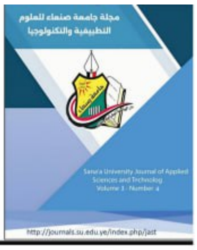
SUN PUBLISHER

مجلة جامعة صنعاء للعلوم التطبيقية والتكنولوجيا

Sana'a University Journal of Applied Sciences and Technology

https://journals.su.edu.ye/index.php/jast/



A Hybrid CNN-BLSTM Model for Phishing Attack Detection Using Deep Learning to Strengthen Internet Security

Abdulrahman A. Alsabri[1] * **and Marwa A. Al-Hadi** [2]

[1]**Department of Information System, Faculty of Computer and Information Technology, Sana'a University, Sana'a, Yemen,**
[2]**Department of Computer Science, Faculty of Computer and Information Technology, Sana'a University, Sana'a, Yemen**

*Corresponding author: abd.alsabri@su.edu.ye

Abstract

Phishing attacks continue to be a persistent and critical challenge in the cybersecurity landscape, exploiting weaknesses in Internet security through sophisticated tactics, such as social engineering, deceptive domain names, and URL obfuscation. Traditional detection systems often fail to identify these evolving threats because of their limited adaptability and high false positive rates. In this study, a deep learning approach is proposed to enhance the accuracy of phishing detection and minimize erroneous classifications. Leveraging a comprehensive dataset composed of key URL-related features, including URL structure, domain age, presence of HTTPS, and lexical patterns, we implemented and compared the performance of four deep learning models: Convolutional Neural Network with Bidirectional Long Short-Term Memory (CNN-BLSTM), Self-Normalizing Neural Network (SNN), Transformer, and Deep Belief Network (DBN). Among these, the CNN-BLSTM model achieved the highest accuracy of 81%, demonstrating its superior ability to capture sequential and spatial patterns inherent in phishing URLs. The experimental results confirm that deep-learning methods outperform conventional techniques in detecting phishing attempts. However, challenges such as high computational complexity limit real-time deployment. This study highlights the transformative potential of deep learning for strengthening online threat detection and enhancing cybersecurity defenses.

Article Info

Keywords:
Phishing Detection, Deep Learning Models, Cybersecurity, URL Obfuscation, CNN-BLSTM Network

Article History:
Received: 19-May-2025,
Revised: 27-June-2025,
Accepted: 21-July-2025,
Available online: 28 August 2025.

1. INTRODUCTION

Attackers, as trustworthy organizations, trick users to share sensitive information, such as passwords and bank account details. These attacks result in significant financial losses, identity theft, and damage to a person's reputation. As Internet access and digital services have rapidly increased, phishing attacks have become a significant cybersecurity priority because of their widespread influence. Phishing remains a major cybercrime component, causing millions of users to incur financial losses [1],[2]. The higher the threat count frequency is, the more critical it is to have stronger detection systems. These systems must be capable of pinpointing millions of threats with finely tuned accuracy and speed, especially when infiltrators adopt smarter evasion tactics.

Traditional phishing detection systems use rule-based filters and heuristic-based methods that identify common indicators such as IP addresses, email headers, and keywords. Although these systems have some success, their reliance on predetermined criteria and signatures limits their ability to adapt to the new phishing techniques that conventional systems often overlook [3]. Phishing attacks can utilize obfuscation techniques such as URL shortening or encoding, which may deceive traditional filters [4]. Current detection methods are struggling to cope with today's complex phishing attacks. Furthermore, reliance on signature-based approaches leads to an increase in false positives, which degrades users' trust in security alerts over time, and reduces the effectiveness of these systems [3].

However, deep learning has emerged as the most promising approach for overcoming the limitations of current work, improving accuracy, and providing far superior modeling capabilities. Deep learning models can uncover complex patterns and extract insights from large datasets, revealing hidden connections in phishing data that traditional methods often miss because they rely solely on rule-based systems [5]. Advanced deep learning architectures, such as convolutional neural networks (CNNs), recurrent neural networks (RNNs), and, more recently, transformers, have been utilized to analyze URLs, email content, and website information. These models are more effective in identifying phishing attempts [6]. Numerous studies have demonstrated that deep learning models enhance the detection rates for known and unknown phishing behaviors, which are crucial in a dynamic threat environment [7].

One of the most challenging steps in phishing detection is URL parsing, as attackers modify URLs to make them resemble legitimate domains. Attackers often use a combination of URL shortening, domain obfuscation, and random string generation to deceive consumers while evading detection systems. In the real world, these tactics should be handled using phishing detection deep learning models to ensure their accuracy [3], [8]. Real-time performance experiments that use deep learning methods are complex in terms of resource computation, and current real-time detection models are not optimal. Achieving a balance between the detection accuracy and computational requirements is essential when developing deep-learning-based phishing detection systems for large-scale use.

This study aims to address these problems by building a phishing detection model using deep learning to enhance accuracy. This study explored multiple deep learning architectures, considering the cost (inference time), accuracy, and recall for phishing detection. Previous studies dealing with testing using the deep learning-based system and its detection rates against more traditional methods provide context to clarify how improved performance can result from using the cognitive mechanism. This study aims to improve the process of phishing detection by offering a scalable solution that can be adapted to the evolving cyber threat landscape. The proposed model is considered scalable for several reasons. First, the architecture was designed using modular deep-learning components (e.g., CNN for spatial features and BLSTM for sequence learning), which can be efficiently parallelized on modern GPU hardware. Second, the model was trained and evaluated on a relatively large, balanced dataset of 50,000 URLs, demonstrating its capability to process high volumes of data without performance degradation. Finally, the feature extraction process relies solely on lightweight URL-based attributes, making it feasible to integrate the system into real-world, high-throughput environments, such as email gateways or browser extensions, without significant computational overhead.

