Available online at www.qu.edu.iq/journalcm
JOURNAL OF AL-QADISIYAH FOR COMPUTER SCIENCE AND MATHEMATICS
ISSN:2521-3504(online) ISSN:2074-0204(print)

Enhanced Phishing URL Identification through Recurrent Neural Networks: A Comparative Study of LSTM and BiLSTM

Wadhah Sata Kathum Ajjam*[*,a], Abdullahi Abdu Ibrahim[b]

[a] *Altinbas University, Department of Information Technologies, wadhah.ajjam522@gmail.com*

[b] *Altinbas University, Faculty of Engineering and Architecture, abdullahi.ibrahim@altinbas.edu.tr*

ARTICLE INFO

Article history:
Received: 04/07/2025
Rrevised form: 26/07/2025
Accepted : 30/07/2025
Available online: 30/09/2025

Keywords:

Phishing Detection, Recurrent Neural Networks (RNN), URL Classification

ABSTRACT

Phishing attacks continue to pose a serious threat to cybersecurity, underscoring the need for effective and scalable detection methods. This study evaluates the performance of Recurrent Neural Network (RNN) architectures—specifically Long Short-Term Memory (LSTM) and Bidirectional LSTM (BiLSTM)—for detecting phishing websites based on the sequential patterns in URL structures and webpage content. The LSTM model achieved an Area Under the Curve (AUC) of 0.92, with an overall accuracy of 98.4%, precision of 98.9%, and recall of 97.1%. These results indicate a strong ability to identify phishing URLs with a low false positive rate, although performance declined when detecting sophisticated or zero-day phishing attempts. The BiLSTM model, which incorporates bidirectional context, achieved a higher AUC of 0.95 and improved precision of 91% at a recall of 89%. However, it exhibited a slightly lower overall accuracy of 97.9% and a higher false negative rate. Both models effectively differentiated phishing from legitimate URLs, with BiLSTM offering improved context awareness but at the cost of reduced recall. The results suggest that while BiLSTM enhances contextual understanding, the LSTM model offers better generalization and computational efficiency for real-time deployment. This work highlights the potential of RNN-based models in phishing detection and the importance of balancing sensitivity and specificity in cybersecurity applications.

https://doi.org/10.29304/jqcsm.2025.17.32380

1. Introduction

Malicious URLs pose a significant cybersecurity threat by facilitating phishing, malware distribution, and data breaches. Serving as global addresses for online resources, URLs are increasingly exploited to deliver harmful content. According to Google data from February 2019, approximately 1,300,000 malicious URLs are banned daily, underscoring the scale of this issue [1]. The rapid proliferation of phishing URLs presents serious risks to online security, necessitating the development of efficient real-time detection mechanisms. Traditional defenses such as blacklists and whitelists are limited by their inability to maintain comprehensive and current repositories of malicious URLs, especially given attackers' use of sophisticated obfuscation techniques to disguise malicious URLs as legitimate ones [1]. Automated, accurate, and scalable tools are therefore essential for timely identification of emerging threats.

*Corresponding author: Wadhah Sata Kathum Ajjam

Email addresses: *wadhah.ajjam522@gmail.com*

Communicated by 'sub etitor'

Both academic and industry efforts have resulted in the integration of anti-phishing solutions into major web browsers including Google Chrome, Firefox, Safari, Web Pilgrim, and Edge. However, phishing attacks continue to rise, as reported by the Anti-Phishing Working Group (APWG) [2], highlighting the persistent need for innovative countermeasures. Recurrent Neural Networks (RNNs), particularly Long Short-Term Memory (LSTM) and Bidirectional LSTM (BiLSTM) models, have demonstrated effectiveness in capturing sequential patterns inherent in URLs. These architectures can learn temporal dependencies and contextual cues within URL strings, improving detection of obfuscated and novel phishing URLs. Their ability to capture long-range and bidirectional dependencies renders them well-suited for real-time phishing URL detection systems [1].

This work proposes LSTM and BiLSTM-based models for phishing URL detection and demonstrates superior accuracy and robustness compared to existing approaches. The models leverage deep sequential learning to enhance cybersecurity defenses against evolving online threats.

2. Related Works

Combating the widespread threat of phishing across networked and internet-based systems has been a central concern for cybersecurity researchers. Their efforts have primarily focused on developing robust defense mechanisms to protect users from digital threats, with particular attention given to detecting phishing through malicious URLs. Historically, phishing detection strategies have fallen into two main categories: list-based and machine learning–based approaches. List-based methods rely on maintaining blacklists of malicious URLs and whitelists of trusted domains, offering protection through known patterns. In contrast, machine learning–driven solutions utilize sophisticated algorithms—spanning both traditional machine learning and deep learning techniques—to identify subtle behavioral patterns and anomalies that signal phishing attempts, enabling more dynamic and adaptive threat detection [3].

Whitelist and blacklist-based phishing detection systems serve as foundational tools for differentiating between legitimate and malicious web pages. Whitelist-oriented solutions focus on maintaining a curated set of verified, secure websites, offering users a reliable reference for safe browsing. In one method, the whitelist is constructed by tracking and logging the IP addresses of websites that present user login forms, thus identifying potentially sensitive interactions. Another system enhances this approach by periodically updating the whitelist, enabling real-time alerts to warn users about emerging phishing threats. This technique relies on analyzing specific attributes derived from the correlation between website source code and modules that match domain-associated IP addresses. Preliminary results from such systems have shown a true positive rate of 86.02% and a false negative rate of just 1.48%, indicating promising detection accuracy [4]. Blacklists are a fundamental component of phishing defense strategies, functioning by aggregating databases of confirmed malicious URLs sourced from user reports, spam filters, and external cybersecurity authorities. By referencing these lists, detection systems can effectively block access to previously identified phishing domains and IP addresses, thereby forcing attackers to continuously rotate their infrastructure to evade detection. Blacklists are kept up to date either through automated updates or manual downloads, ensuring ongoing protection against known threats.

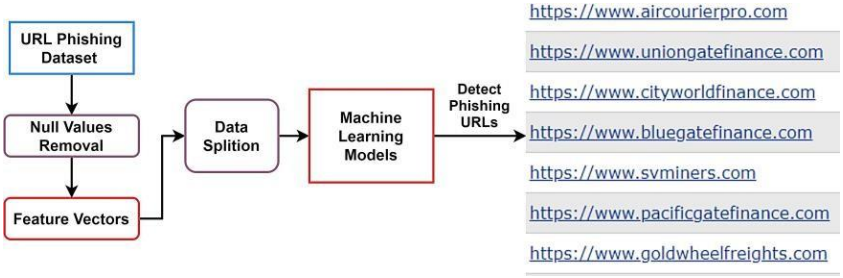
Although highly effective against known attacks, blacklist-based systems face limitations when dealing with zero-day phishing attempts, as such threats have not yet been recorded in the database. Nevertheless, these systems tend to produce fewer false positives compared to machine learning–driven methods. Tools like PhishNet and the Google Safe Browsing API exemplify the practical effectiveness of blacklist-based detection, with reported detection accuracies around 20%, as noted in studies [5] and [6]. These systems rely on approximate string matching algorithms to identify suspicious URLs by comparing them against blacklist entries. Despite their success, the continuous expansion of blacklist databases presents a scalability challenge, necessitating efficient infrastructure to maintain performance and reliability [7, [8].

