Gao, Y. (2023). Multiclass malicious URL attack type detection via capsule-based neural network. In Third International Seminar on Artificial Intelligence, Networking, and Information Technology (AINIT 2022), 12587: 520-525. https://doi.org/10.1117/12.2667245

[18] Yuan, H., Yang, Z., Chen, X., Li, Y., Liu, W. (2018). URL2Vec: URL modeling with character embeddings for fast and accurate phishing website detection. In 2018 IEEE Intl Conf on Parallel & Distributed Processing with

Applications, Ubiquitous Computing & Communications, Big Data & Cloud Computing, Social Computing & Networking, Sustainable Computing & Communications, Melbourne, VIC, Australia, pp. 265-272. https://doi.org/10.1109/BDCloud.2018.00050

[19] Mangalam, K., Subba, B. (2024). PhishDetect: A BiLSTM based phishing URL detection framework using FastText embeddings. In 2024 16th International Conference on COMmunication Systems & NETworkS (COMSNETS), Bengaluru, India, pp. 637-641. https://doi.org/10.1109/COMSNETS59351.2024.10427067

[20] Amanullah, M., Selvakumar, V., Jyot, A., Purohit, N., Fahlevi, M. (2022). CNN based prediction analysis for web phishing prevention. In 2022 International Conference on Edge Computing and Applications (ICECAA), Tamilnadu, India, pp. 1-7. https://doi.org/10.1109/ICECAA55415.2022.9936112

[21] Sawant, S., Savakhande, R., Sankhe, O., Tamboli, S. (2024). Phishing detection by integrating machine learning and deep learning. In 2024 11th International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India, pp. 1078-1083. 10.23919/INDIACom61295.2024.10499100

[22] Zhang, G., Luo, Y., Xie, H., Dai, Z. (2024). Crispr-SGRU: Prediction of CRISPR/Cas9 off-target activities with mismatches and indels using stacked BiGRU. International Journal of Molecular Sciences, 25(20): 10945. https://doi.org/10.3390/ijms252010945

[23] Owoh, N., Adejoh, J., Hosseinzadeh, S., Ashawa, M., Osamor, J., Qureshi, A. (2024). Malware detection based on API call sequence analysis: A gated recurrent unit–generative adversarial network model approach. Future Internet, 16(10): 369. https://doi.org/10.3390/fi16100369

[24] Shaikh, Z.M., Ramadass, S. (2024). Unveiling deep learning powers: LSTM, BiLSTM, GRU, BiGRU, RNN comparison. Indonesian Journal of Electrical Engineering and Computer Science, 35(1): 263-273.

[25] Giustino, J.K., Santosa, Y.P. (2024). Toxic comment classification comparison between LSTM, BiLSTM, GRU, and BiGRU. Proxies: Jurnal Informatika, 7(2): 115-127. https://doi.org/10.24167/proxies.v7i2.12471

[26] Benavides-Astudillo, E., Fuertes, W., Sanchez-Gordon, S., Nuñez-Agurto, D., Rodríguez-Galán, G. (2023). A phishing-attack-detection model using natural language processing and deep learning. Applied Sciences, 13(9), 5275. https://doi.org/10.3390/app13095275

[27] Sakthipriya, N., Govindasamy, V., Abhinesh, C., Krishna Govarthini, T., Rajesh, R., Swedha, R. (2024) Enhancing phishing detection: A hybrid deep learning model integrating Bi-LSTM and Bi-GRU algorithms for URL and content analysis. International Education and Research Journal, 10(3). https://doi.org/10.21276/IERJ24354343237105

[28] Korkmaz, M., Kocyigit, E., Sahingoz, O., Diri, B. (2022). A hybrid phishing detection system using deep learning-based URL and content analysis. Elektronika ir Elektrotechnika, 28(5).

[29] Aldakheel, E.A., Zakariah, M., Gashgari, G.A., Almarshad, F.A., Alzahrani, A.I. (2023). A deep learning-based innovative technique for phishing detection in modern security with uniform resource locators. Sensors, 23(9), 4403. https://doi.org/10.3390/s23094403

[30] Alshingiti, Z., Alaqel, R., Al-Muhtadi, J., Haq, Q.E.U., Saleem, K., Faheem, M.H. (2023). A deep learning-based phishing detection system using CNN, LSTM, and LSTM-CNN. Electronics, 12(1): 232. https://doi.org/10.3390/electronics12010232

[31] Driss, A.I.T., Zougagh, H. (2024). Improving online security: A deep learning model for phishing URL detection. https://doi.org/10.21203/rs.3.rs-5363511/v1

[32] Alsubaei, F.S., Almazroi, A.A., Ayub, N. (2024). Enhancing phishing detection: A novel hybrid deep learning framework for cybercrime forensics. IEEE Access, 12: 8373-8389. https://doi.org/10.1109/ACCESS.2024.3351946

[33] Linh, D.M., Hung, H.D., Chau, H.M., Vu, Q.S., Tran, T.N. (2024). Real-time phishing detection using deep learning methods by extensions. International Journal of Electrical and Computer Engineering (IJECE), 14(3): 3021-3035.

[34] Zara, U., Ayub, K., Khan, H.U., Daud, A., Alsahfi, T., Gulzar, S. (2024). Phishing website detection using deep learning models. IEEE Access, 14(3): 3021-3035. https://doi.org/10.1109/ACCESS.2024.3486462

[35] Siddhartha, M., Malicious URLs dataset. Kaggle. https://www.kaggle.com/datasets/sid321axn/malicious-urls-dataset.

[36] Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P. (2002). SMOTE: Synthetic minority over-sampling technique. Journal of Artificial Intelligence Research, 16: 321-357. https://doi.org/10.1613/jair.953