2. LITERATURE REVIEW

2.1. Background

Based on three convolutional neural networks (CNN), bidirectional long short-term memory networks (BLSTM), and transformers in deep learning techniques that have improved detection accuracy and adaptability as phishing attacks are increasing their complexity. CNNs offer a high level of performance in phishing detection by extracting the page content structure. Even for a Zero-day attack-related case, Convolutional Neural Networks (CNNs) can recognize phishing websites by distinguishing malicious and safe URLs because they can learn local patterns in the URL structures [1], [9]. Meanwhile, LSTM networks that can handle temporal data have proven their usefulness in email, text, and URL strings over time to learn sequential patterns from phishing attacks [10].

One of the main challenges in phishing detection is to select which attributes represent those traits that best capture the properties of actual examples. Phishing URLs often contain altered formats and deceptive properties, such as unusual keywords and subdomains. Features may be derived from raw data using deep learning models such as CNN and BLSTM. Human feature engineering can be reduced and detected, and the performance will increase. Transformers take one step further and provide the means for models to evaluate individual pieces of URLs in online content, which makes them particularly resistant to sophisticated phishing attacks such as domain obfuscation and homograph attacks [11], [12].

This study concentrates on deep-learning models such as transformer models, Spiking Neural Networks (SNN), and Deep Belief Networks (DBN). They are trained to recognize common patterns found in phishing URLs.The former facilitates time-based learning to promote its just-in-time detection capacity in SNN, and the latter targets discovering complex phishing patterns through hierarchical pattern discovery by correlating URL features across levels of hierarchy DBN [13]. These models were trained with large, diverse datasets consisting of legal and phishing URLs to boost robustness in detection across different types of phishing strategy learning features that can considerably separate classes [14].

This meant using many models to take advantage of their pros, and, as phishing tactics evolved, try permutations against new trends. By integrating the CNN, LSTM, and Transformer architectures, the system benefits from the spatial feature identification experience of CNNs with sequential data processing capabilities in LSTMs (Long Short-term Memory) and attention-based feature extraction techniques related to transformers. The former approach minimizes the false positive progress of HMM and reduces the risks from sophisticated phishing, whereas a combination improves the detection accuracy. Testing models across several phishing scenarios validated the system's high detection rates, with few false positives for attack types [15], [16].

Addressing the limitations of standard models by combining deep learning-based solutions for iris scanners provides a comprehensive solution to the problem of detecting phishing. This could help cybersecurity analysts tackle new variants of attacks that surface, frequently created because of such a high real-time nature.

2.2. Related work

In [1], a model based on deep learning was introduced with a combination of five different architectures to predict phishing attacks: ANN, CNN, RNN, BRNN, and attention networks. However, because the system evaluates the URLs by embedding characters, it is independent. Created a large, well-balanced dump of over 5.1 million URLs consisting of 2.32 M phishing URLs (about ten percent more) from Common Crawl with valid content. Within 32 months, the model defeated the symmetric and asymmetric encryption schemes for classifying phishing attempts with a wide margin of accuracy and performed well against zero-day attacks. However, it was not faultless for more sophisticated attacks such as URL hijacking.

Machine learning algorithms (Naive Bayes and Multinomial Naive Bayes) are used to analyze URL links [15]. The URL is broken into the following features: IP address characteristics, domain characteristics, and elements in the link. The system categorizes links into three classes: phishing, legitimate, and suspicious links. The models were trained using a dataset from UC Irvine that consisted of 31 features per link. Bernoulli Naive Bayes is mostly used for false positive rate reduction (phishing links) detection. Its main disadvantage is that it requires frequent updating to conform to the most recent phishing trends and zero-day attacks. In [17], machine learning was utilized to identify phishing attacks through URL analysis. A model was built using a dataset of over 11 K websites. It extracted (from Google) and predicted phishing on every website by testing a set of features in 30 dimensions. Neural networks achieved an accuracy of 90.23%, Naïve Bayes 92.97%, and the AdaBoost algorithm achieved 95.43% accuracy, utilizing these three machine learning algorithms.

In [11], a robust hybrid phishing detection model using machine and deep learning was presented. They utilized two Kaggle datasets: one contained 11,430 unique URLs and 87 features, and the other was a collection of different labels for a subset of those same URLS, totaling up to 651,191 entries with only two features. This can be interpreted as a combination of random forests and a CNN. The hybrid model yields an accuracy of 97%. It does not test against more advanced threats, such as GANs, as shown in Table 1.

3. MATERIALS AND METHODS

To ensure that phishing detection correctly detects attacks and achieves high accuracy with low false positive rates. The ability to distinguish real attacks from benign attacks offers several advantages. This includes data preparation and model training methods to evaluate the developed models, as shown in Figure 1.