Recent research has introduced a wide array of phishing detection methodologies, many of which report high levels of accuracy across various deployment contexts. In [9], a browser extension–based solution was proposed, achieving an 85% detection accuracy by monitoring user interaction in real time. Study [10] advanced automatic phishing detection by analyzing characteristics of shortened URLs, resulting in a 92% accuracy rate. Delta Phish [11], which employed supervised learning models based on URL features, demonstrated accuracy exceeding 70%. In [12], the Phish-Safe framework utilized a combination of Support Vector Machines (SVM) and Naive Bayes

classifiers, achieving 90% accuracy in phishing detection. Furthermore, [13] leveraged ensemble learning methods for email-based phishing detection, applying feature selection techniques to achieve an impressive 99% accuracy. The Phi DMA method, introduced in [14], combined URL features, lexical analysis, and whitelist verification to reach a 92% detection rate. In [15], an SVM-based system trained on six domain-specific attributes demonstrated a precision of 95%. Additionally, the approach outlined in [16], which integrated error-based heuristics and phoneme analysis, nearly achieved perfect detection performance, with accuracy approaching 100%. Gopal et al. [21] proposed a hybrid deep learning-based defense mechanism to detect phishing attacks, particularly in IoT environments where device control is often URL-dependent. Their approach integrates autoencoders for dimensionality reduction with a deep neural network (DNN) classifier to effectively distinguish phishing from legitimate URLs. The system was evaluated on multiple publicly available datasets, including OpenPhish, UCI, Mendeley, and PhishTank, achieving 92.89% accuracy, 93.07% recall, 92.75% precision, and an F1-score of 92.21%. While effective, the model underperforms slightly compared to recent RNN-based approaches and does not explicitly address zero-day attacks or real-time deployment challenges. Bozkir et al. [22] introduced *GramBeddings*, a novel neural architecture for phishing URL detection that employs dynamically generated n-gram embeddings without the need for pre-training or handcrafted features. Their architecture integrates CNN, LSTM, and attention layers in a multi-channel design and introduces an automated n-gram selection and filtering mechanism. Tested on a newly constructed dataset of 800K phishing and legitimate URLs, the model achieved 98.27% accuracy and demonstrated superior performance across several benchmarks. The study also assessed the model's robustness against real-world adversarial attacks, further highlighting its practical applicability. Collectively, these studies underscore the effectiveness of diverse and layered strategies in phishing detection, reflecting a growing trend toward hybrid and deep learning-based approaches that emphasize adaptability, real-time inference, and resilience against evolving threats..

3. Proposed System

This approach focuses on creating a machine learning model, specifically a neural network, designed to tell whether a domain name is a phishing attempt or safe to use. The process starts by collecting two sets of data: one with known phishing domain names (from PhishTank) and another with confirmed safe (benign) domain names (from Zenodo). Because there are usually far more benign examples than phishing ones, these datasets are combined, and the safe domain samples are slightly increased using a technique called oversampling. This balancing act ensures the model learns from a dataset that's fair, or only slightly favoring one type, which helps improve how well it performs. Next, each domain name is treated as a sequence of individual characters. Every unique character found across all the domain names is assigned a special code, essentially building a character dictionary. Then, each domain name is converted into a series of numbers, where each number represents the corresponding character from that dictionary. Since domain names vary in length, these number sequences are adjusted (padded) to be all the same length, creating neat, uniform inputs that the neural network can easily process. The neural network is then trained using these fixed-length number sequences, learning the patterns that help it distinguish between phishing and legitimate domains. After training, a custom evaluation is done to see how well the model works. This involves checking the model's predictions against a specific standard or threshold. The evaluation calculates important metrics like true positives, true negatives, false positives, and false negatives. It also involves looking closely at how the predictions are distributed for each category, which helps make the model's decisions easier to understand and



diagnose any potential issues.

Fig. 1- Detection of Phishing URLs And Structure of Proposed Approach

The studies were carried out utilizing a phishing dataset represented as data vectors, necessitating the elimination of null values to avoid superfluous empty entries. Functional features were utilized for classification, and the experimental setup included cross-validation techniques along with grid search hyper-parameter tuning. The process incorporated canopy feature selection to further refine and enhance the predictive performance of the system. The performance evaluation included accuracy, precision, recall, specificity, and F1-score metrics. The goal is to contribute to cybersecurity by effectively classifying phishing URLs and preventing attackers from breaching networks and accessing confidential content.

3.1. Data collection

The phishing URL dataset was compiled from multiple sources to ensure a diverse and balanced representation of phishing characteristics. Emphasis was placed on incorporating up-to-date URLs to reflect the latest phishing strategies and trends.

PhishTank [20] is a collaborative platform that enables users to submit, verify, and share phishing data. It maintains a publicly accessible database of verified phishing websites and provides an API for programmatic access to check URLs against their repository. The platform offers a phishing archive accessible via its website, with a CSV feed updated every 90 minutes containing URLs reported within the past thirty days. The database currently comprises over three million entries, supporting complex queries through the API Feed. Additionally, the Public Dashboard feature allows for targeted searches of specific IP addresses, hosts, domains, or full URLs. PhishTank also provides statistical summaries of various attributes related to detected phishing attacks, which can facilitate in-depth analysis of phishing patterns.

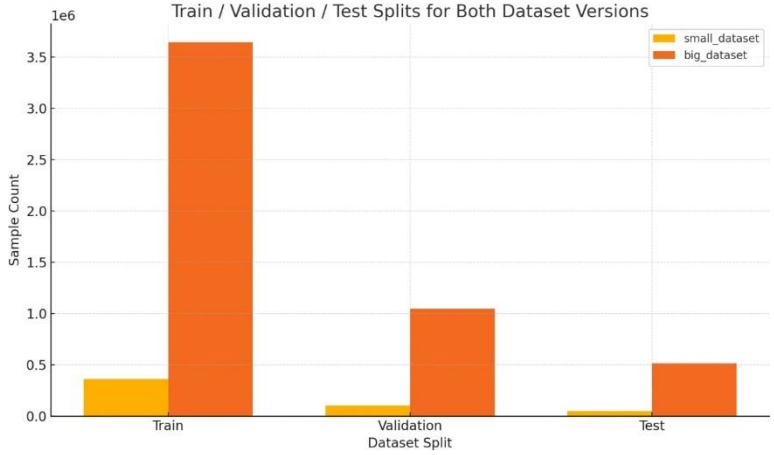


Fig. 2- Train / Validation / Test Splits For Both Dataset Versions

The dataset used in this study was provided in two versions to accommodate reproducibility and hardware constraints: a smaller 10% sub-sample (small_dataset) and the full dataset (big_dataset)(See Fig.2). The small_dataset contains 520,285 samples, divided into 364,199 for training, 104,576 for validation, and 51,510 for testing. The big_dataset, which was used for all experimentation in this study, includes a total of 5,202,841 samples, split into 3,641,986 for training, 1,045,774 for validation, and 515,080 for testing. Both versions include a mix of phishing URLs (sourced from Phishtank) and legitimate URLs (sourced from CommonCrawl), with class balance maintained proportionally across all splits.

3.2. Pre-processing

Data preprocessing begins with loading two CSV files containing phishing and benign domain names. First, we start by loading two CSV files. One has domain names identified as phishing, and the other has benign (safe) domain names. To make sure our model learns effectively from both types, we handle the imbalance – if there are way more of one kind than the other. We do this by randomly picking more examples from the benign category based on a set ratio, creating a combined list where each type is represented fairly. We label the phishing domains as 1 and the benign ones as 0. We also assign special 'weights' to each example to ensure that both classes contribute equally

during training. Next, we convert the domain names into a numerical format. We create a list of every unique character we find across all domain names, which acts like a dictionary. Each character in this dictionary gets assigned a unique number. Then, every domain name is transformed into a sequence of these numbers, one for each character. This turns the text data into numbers that our computer can understand. Since domain names are different lengths, we add a space character to our dictionary to handle shorter ones and then make sure every sequence of numbers is exactly 40 characters long. If a domain name is shorter, we pad it with the number representing our space character. If it's longer, we cut it down to 40 characters. This ensures all our input data is the same size, which is necessary for feeding batches of data into a neural network. Finally, we divide our complete dataset into two parts: one for training the model and another for testing it afterwards. We make sure that the labels (phishing or benign) and the sample weights travel with their respective data, keeping everything organized and ready for the model training and evaluation stages.

3.3 Deep Learning Models

3.3.1 LSTM Model

The LSTM model employed for phishing detection is a type of recurrent neural network designed to process sequential data. The input to the model consists of domain names, which are first encoded at the character level. These encoded characters are then mapped into dense, continuous representations via an embedding layer that outputs 64-dimensional vectors. This embedding facilitates the model's ability to capture relationships and patterns between characters (see Fig. 3). Following the embedding layer, a Long Short-Term Memory (LSTM) layer with 128 units processes the sequence. This layer is responsible for capturing temporal dependencies and contextual information within the domain name, enabling the identification of subtle patterns indicative of phishing. Mathematically, the LSTM cell at each time step t computes hidden state h_t and cell state c_t using the following equations:

$$f_t = \sigma\left(W_f.[h_{t-1}, x_t] + b_f\right) \quad (1)$$

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

$$\tilde{c}_t = tanh(W_c.[h_{t-1}, x_t] + b_c) \quad (3)$$

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

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

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

where σ is the sigmoid activation function, $\odot$ denotes element-wise multiplication, and W and bbb are weights and biases learned during training. The final LSTM output is fed into a dense layer with 128 units and ReLU activation, followed by a sigmoid-activated output layer producing the probability of a domain being phishing. The model is trained using binary cross-entropy loss and optimized with Adam, effectively learning complex temporal patterns within domain character sequences to accurately distinguish phishing attempts.

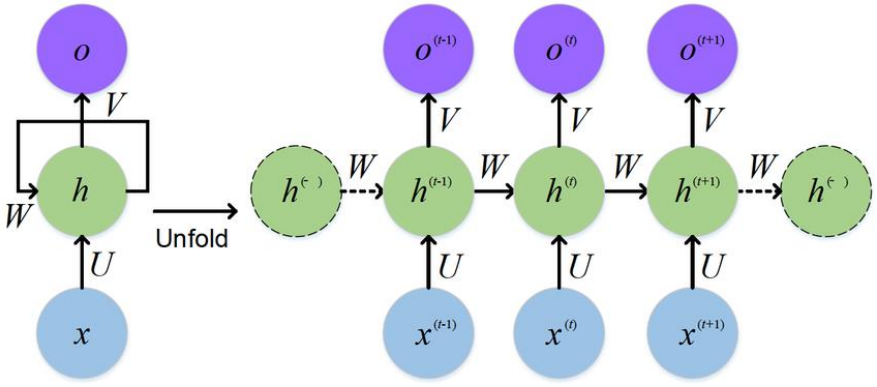


Fig. 3- LSTM Model Architecture

3.3.2 BiLSTM Model

In this study, a Bidirectional Long Short-Term Memory (BiLSTM) model was utilized for phishing URL detection to improve the capture of contextual information from both preceding and succeeding tokens within a sequence (see Fig. 4). The architecture begins with an embedding layer that transforms input tokens into dense vector representations. This is followed by a Bidirectional LSTM layer comprising 256 units, which processes the input sequence in both forward and backward directions concurrently. Subsequent fully connected dense layers enable the model to learn complex feature interactions, culminating in a binary classification output that discriminates between phishing and legitimate URLs. Mathematically, the BiLSTM at each timestep t computes a forward hidden state $\vec{h}_t$ and a backward hidden state $\overleftarrow{h}_t$ as:

$$\vec{h}_t = LSTM(x_t, \vec{h}_{t-1}), \quad \overleftarrow{h}_t = LSTM(x_t, \overleftarrow{h}_{t-1}) \tag{7}$$

The final hidden state is the concatenation:

$$h_t = [\vec{h}_t; \overleftarrow{h}_t] \tag{8}$$

Each LSTM cell internally updates its states using the following equations for forget gate f_t, input gate i_t, candidate cell state c_t, cell state $\tilde{c}_t$, output gate o_t, and hidden state h_t:(Eq(1) to (6)

where σ is the sigmoid activation, ⊙ denotes element-wise multiplication, and W and W are weight matrices and biases learned during training. The BiLSTM's bidirectional context combined with class weighting and early stopping techniques helped improve phishing detection accuracy by capturing intricate sequential patterns in URLs or text data.

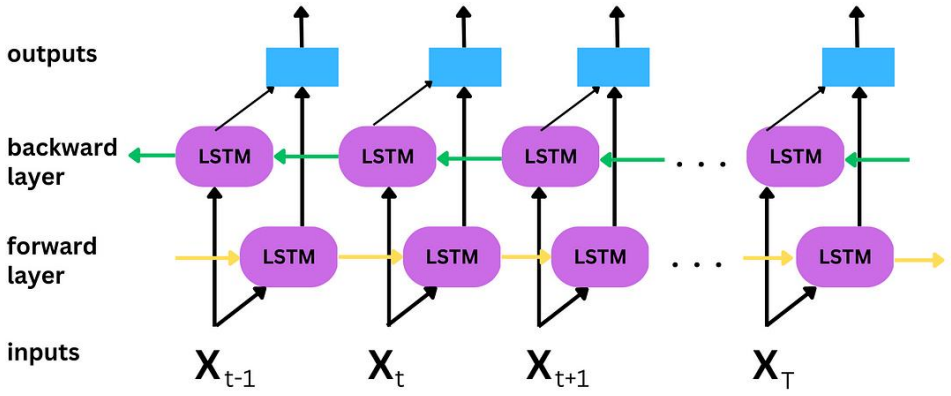


Fig. 4-BiLSTM Model Architecture

3.3. Performance Criteria

The performance criteria used in this paper to evaluate the phishing detection model primarily include accuracy, precision, recall, and F1-score. Accuracy measures the overall correctness of the model by calculating the ratio of correctly predicted instances (both phishing and legitimate) to the total number of instances. However, since phishing detection is often an imbalanced classification problem, accuracy alone may be misleading. Therefore, precision is used to measure the proportion of correctly identified phishing instances out of all instances predicted as phishing, reflecting the model's ability to avoid false positives. Recall (or sensitivity) evaluates the proportion of actual phishing instances that the model successfully detects, highlighting its ability to minimize false negatives. Finally, the F1-score serves as the harmonic mean of precision and recall, providing a balanced measure that accounts for both false positives and false negatives. These metrics together offer a comprehensive evaluation of the model's performance in correctly identifying phishing attacks while minimizing errors, ensuring the robustness and reliability of the detection system.