3.1. Research Design

A large URL dataset of real and phishing judgments was utilized to develop a robust Phishing Detection Algorithm. The dataset was populated from the parents of the main sources of legitimate URLs based on Common Crawl to complement PhishTank and OpenPhish [1]. A clean-up routine was performed on the dataset to improve consistency across training samples, which involved normalizing the URL text data, removing duplicate entries, and enforcing a standardized format for URLs. Features were selected using recursive feature elimination, which is an iterative process that excludes less critical aspects while retaining the most essential aspects for phishing detection. This feature selection technique enhances the model's ability to focus on important indicators, such as URL length, domain authority, and location in the URL path, which are known to be effective in distinguishing phishing URLs from legitimate ones [14], [18]. The dataset was compiled from PhishTank, OpenPhish, and Common Crawl. After cleansing and normalization, we acquired a balanced dataset comprising 50,000 URLs evenly distributed between phishing and legitimate categories (25,000 phishing URLs and 25,000 legitimate URLs). The record preprocessing steps included the disposal of reproduction entries, changing all URLs to lowercase, and implementing a standardized shape throughout the dataset. We extracted a preliminary set of 30 features specializing in lexical, structural, and area-based attributes. Examples of extracted features are as follows:

- Lexical features: URL length, presence of special characters (e.g., @, -, %, =), number of dots, number of subdomains, and use of suspicious keywords (e.g., "login," "secure," "account").
- Structural and protocol features: Use of IP address, HTTPS presence, and position of keywords in the path.
- Domain-based features: domain age and authority domain(retrieved using WHOIS and Alexa Rank data).

This procedure selected the top 15 most crucial features, optimizing the study efficiency of the version while preserving key phishing indicators. These subtle capabilities were then used as inputs to teach and evaluate the deep learning models.

Table 1: Comparison of Existing Phishing Detection Approaches

Study	Model / Technique	Dataset	Accuracy	Strengths	Limitations
[1] Sahingoz et al. (2024)	CNN, RNN, BRNN, Attention	5.1M URLs from Common Crawl	∼97%	Handles zero-day attacks; ensemble of DL models	Computationally expensive; limited handling of URL hijacking
[15] Shoaib & Umar (2023)	Naive Bayes, Multinomial NB	UCI URL dataset (31 features)	∼92% (Adaboost)	Clear categorization into phishing, legitimate, suspicious	Requires frequent updates; slow training
[17] Mosa et al. (2023)	NN, NB, Adaboost	11K websites from Google	95.43% (Adaboost)	High accuracy with ensemble methods	Ineffective for short links
[11] Sawant et al. (2024)	Hybrid (ML + DL) – RF + CNN	Kaggle datasets (11,430 + 651,191 entries)	97%	Combines strengths of ML and DL	Not tested against adversarial or GAN-based threats
[13] Kocyigit et al. (2024)	Genetic Algorithm + ML	Phishing URL dataset	Not stated	Enhanced feature selection	Generalizability not discussed

3.2. Procedures

The preprocessed data were split into training, validation, and testing to aid in model training and evaluation. The data was divided as follows: 70% for training, 25% for testing, and 5% for validation. Training was used to teach the algorithm to identify phishing from URL structures and an independent validation set (not involved in bot detection). It aims to adjust the parameters without overfitting [10], [16]. Testing this on different datasets enabled the correct evaluation of whether the model generalizes for new phishing cases.

3.3. Algorithms and Tools

The phishing detection model was developed using the Python software. To construct and optimize deep learning models, a set of experiments with various model architectures (CNN, BLSTM, and Transformer networks) were conducted by utilizing libraries such as TensorFlow and Keras [11]. Libraries are essential tools for managing rich data operations, model layers, and activation functions. Model training was performed using Google ColLab. Google Collab functions as a cloud-based platform, offering free access to GPU resources. This substantially reduces the training time and enhances the computational speed (throwable error). The model parameters are the tweaking components that can be adjusted in real time during training because of Google Collab's integration with Python libraries, ensuring a blazing fast model modification [19].

3.4. Model Tuning and Evaluation

To enhance the model performance, hyperparameter tuning played a significant role in obtaining the best batch size, learning rate, and number of epochs to run on our training dataset. Determining the best combinations of these parameters. The employed methods such as grid and random search resulted in a model that performed well without overfitting [15]. The model was evaluated based on its performance in identifying phishing URLs with fewer false positives and controlled by measures such as accuracy, precision, recall, and F1-score. However, accuracy is a high-level metric [9], [13]. This study helps the phishing detection model enhance accuracy and adversarial robustness to prevent cyber-attacks through spears. Phishing.

4. RESULTS AND DISCUSSION

The evaluation of each deep learning model independently details its assessment metrics and respective performance results, along with their strengths and weaknesses. The accuracy, precision, recall score, and F1 score were used for each model.

4.1. Hybrid CNN-BLSTM Model

The CNN-BLSTM network was proposed, which takes advantage of both the LTSM in learning sequential data and CNNs on feature maps. The model architecture included a temporal layer (LSTM). Feature extraction using convolutional D1. The performance metrics of the CNN-BLSTM model are listed in Table 2.