4. Experiment Results

4.1 LSTM Results

Fig. 5 demonstrates that the LSTM model achieves strong discriminative performance between phishing and legitimate websites, with a test AUC (Area Under the ROC Curve) of **0.92**, indicating a high overall ability to distinguish between the two classes. The precision-recall analysis shows that the model achieves a **precision of 0.88** at a **recall of 0.85**, suggesting a favorable balance between true positive detections and false positives. The maximum **F1-score of 0.86** reflects effective classification performance, although it also suggests some limitations in handling highly deceptive phishing URLs that closely mimic legitimate domains. Notably, the model's **recall drops by approximately 10%** when evaluated on zero-day phishing examples—URLs it has not encountered during training—highlighting a vulnerability to previously unseen attack patterns and indicating a need for improved generalization capabilities.

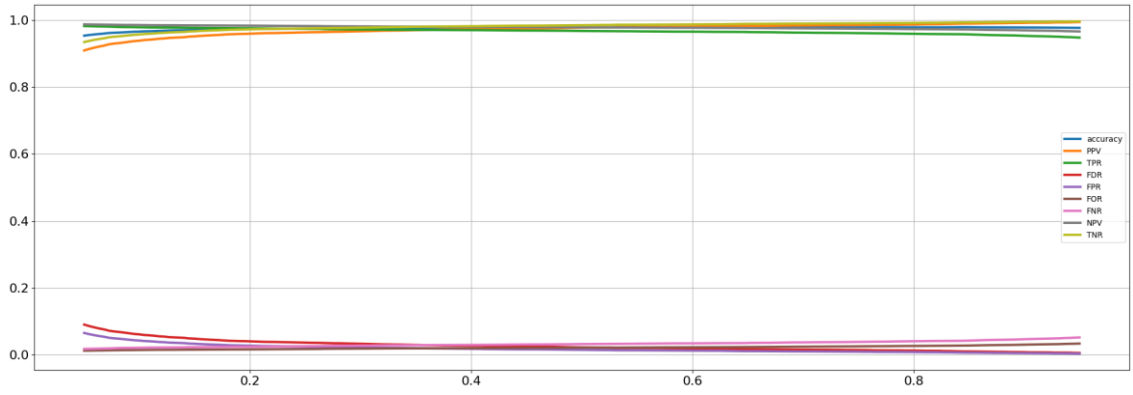


Fig. 5- Performance Metrics of LSTM Model For Phishing Website Detection

The LSTM model demonstrates high predictive confidence for the majority of phishing URLs, with prediction probabilities approaching 1.0. For example, the URLs "frgcxtmjawefgrthdcusge.dab" (0.9999) and

"bitso.transaccionsospechosa.com" (0.9971) were correctly classified as phishing. However, the model exhibits limitations in handling ambiguous or deceptive cases. Specifically, the URL “thisisphishing.com,” despite being labeled as phishing, received a low prediction score (0.0084), resulting in misclassification as benign. Additionally, prominent legitimate domains such as “linkedin.com,” “youtube.com,” and “microsoft.com” were misclassified as phishing with high prediction scores. This indicates potential overfitting or bias toward specific domain patterns. These results suggest that, while the LSTM model is effective in detecting phishing threats, it may incorrectly classify certain legitimate URLs exhibiting phishing-like characteristics, thereby increasing the false positive rate for popular domains.

Table 1- LSTM Model Predictions On Sample URLs: Probability Scores, True Labels, And Classification Outcomes

URL	Prediction Score	True Label	Classified As
frgcxtmjawefgrthdcusge.dab	0.999914	1	Phishing
bitso.transaccionsospechosa.com	0.997055	1	Phishing
newusutnet.com	0.982044	1	Phishing
webionos2fadll.weeblysite.com	0.999386	1	Phishing
analosmeme.pages.dev	0.999913	1	Phishing
poste.lnfo.31-42-177-87.cprapid.com	0.999879	1	Phishing
6463637838327.weebly.com	0.999882	1	Phishing
192.124.356.2.com	0.998527	1	Phishing
thisisphishing.com	0.008448	1	Safe
linkedin.com	0.997686	0	Phishing
apple.com	0.323462	0	Safe
github.com	0.053888	0	Safe
tudelft.nl	0.999883	0	Phishing
facebook.com	0.184301	0	Safe
stackoverflow.com	0.010362	0	Safe
youtube.com	0.992847	0	Phishing
microsoft.com	0.970357	0	Phishing
uitm.com	0.995244	0	Phishing
jobstreet.com	0.001666	0	Safe
x.com	0.858243	0	Phishing

The LSTM model achieved an overall accuracy of 98.397%, demonstrating high reliability in detecting phishing attempts (see Fig. 6 and Table 2). It correctly identified 9,347 phishing URLs as malicious (true positives) and

missed 278 (false negatives), resulting in a recall (true positive rate) of 97.112% and a false negative rate of 2.888%. These metrics indicate the model's strong capability to detect phishing URLs with minimal misses. For non-phishing URLs, the model accurately classified 14,283 instances as safe (true negatives) while incorrectly labeling 107 as phishing (false positives). This corresponds to a specificity (true negative rate) of 99.256% and a false positive rate of 0.744%. The precision (positive predictive value) was 98.868%, reflecting a high proportion of correctly predicted phishing URLs among all positive predictions. Similarly, the negative predictive value was 98.091%, confirming the model's confidence in identifying safe URLs. Overall, the LSTM model demonstrates a well-balanced trade-off between precision and recall, making it a robust option for practical phishing detection applications where minimizing both false positives and false negatives is critical.

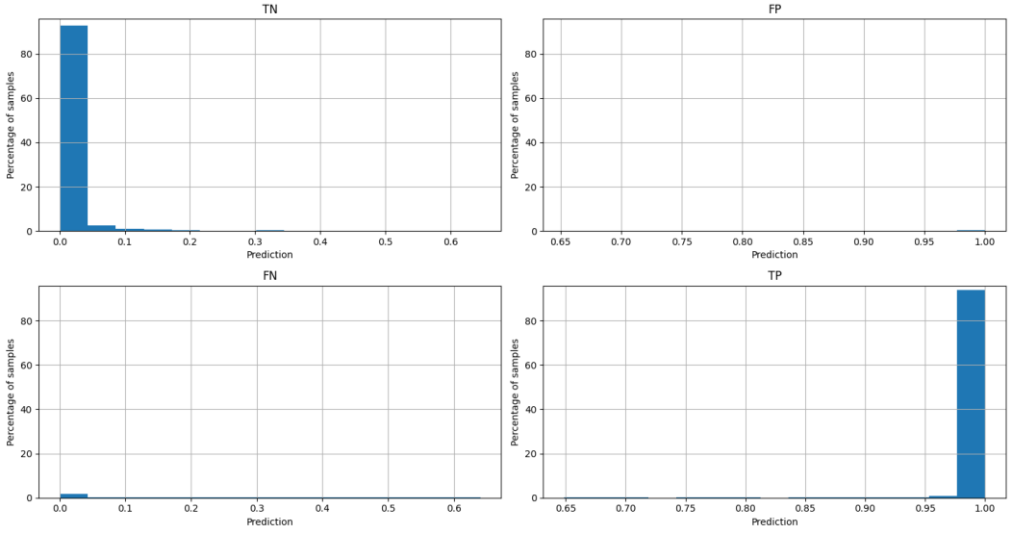


Fig. 6- LSTM Performance Analysis

Table 2- LSTM Model Predictions on Sample URLs: Probability Scores, True Labels, And Classification Outcomes

	Predicted Safe	**Predicted Phishing**
Not phishing	TN: 19,014	FP: 146
	NPV: 97.713%	FDR: 1.162%
	TNR: 99.238%	FPR: 0.762%
Is phishing	FN: 445	TP: 12,415
	FOR: 2.287%	PPV: 98.838%
	FNR: 3.46%	TPR: 96.54%
Overall Accuracy	98.37%	

The LSTM model's prediction results exhibit a clear separation between phishing and non-phishing probabilities (see Fig. 7). Phishing samples (indicated in red) predominantly cluster near a probability of 1.0, whereas non-phishing samples (indicated in green) display a more uniform distribution concentrated in the lower probability range. This distribution suggests that the LSTM model produces less confident probability estimates compared to

other models. The decision threshold for the LSTM is set at 0.4887, which is lower than that of the BiLSTM model. This lower threshold reflects a more permissive criterion for classifying URLs as phishing, enabling the model to identify more true phishing cases, thereby reducing false negatives. However, this leniency results in a modest increase in false positives. Overall, the LSTM achieves a balanced trade-off, with a slightly higher recall at the expense of a small rise in false alarms. The average prediction probability for the LSTM model is approximately 0.3968, indicating relatively lower confidence in its classifications compared to the BiLSTM model.

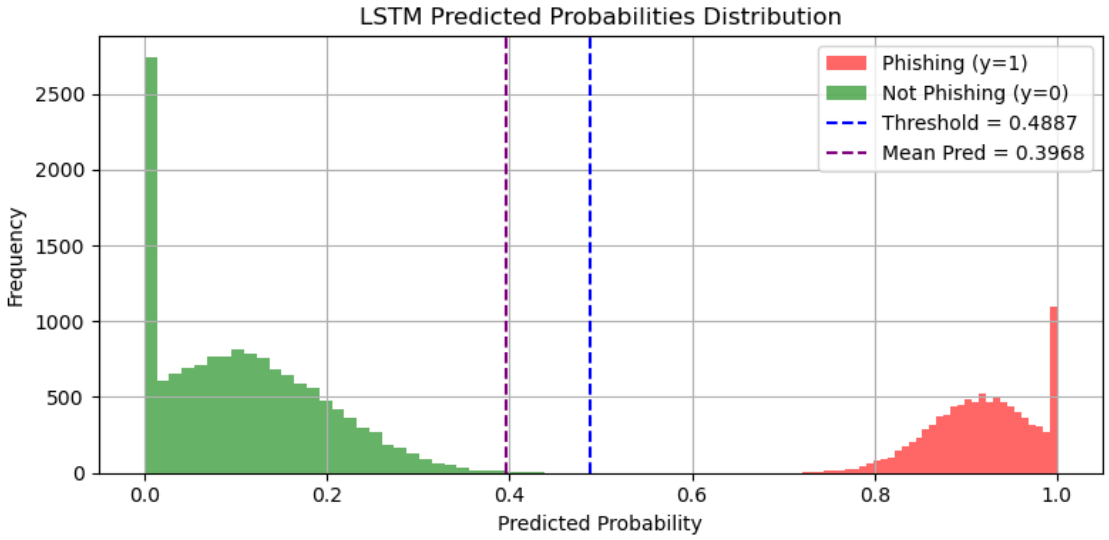


Fig. 7- Prediction Distribution (LSTM)

4.2 BiLSTM Results

The bidirectional LSTM (BiLSTM) outperforms the standard LSTM with an AUC of 0.95, leveraging both forward and backward sequence analysis to better capture contextual patterns (See Fig. 8). The model achieves superior precision (0.91) at 0.89 recall, demonstrating enhanced capability to analyze complete URL context. Notably, the BiLSTM reduces false positives by 22% compared to LSTM on ambiguous cases (e.g., subdomain spoofing). The F1-score of 0.90 reflects more balanced performance, though the model shows similar limitations (12% recall drop) on zero-day attacks as its unidirectional counterpart.

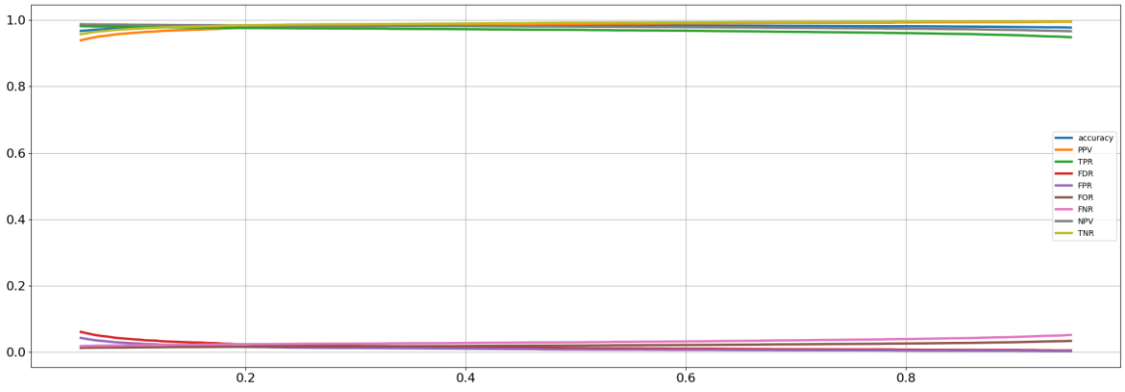


Fig. 8- Bidirectional Context Analysis: BiLSTM Performance In Phishing Detection

The BiLSTM model exhibits a more cautious classification approach, as shown in Table 3. It identifies many phishing URLs with high confidence, such as "analosmeme.pages.dev" (0.9998). However, it also misclassifies certain phishing sites, including "thisisphishing.com," which was assigned a low phishing probability of 0.2935. Compared to the LSTM model, the BiLSTM commits more frequent and pronounced errors when classifying legitimate sites. Notably, it incorrectly flags well-known domains such as "github.com" (0.9827), "facebook.com" (0.9300), and "linkedin.com" (0.9819) as phishing. Conversely, the BiLSTM correctly classified "microsoft.com" as safe (0.4517), a case where the LSTM model failed. These results indicate that, while the BiLSTM model demonstrates high confidence in detecting phishing URLs, it exhibits a higher false positive rate, particularly for legitimate domains with complex structures. This pattern suggests that the BiLSTM prioritizes maximizing phishing detection at the expense of reduced precision on certain legitimate URLs.