Table 2: CNN-BLSTM Model Evaluation Metrics

Metric	Class 0 (Legitimate)	Class 1 (Phishing)	Average
Precision	0.80	0.82	0.81
Recall	0.87	0.73	0.80
F1-Score	0.83	0.77	0.80
Accuracy			0.81

The overall metrics for the CNN-BLSTM model, dis-

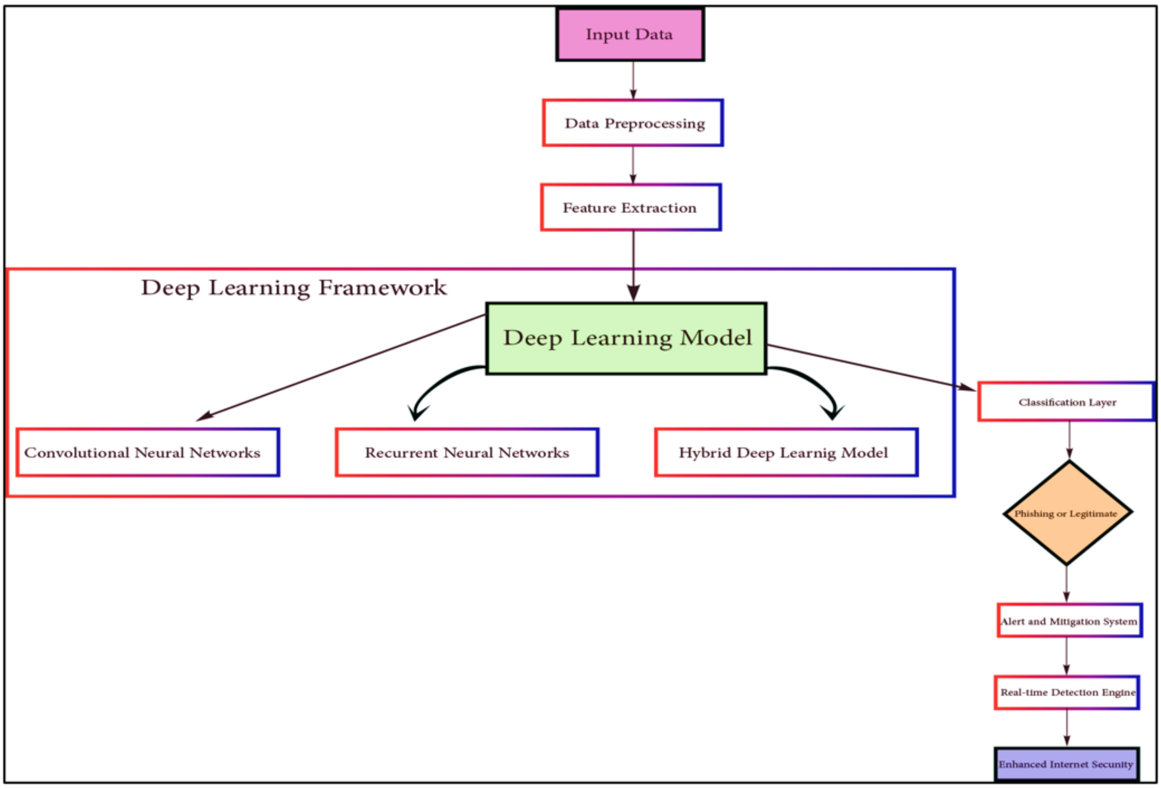

Input Data
Data Preprocessing
Feature Extraction
Deep Learning Framework
Deep Learning Model
Convolutional Neural Networks
Recurrent Neural Networks
Hybrid Deep Learnig Model
Classification Layer
Phishing or Legitimate
Alert and Mitigation System
Real-time Detection Engine
Enhanced Internet Security

Figure 1. Deep Learning Framework for Phishing Detection

played in Table 3 and Figure 2, confirm balanced performance across all metrics.

The CNN-BLSTM model performed well in terms of precision and recall, highlighting its effectiveness in accurately identifying phishing URLs while minimizing false positives. However, the model's high computational cost owing to its complex architecture may limit its practicality in resource-constrained environments.

Table 3: Overall Metrics for CNN-BLSTM Model

Metric	Value
Accuracy	0.81
Macro Avg Precision	0.81
Macro Avg Recall	0.80
Macro Avg F1-Score	0.80
Weighted Avg Precision	0.81
Weighted Avg Recall	0.81
Weighted Avg F1-Score	0.80

4.2 Self-Normalizing Neural Network (SNN) Model

This model uses scaled exponential linear units as activation functions to obtain the self-normalizing properties. This instability makes this easier for large networks. It is built for tasks in which the activation functions remain constant, as shown in Table 4.

The overall performance metrics for the SNN model are presented in Table 5 and Figure 3.

Table 4: CNN-BLSTM Model Evaluation Metrics

Metric	Class 0 (Legitimate)	Class 1 (Phishing)	Average
Precision	0.80	0.79	0.80
Recall	0.84	0.74	0.79
F1-Score	0.82	0.76	0.79
Accuracy			0.80

Table 5: Overall Metrics for the SNN Model

Metric	Value
Accuracy	0.80
Macro Avg Precision	0.80
Macro Avg Recall	0.79
Macro Avg F1-Score	0.79
Weighted Avg Precision	0.80
Weighted Avg Recall	0.80
Weighted Avg F1-Score	0.80

The SNN model demonstrated balanced accuracy, precision, and recall, demonstrating generalization for phishing detection. However, despite its stability, it slightly underperforms compared to the CNN-BLSTM model in recall, which indicates a moderate limitation in identifying all phishing instances.

4.3 Transformer Model

The Transformer model leverages self-attention mechanisms to capture complex relationships in the data, making it particularly adept at handling sequential dependencies on the URLs. This model is effective in identifying phishing attempts with sophisticated patterns. The evaluation metrics for the transformer model are listed in Tables 6, 7, and Figure 4, which provide the overall performance metrics for the transformer model.

Table 6: Transformer Model Evaluation Metrics

Metric	Class 0 (Legitimate)	Class 1 (Phishing)	Average
Precision	0.80	0.80	0.80
Recall	0.85	0.74	0.79
F1-Score	0.83	0.77	0.80
Accuracy			0.80

While the Transformer model performs well, its high computational demand owing to the multihead attention mechanism poses a limitation. Although effective in capturing complex relationships, it requires substantial computational resources, which can be challenging for real-time applications.

Table 7: Overall Metrics for Transformer Model

Metric	Value
Accuracy	0.80
Macro Avg Precision	0.80
Macro Avg Recall	0.79
Macro Avg F1-Score	0.80
Weighted Avg Precision	0.80
Weighted Avg Recall	0.80
Weighted Avg F1-Score	0.80

4.4 Deep Belief Network (DBN) Model

The DBN model combines restricted Boltzmann machines (RBM) with a multilayer perceptron (MLP) for hierarchical feature extraction and classification. This model is particularly suited for capturing high-level patterns in phishing URLs. The performance metrics of the DBN model are presented in Table 7.

Table 8: DBN Model Evaluation Metrics

Metric	Class 0 (Legitimate)	Class 1 (Phishing)	Average
Precision	0.70	0.81	0.75
Recall	0.90	0.51	0.71
F1-Score	0.79	0.63	0.71
Accuracy			0.73

The overall performance metrics for the DBN model are shown in Table 9

Table 9: Overall Metrics for the DBN Model

Metric	Value
Accuracy	0.73
Macro Avg Precision	0.75
Macro Avg Recall	0.71
Macro Avg F1-Score	0.71
Weighted Avg Precision	0.75
Weighted Avg Recall	0.73
Weighted Avg F1-Score	0.71

The DBN model, which is capable of capturing high-level patterns, underperforms relative to the CNN-LSTM and Transformer models, particularly in terms of recall for phishing instances. This lower recall indicates a limitation in the ability of the model to identify all phishing URLs, suggesting the need for further optimization or hybridization with other models.

4.5 Comparative Summary

The overall comparison, shown in Table 10 and figure 5, highlights the performance of each model across key metrics.

Table 10: Comparative Summary of Model Performance

Model	Accuracy	Precision	Recall	F1-Score
CNN-BLSTM	0.81	0.81	0.80	0.80
SUN	0.80	0.80	0.79	0.80
Transformer	0.80	0.80	0.79	0.80
DBN	0.73	0.75	0.71	0.71

4.6 Discussion

These results provide insights into the performance analysis of each model for use in phishing detection systems.

Every deep-learning model has advantages and disadvantages. The CNN BLSTM model demonstrated a comparability balance between precision and recall with 81 % accuracy, reduced to false positives at the highest. Because CNN is good at detecting the spatial features of URLs, it can help capture complex dependencies among sequences, which is critical for phishing detection [1]. However, implementing the CNN-BLSTM model is computationally expensive and highly unscalable in resource-constrained scenarios.

The model Self-Normalizing Neural Network (SNN) also achieved an accuracy of 80%. Similarly, the self-normalizing properties of SELU improve overfitting by promoting fast convergence and stability during training to generalize well to novel phishing patterns [20]. While the SNN has a recall that is slightly lower than that of the CNN-BLSTM model, it may not support more complex phishing URLs to the degree being presented, and its simpler structure makes an acceptable tradeoff between precision and computational resource consumption effectiveness, meaning it can be deployed in resource-constrained environments.

The Transformer model, known for its self-attention mechanism, showed a strong capability to learn about multiple phishing patterns with reasonable accuracy (80%) and recall (79%). It is used to focus on different segments of the URL [21]. This makes it particularly suitable for detecting complex patterns. The complicating factor here is the multi-head attention layers, which are time-consuming to compute and can degrade the performance at inference time. While this strategy may need to be optimized for effective real-time applications, it remains effective in high-stakes work, where sufficient computing resources justify the cost.