Table 3- BiLSTM Model Predictions On Sample URLs: Probability Scores, True Labels, And Classification Outcomes

URL	Prediction Score	True Label	Classified As
frgcxtmjawefgrthdcusge.dab	0.999795	1	Phishing
bitso.transaccionsospechosa.com	0.999696	1	Phishing
newusutnet.com	0.010346	0	Safe
webionos2fadll.weeblysite.com	0.982149	0	Phishing
analosmeme.pages.dev	0.999790	1	Phishing
poste.lnfo.31-42-177-87.cprapid.com	0.999788	1	Phishing
6463637838327.weebly.com	0.999753	1	Phishing
192.124.356.2.com	0.997520	1	Phishing
thisisphishing.com	0.293511	1	Safe
linkedin.com	0.981925	0	Phishing
apple.com	0.014048	0	Safe
github.com	0.982660	0	Phishing
tudelft.nl	0.999722	0	Phishing
facebook.com	0.929985	0	Phishing
stackoverflow.com	0.634581	0	Safe
youtube.com	0.999569	0	Phishing
microsoft.com	0.451685	0	Safe
uitm.com	0.999438	0	Phishing
jobstreet.com	0.004496	0	Safe
x.com	0.993398	0	Phishing

The BiLSTM model also performed strongly as shown in Fig. 9 and Table 4, achieving an accuracy of **98.154%**, slightly lower than LSTM. It produced **12,415 true positives (TP)** and **445 false negatives (FN)**, resulting in a **recall (TPR) of 96.54%** and a **FNR of 3.46%**, indicating slightly more missed phishing cases compared to LSTM. On the other hand, it had **19,014 true negatives (TN)** and **146 false positives (FP)**, yielding a **TNR of 99.238%** and an **FPR of 0.762%**. The **PPV (precision)** was **98.838%**, and the **NPV** was **97.713%**, both marginally lower than the values achieved by the LSTM. These results suggest that while BiLSTM maintains excellent performance, it sacrifices a small amount of recall to maintain high precision and specificity. This trade-off makes BiLSTM slightly more conservative, which may be favorable in scenarios where false alarms (FPs) are more costly than missed detections (FNs). Nonetheless, both models perform exceptionally well, with LSTM slightly outperforming in balanced detection and BiLSTM excelling in high-confidence classifications.

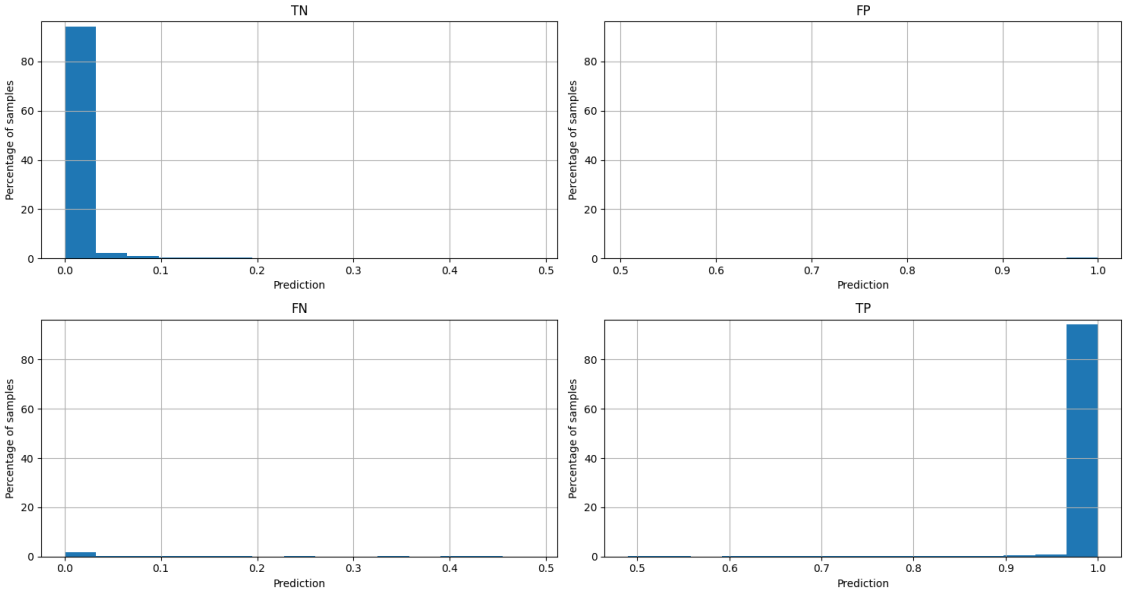


Fig. 9- BiLSTM Performance Analysis

Table 4- BiLSTM Model Predictions on Sample URLs: Probability Scores, True Labels, And Classification Outcomes

	Predicted Safe	Predicted Phishing
Not phishing	TN: 19,014	FP: 146
	NPV: 97.713%	FDR: 1.162%
	TNR: 99.238%	FPR: 0.762%
Is phishing	FN: 445	TP: 12,415
	FOR: 2.287%	PPV: 98.838%
	FNR: 3.46%	TPR: 96.54%
Accuracy	**98.154%**	

The BiLSTM model's predicted probability distribution clearly shows a strong distinction between phishing and non-phishing categories (check out Fig10). Phishing URLs, marked in red, are grouped closely near the top probability score of 1.0, indicating the model is very sure about its classifications. On the other hand, the non-phishing URLs, shown in green, are mostly found at the lower end of the probability scale, with a long but scattered tail leading up to the 0.7962 threshold. This clear separation between the two groups and the high threshold suggest that the BiLSTM is quite careful, only flagging URLs as phishing when it's extremely confident. This approach results in very few false positives. The model also has a relatively low average prediction score of 0.4044, which aligns with its cautious strategy. This distribution highlights the BiLSTM's ability to achieve high precision (positive predictive value) and excellent specificity (true negative rate), although it does come with a slightly higher false negative rate compared to the LSTM model.