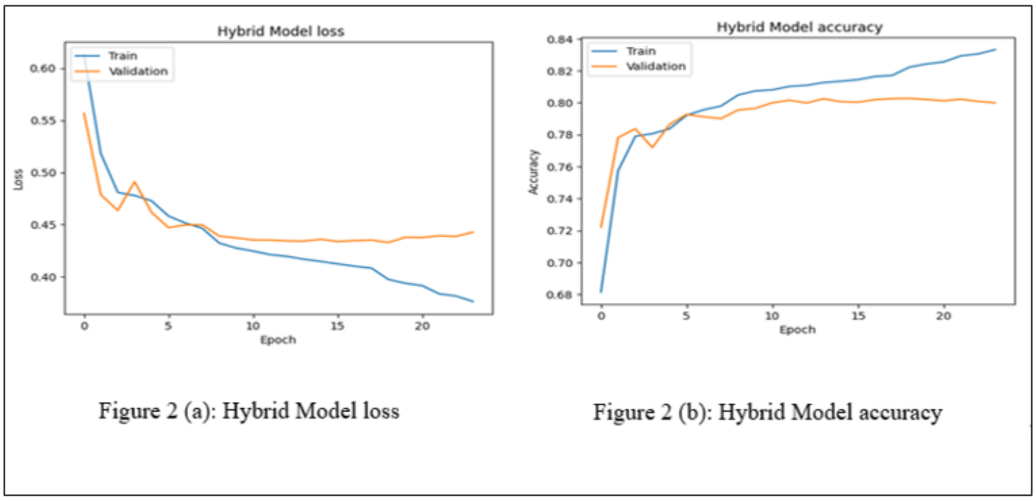


Figure 2 (a): Hybrid Model loss

Figure 2 (b): Hybrid Model accuracy

Figure 2. Performance of CNN-LSTM Model

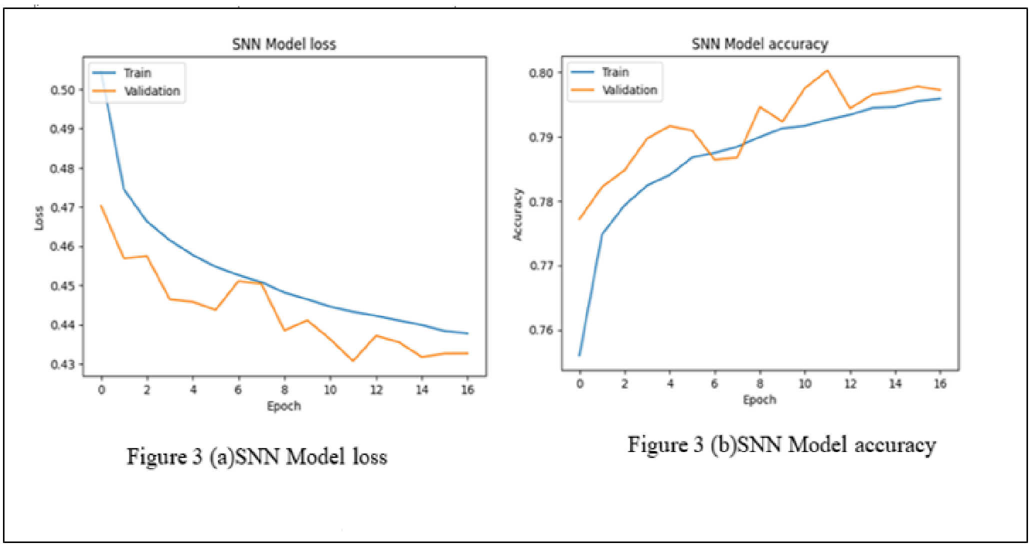


Figure 3 (a)SNN Model loss

Figure 3 (b)SNN Model accuracy

Figure 3. Performance of SNN Model

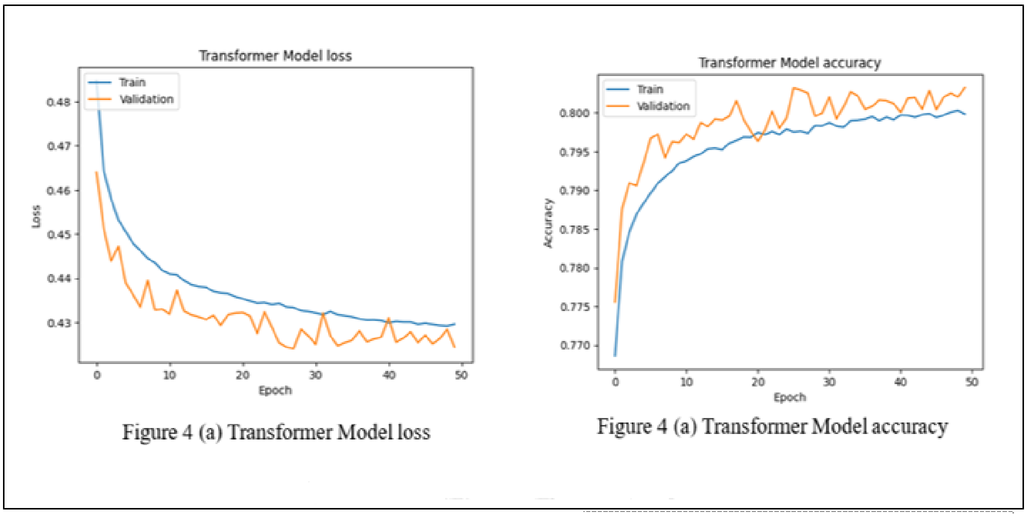


Figure 4 (a) Transformer Model loss

Figure 4 (a) Transformer Model accuracy

Figure 4. Performance of Transformer Model

The Deep Belief Network (DBN) model demonstrated the ability to learn hierarchical feature representations

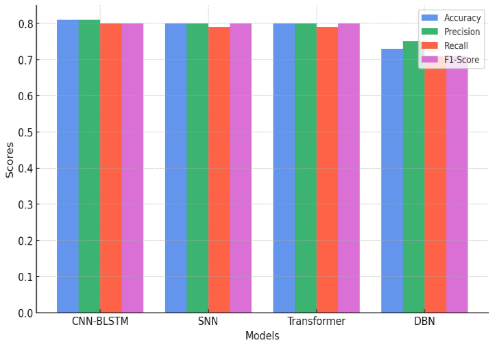


Figure 5. Comparative Performance of All Models

better than other models in our evaluation, but its accuracy was lower compared to all votes, and the Deep Belief Network (DBN) model demonstrated the ability to learn hierarchical feature representations better than other models, but its accuracy was lower than that of all votes. While the DBN's stacked Restricted Boltzmann Machines (RBMs) can capture a large amount of information, its recall rate is low, and it missed a few phishing ones. Future extensions with more layers or hybrid integration of other models may enhance the effectiveness of the DBN, which is designed for applications where resource efficiency has more priority over detection accuracy owing to its lower computing requirement compared with CNN-BLSTM and Transformer models [22].

From a feature-selection standpoint, it was necessary to optimize the model and enhance its efficiency. The Recursive Feature Elimination (RFE) approach is used to select relevant features to ensure that each model focuses [9], [23] on essential signals while ignoring noise. The structural attributes of phishing URLs, such as the number and length of subdomains and unusual characters, are more important to detect than the URL structure itself because our dataset contained numerous phishing domains that used normal tiny-like blogspots or WordPress names. CNN and Transformer models were able to successfully understand these spatial and sequential patterns [24]. In particular, when working with data that are not ordinally distributed, such feature selection helped to reduce the number of false positives in the DBN and SNN models [6]. The official-looking emails strung users to URLs. It had HTTPS, or IP addresses common indicators of playful-rendering phishing attacks. Thus, the model learned to identify URLs for hacking purposes without a learning process [25]. Take keyword analysis, for example, key phrases comprising a high percentage of attack websites can be discovered, as seen in an attention mechanism-focused use case like the transformer model on textual patterns [26].

This experiment proves the efficiency of the models compared with techniques such as Naïve Bayes and Logistic Regression [1]. Traditional machine learning models showed low-performance metrics with an accuracy of 61%, precision of 66%, and recall of Naïve Bayes of around (62%), signifying their limited efficacy in complex phishing detection settings. The design of conventional models restricts them from being adaptable to a type of phishing attack, owing to their predefined set features and rudimentary pattern analysis. On the other hand, deep learning models such as CNN-BLSTM and transformers are good at capturing more complex patterns without human-defined features. The performance of CNN-BLSTM, with an accuracy of 81%, substantially exceeded the Naïve Bayes algorithm [7].

The study showed that deep learning models successfully improve phishing detection by eliminating false positives and adapting quickly to new types of attacks. In this area, their ability to learn complex high-dimensional properties allows them flexibility in behaviors and yields more accurate and informed classification of samples. While deep learning is more compute-hungry, a comparison with traditional models reflects improvements.

Although the proposed models demonstrated considerable accuracy, precision, and recall, some challenges must be addressed for effective real-time phishing detection. Although powerful, CNN-LSTM and Transformer models require significant computing resources owing to their complex topologies. Multi-head attention layers in the transformer likely take longer to run and process sequentially through the data with a CNN-LSTM, which could introduce a lag for real-time systems. Although these models form an interpretable ensemble immediately, they are infeasible for real-time applications without speed optimization.

In such cases, simple models, such as SNN or optimized DBN, require far less computing power and could be more desirable. The SNN model achieves a favorable trade-off between the computational efficiency and precision. This allows real-time deployment. Real-time detection can be achieved with additional performance enhancements, such as model quantization or pruning, and can reduce the computational overhead [27].

Balancing speed and accuracy are key factors. In contrast, even though the Transformer and CNN-BLSTM models exhibit superior detection performance (Table 3), they are not fast enough to comply with the real-time demands of this type of system. Hybrid models that combine the precision of deep learning with improved efficiency from simple approaches can provide a workable solution. For instance, lightweight models may be used as an initial filter and can perform full CNN-BLSTM or Transformer analysis for flagged URLs. By balancing the trade-offs in terms of speed and detection efficacy, Own et al. proposed that this hybrid method would enable timely high-quality phishing identification.

5. CONCLUSION AND FUTURE WORK

This study concentrated on supplying a comparative evaluation of deep getting-to-know fashions for phishing attack detection using URL-primary-based functions. Four architectures—CNN-BLSTM, Transformer, Self-Normalizing Neural Network (SNN), and Deep Belief Network (DBN)—were implemented and evaluated using a balanced dataset of 50,000 URLs. Among these, the hybrid CNN-BLSTM version achieved the highest detection accuracy of 81%, demonstrating advanced ability. The Transformer version also produced robust results, leveraging self-attention mechanisms for complex pattern reputations. In contrast, the DBN version exhibited decreased consideration of phishing URLs, indicating barriers in identifying diverse phishing behaviors. The results verify that deep mastering methods outperform traditional machine-gaining knowledge of processes in phishing detection by improving the accuracy and decreasing fake positives. The study also highlighted exchange-offs between performance detection and computational performance, specifically in real-time situations. Despite these promising findings, several obstacles remain. The models, mainly CNN-BLSTM and transformers, require extensive computational resources, which may constrain actual-time deployment. The dataset focused on URL functions and did not include webpage content or e-mail metadata, which can similarly enhance detection. A light architecture adapted to promote future work will be created via the integration of real-time data from browser plugins or e-post clients, and will also be considered to improve the accuracy of the dynamic environment. Unfavorable training and interpretable AI techniques will be explored to enhance the robustness and interpretability of the model, ensuring safer and more transparent safety systems.

REFERENCES

[1] O. K. Sahingoz, E. Bube, and E. Kugu, "Dephides: Deep learning based phishing detection system," IEEE Access **12**, 8052–8070 (2024).

[2] Z. Alkhalil, C. Hewage, L. Nawaf, and I. Khan, "Phishing attacks: A recent comprehensive study and a new anatomy," Front. Comput. Sci. **3**, 563060 (2021).

[3] H. Tupsamudre, A. K. Singh, and S. Lodha, "Everything is in the name–a url-based approach for phishing detection," in *International Symposium on Cyber Security Cryptography and Machine Learning,* (Springer, 2019), pp. 231–248.