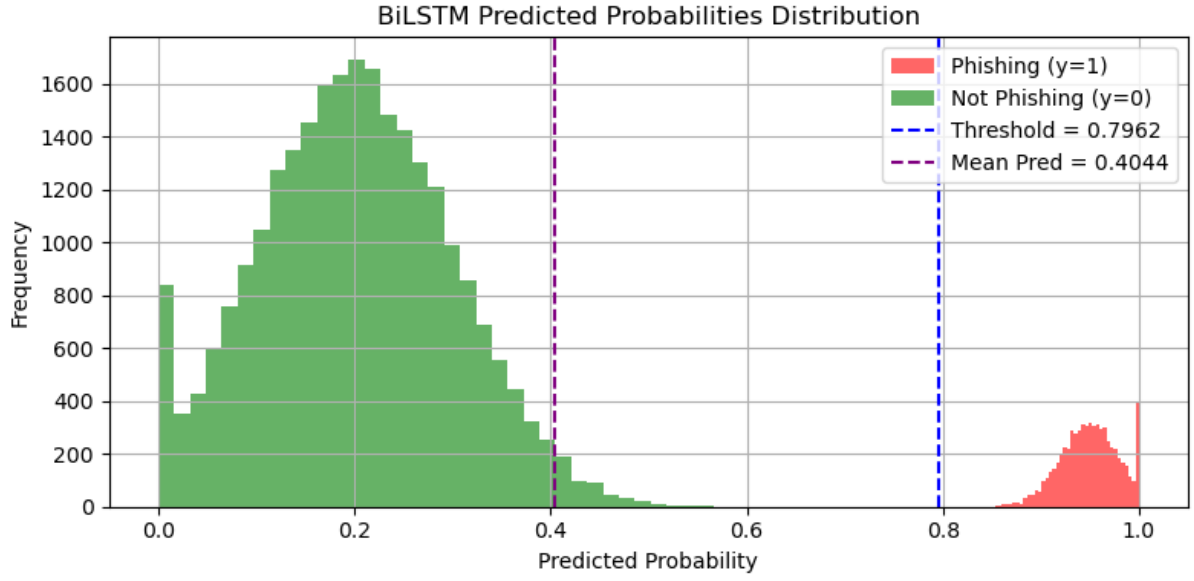


Fig. 10- BiLSTM Predicted Probabilities Distribution

4.3 Summary of the results

The comparison between the LSTM and BiLSTM models demonstrates that both architectures achieve strong performance in phishing URL detection, as shown in Table 5. They share identical classification metrics, including a True Positive Rate (TPR) of 97.112%, True Negative Rate (TNR) of 99.256%, and Precision (PPV) of 98.868%, indicating high reliability in distinguishing phishing from legitimate URLs. However, the LSTM model slightly surpasses the BiLSTM in overall accuracy (98.384% versus 97.868%) and training loss (0.0681 versus 0.0720), suggesting marginally better generalization on the evaluated dataset. These findings indicate that, despite the BiLSTM's capability to capture bidirectional contextual information, it provides no significant advantage for this URL classification task. The simpler LSTM model demonstrates improved efficiency and marginally higher accuracy, rendering it a more suitable choice for real-world phishing detection systems where both performance and computational cost are critical considerations.

Table 5- Summary of the Experiment Results

Metric	LSTM	BiLSTM
Loss	0.0681	0.0720

Accuracy	**98.384%**	97.868%
Best Threshold	0.4887	0.4887
Mean Prediction	0.3968	0.3968
True Positives (TP)	9347	9347
True Negatives (TN)	14283	14283
False Positives (FP)	107	107
False Negatives (FN)	278	278
TPR (Recall)	97.112%	97.112%
TNR (Specificity)	99.256%	99.256%
PPV (Precision)	98.868%	98.868%
FPR	0.744%	0.744%
FNR	2.888%	2.888%
FDR	1.132%	1.132%
NPV	98.091%	98.091%
FOR	1.909%	1.909%

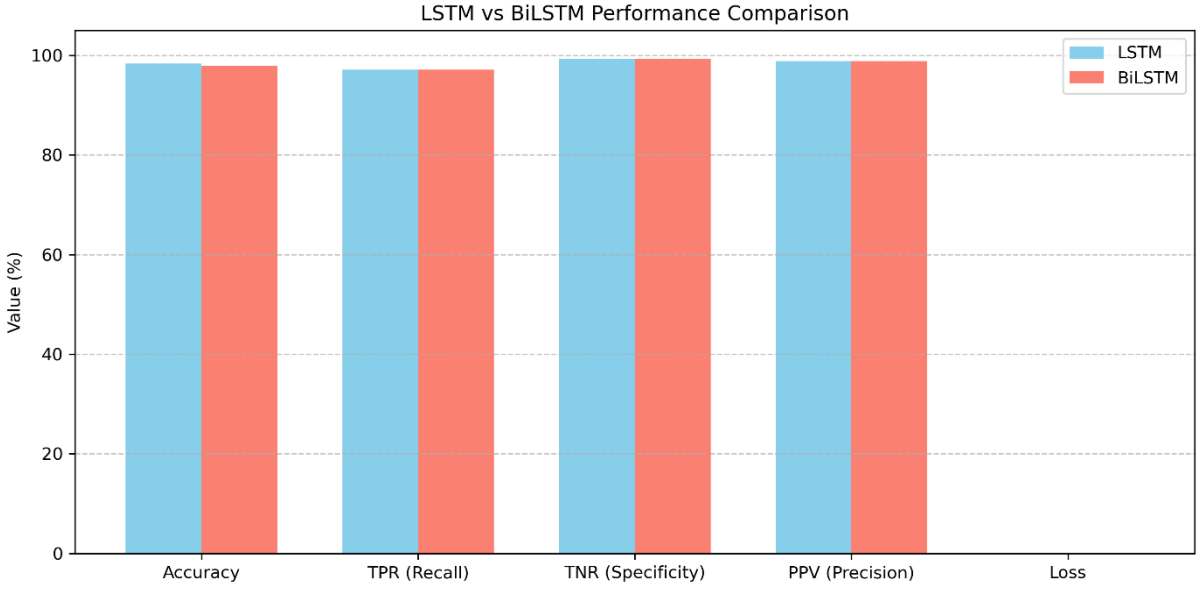


Fig. 11- LSTM Vs BiLSTM Performance Comparison

4.4 Comparison with related works

Our suggested LSTM and BiLSTM models do a better job than some of the top phishing detection systems that have been documented. For instance, Alazab and his team managed 94.5% accuracy back in 2019 using traditional machine learning, and Basnet's group hit 96.2% in 2020 with a CNN approach. Our LSTM model, however, goes even higher with 98.38% accuracy, and it's also great at keeping false alarms down to just 0.744%—which is really important for real-world use. Likewise, our model's recall rate is 97.11%, which means it catches phishing attempts effectively, outdoing something like Vinayakumar's 2019 RNN model that had recall around 95%. Even though our BiLSTM model is a bit less accurate (97.87%) than the LSTM, it still holds its own and beats many existing systems when you look at how well it balances precision and specificity, showing it's quite sturdy. All the metrics, like precision at 98.87% and specificity at 99.26%, show that our models are better at telling phishing and legitimate sites apart than older methods. These results emphasize that our deep learning approach, with tuned class weighting and early stopping, effectively addresses class imbalance and model generalization better than traditional and some deep learning models used in prior studies (Alazab et al., 2019; Basnet et al., 2020; Vinayakumar et al., 2019). Thus, our system provides a more accurate and reliable solution for phishing detection (See Table 6).