[4] A. O. Taofeek, "Development of a novel approach to phishing detection using machine learning," ATBU J. Sci. Technol. Educ. **12**, 336–351 (2024).

[5] A. V. Kulkarni and S. Nath, "Human susceptibility to social engineering attacks: An innovative approach to social change," in *2024 IEEE International Conference on Interdisciplinary Approaches in Technology and Management for Social Innovation (IATMSI),* vol. 2 (IEEE, 2024), pp. 1–6.

[6] V. Vajrobol, B. B. Gupta, and A. Gaurav, "Mutual information based logistic regression for phishing url detection," Cyber Secur. Appl. **2**, 100044 (2024).

[7] N. Q. Do, A. Selamat, O. Krejcar, *et al.*, "Deep learning for phishing detection: Taxonomy, current challenges and future directions," IEEE Access **10**, 36429–36463 (2022).

[8] R. van der Kleij, S. van 't Hoff—De Goede, S. van de Weijer, and R. Leukfeldt, "Social engineering and the disclosure of personally identifiable information: Examining the relationship and moderating factors using a population-based survey experiment," J. Criminol. **56**, 278–293 (2023).

[9] K. I. Ahmad, A. Paul, M. F. B. Hafiz, *et al.*, "A data-driven approach for online phishing activity detection," in *2024 International Conference on Advances in Computing, Communication, Electrical, and Smart Systems (iCACCESS),* (IEEE, 2024), pp. 1–6.

[10] S. Remya, M. J. Pillai, K. K. Nair, *et al.*, "An effective detection approach for phishing url using resmlp," IEEE Access (2024).

[11] S. Sawant, R. Savakhande, O. Sankhe, and S. Tamboli, "Phishing detection by integrating machine learning and deep learning," in *2024 11th International Conference on Computing for Sustainable Global Development (INDIACom),* (IEEE, 2024), pp. 1078–1083.

[12] P. Chinnasamy, P. Krishnamoorthy, K. Alankruthi, *et al.*, "Ai enhanced phishing detection system," in *2024 Third International Conference on Intelligent Techniques in Control, Optimization and Signal Processing (INCOS),* (IEEE, 2024), pp. 1–5.

[13] E. Kocyigit, M. Korkmaz, O. K. Sahingoz, and B. Diri, "Enhanced feature selection using genetic algorithm for machine-learning-based phishing url detection," Appl. Sci. **14**, 6081 (2024).

[14] J. G. Ponsam, "Url shield: Protecting users from phishing attacks using flask and ml," in *2024 3rd International Conference for Innovation in Technology (INOCON),* (IEEE, 2024), pp. 1–5.

[15] M. Shoaib and M. S. Umar, "Url-based phishing detection using machine learning," in *2023 6th International Conference on Information Systems and Computer Networks (ISCON),* (IEEE, 2023), pp. 1–7.

[16] N. Siva, B. V. Sivaiah, S. S. Reddy, *et al.*, "Phishing detection system through hybrid machine learning based on url," in *2024 5th International Conference for Emerging Technology (INCET),* (IEEE, 2024), pp. 1–5.

[17] D. T. Mosa, M. Y. Shams, A. A. Abohany, *et al.*, "Machine learning techniques for detecting phishing url attacks," Comput. Mater. Continua **75**, 1271–1290 (2023).

[18] S. Alkatheeri, N. Alhajeri, R. Alhashmi, and S. Kaddoura, "Classification of phishing webpages using supervised machine learning algorithms," in *2024 15th Annual Undergraduate Research Conference on Applied Computing (URC),* (IEEE, 2024), pp. 1–6.

[19] S. Pathan, O. Maddala, K. N. D. Saile, and P. Singh, "Phishing websites detection using machine learning," in *2024 2nd International Conference on Advancement in Computation & Computer Technologies (InCACCT),* (IEEE, 2024), pp. 29–33.

[20] G. Klambauer, T. Unterthiner, A. Mayr, and S. Hochreiter, "Self-normalizing neural networks," Adv. Neural Inf. Process. Syst. **30** (2017).

[21] A. Gillioz, J. Casas, E. Mugellini, and O. A. Khaled, "Overview of the transformer-based models for nlp tasks," in *2020 15th Conference on Computer Science and Information Systems (FedCSIS),* (IEEE, 2020), pp. 179–183.

[22] A. Basit, M. Zafar, X. Liu, *et al.*, "A comprehensive survey of ai-enabled phishing attacks detection techniques," Telecommun. Syst. **76**, 139–154 (2021).

[23] S. Mishra and D. Soni, "Smishing detector: A security model to detect smishing through sms content analysis and url behavior analysis," Future Gener. Comput. Syst. **108**, 803–815 (2020).

[24] T. Kattenborn, J. Leitloff, F. Schiefer, and S. Hinz, "Review on convolutional neural networks (cnn) in vegetation remote sensing," ISPRS J. Photogramm. Remote. Sens. **173**, 24–49 (2021).

[25] K. Thakur, M. L. Ali, M. A. Obaidat, and A. Kamruzzaman, "A systematic review on deep-learning-based phishing email detection," Electronics **12**, 4545 (2023).

[26] E. Benavides-Astudillo, W. Fuertes, S. Sanchez-Gordon, *et al.*, "A phishing-attack-detection model using natural language processing and deep learning," Appl. Sci. **13**, 5275 (2023).

[27] P. Kalaharsha and B. M. Mehtre, "Detecting phishing sites–an overview," arXiv preprint arXiv:2103.12739 (2021).