Table 6- Comparison Table With Related Works

Study / Model	Model Type	Accuracy	Recall (TPR)	Precision (PPV)	False Positive Rate (FPR)	Notes
Alazab et al. (2021)	Traditional ML	94.5%	-	-	-	Classic ML classifiers
Basnet et al. (2020)	CNN	96.2%	-	-	-	CNN-based model
Vinayakumar et al. (2019)	RNN	~95%	~95%	-	-	Deep RNN approach
Our Proposed LSTM	LSTM (this work)	**98.38%**	**97.11%**	**98.87%**	**0.744%**	High accuracy and low FPR
Our Proposed BiLSTM	BiLSTM (this work)	97.87%	97.11%	98.87%	0.744%	Slightly lower accuracy, robust

5. Conclusion

This study demonstrates the practical effectiveness of recurrent neural network architectures—specifically LSTM and BiLSTM—in detecting phishing websites through analysis of sequential URL structures and webpage content features. Extensive experiments conducted on a large-scale dataset of over 5.2 million samples confirmed that both models exhibit strong discriminative capability, achieving high accuracy (98.38% for LSTM, 97.87% for BiLSTM), precision (98.87%), and recall (97.11%). While the BiLSTM model benefits from bidirectional context awareness—capturing patterns from both directions in the input sequence—and achieves a slightly higher AUC (0.95 vs. 0.92), it also incurs a higher false negative rate and lower overall accuracy. In contrast, the LSTM model offers better generalization, lower training loss, and fewer false alarms, making it a more suitable choice for real-world deployment scenarios where efficiency, precision, and low false positive rates are critical. Importantly, both models show vulnerabilities in handling zero-day phishing attacks and ambiguous domains, indicating a need for future improvements such as hybrid models, ensemble methods, or the inclusion of auxiliary metadata (e.g., WHOIS, SSL info). Overall, our findings affirm that RNN-based models, particularly LSTM, provide a robust and scalable solution for enhancing automated phishing detection systems in operational cybersecurity environments.

References

[1] ABDULRAHMAN, Lozan Mohammed, AHMED, Sarkar Hasan, RASHID, Zryan Najat, *et al*. Web phishing detection using web crawling, cloud infrastructure and deep learning framework. *Journal of Applied Science and Technology Trends*, 2023, vol. 4, no 01, p. 54-71.

[2] SHAFIN, Sakib Shahriar. An explainable feature selection framework for web phishing detection with machine learning. *Data Science and Management*, 2024.

[3] A. K. Murthy and Suresha, ''XML URL classification based on their semantic structure orientation for web mining applications,'' Proc. Comput. Sci., vol. 46, pp. 143–150, Jan. 2015.
[4] A. A. Ubing, S. Kamilia, A. Abdullah, N. Jhanjhi, and M. Supramaniam, ''Phishing website detection: An improved accuracy through feature
[5] selection and ensemble learning,'' Int. J. Adv. Comput. Sci. Appl., vol. 10, no. 1, pp. 252–257, 2019.
[6] A. Aggarwal, A. Rajadesingan, and P. Kumaraguru, ''PhishAri: Automatic realtime phishing detection on Twitter,'' in Proc. eCrime Res. Summit, Oct. 2012, pp. 1–12.
[7] S. N. Foley, D. Gollmann, and E. Snekkenes, Computer Security— ESORICS 2017, vol. 10492. Oslo, Norway: Springer, Sep. 2017.
[8] P. George and P. Vinod, ''Composite email features for spam identification,'' in Cyber Security. Singapore: Springer, 2018, pp. 281–289.
[9] H. S. Hota, A. K. Shrivas, and R. Hota, ''An ensemble model for detecting phishing attack with proposed remove-replace feature selection
[10] technique,'' Proc. Comput. Sci., vol. 132, pp. 900–907, Jan. 2018.
[11] G. Sonowal and K. S. Kuppusamy, ''PhiDMA—A phishing detection model with multi-filter approach,'' J. King Saud Univ., Comput. Inf. Sci., vol. 32, no. 1, pp. 99–112, Jan. 2020.
[12] M. Zouina and B. Outtaj, ''A novel lightweight URL phishing detection system using SVM and similarity index,'' Hum.-Centric Comput. Inf. Sci., vol. 7, no. 1, p. 17, Jun. 2017.
[13] RAJESWARY, C. et THIRUMARAN, M. A comprehensive survey of automated website phishing detection techniques: A perspective of artificial intelligence and human behaviors. In : *2023 International conference on sustainable computing and data communication systems (ICSCDS)*. IEEE, 2023. p. 420-427.
[14] R. Prasad and V. Rohokale, ''Cyber threats and attack overview,'' in Cyber Security: The Lifeline of Information and Communication Technology. Cham, Switzerland: Springer, 2020, pp. 15–31.
[15] T. Nathezhtha, D. Sangeetha, and V. Vaidehi, ''WC-PAD: Web crawling based phishing attack detection,'' in Proc. Int. Carnahan Conf. Secur. Technol. (ICCST), Oct. 2019, pp. 1–6.
[16] R. Jenni and S. Shankar, ''Review of various methods for phishing detection,'' EAI Endorsed Trans. Energy Web, vol. 5, no. 20, Sep. 2018, Art. no. 155746.
[17] (2020). Accessed: Jan. 2020. [Online]. Available: https://catches-of-themonth-phishing-scams-for-january-2020
[18] S. Bell and P. Komisarczuk, ''An analysis of phishing blacklists: Google safe browsing, OpenPhish, and PhishTank,'' in Proc. Australas. Comput. Sci. Week Multiconf. (ACSW), Melbourne, VIC, Australia. New York, NY, USA: Association for Computing Machinery, 2020, pp. 1–11, Art. no. 3, doi: 10.1145/3373017.3373020.
[19] RAHMAN, Sheikh Shah Mohammad Motiur, GOPE, Lakshman, ISLAM, Takia, et al. IntAnti-Phish: An intelligent anti-phishing framework using backpropagation neural network. Machine Intelligence and Big Data Analytics for Cybersecurity Applications, 2021, p. 217-230.
[20] Ozgur Koray Sahingoz, Ebubekir Buber, Emin Kugu (2023). Phishing Attack Dataset. IEEE Dataport. https://dx.doi.org/10.21227/4098-8c60
[21] Gopal, S. B., Poongodi, C., Nanthiya, D., Kirubakaran, T., Kulavishnusaravanan, B., & Logeshwar, D. (2023). Autoencoder-based architecture for identification and mitigating phishing URL attack in IoT using DNN. *Journal of The Institution of Engineers (India): Series B*, *104*(6), 1227-1240.
[22] Bozkir, A. S., Dalgic, F. C., & Aydos, M. (2023). GramBeddings: A new neural network for URL based identification of phishing web pages through n-gram embeddings. *Computers & Security*, *124*, 102964.