DL-Phish: Optimized Features Learning for Phishing URL Detection Using Deep Neural Networks

Mana Saleh Al Reshan [1,2]

[1] *Department of Information System, College of Computer Science and Information Systems, Najran University, Najran 61441, Saudi Arabia*
[2] *Emerging Trends and Technologies Lab (ETRL), College of Computer Science and Information Systems, Najran University, Najran 61441, Saudi Arabia*

Abstract **– Phishing persists as a serious cybersecurity concern in which consumers are tricked into divulging private information by using phony websites. It is necessary to accurately and competently identify such hazardous links in order to secure the internet environment. Because Deep Learning (DL) approaches can automatically learn complex patterns, and proved to be effective tools for identifying such attacks. Five DL models were tested in this study using a dataset gathered from Kaggle: Recurrent Neural Network (SimpleRNN), Long Short-Term Memory (LSTM), Multi-Layer Perception (MLP), Conventional Neural Network (CNN), and Gated Recurrent Unit (GRU). The most useful URL attributes were selected using Random Forest-based feature significance techniques and Chi-Square feature selection to maximize the model's efficiency. To achieve the global performance study, the models were evaluated using a variety of assessment metrics, including accuracy, precision, recall, and F1 score, with the help of loss graphs and confusion matrices. LSTM successfully proved its effectiveness in dealing with sequential patterns in phishing URLs with a highly accurate result of 98.80%. The DL models given in the paper, especially the recurrent neural networks, performed significantly better compared to the highest standard of accuracy and reliability established in the previous papers. The experimental results confirm that a reliable method for identifying phishing URLs can be formed using recurrent DL and effective feature selection.**

Keywords **– Phishing URL, cyber threat, DL models, feature selection.**

DOI: 10.18421/TEM153-15
https://doi.org/10.18421/TEM153-15

Corresponding author: Mana Saleh Al Reshan,
Department of Information System, College of Computer Science and Information Systems, Najran University, Najran 61441, Saudi Arabia
Email: msalreshan@nu.edu.sa

Received: 27 July 2025.
Revised: 21 December 2025.
Accepted: 07 May 2026.
Published: 27 August 2026.

1. Introduction

The popularity of smartphone technology, smart sensors, and Internet of Things leads the Internet to be part of everyday life. With the help of trusted infrastructure and access to a huge target market, the online business environment has gained momentum. However, risks such as social engineering are rapidly multiplying, thus the modern art of networking is also posing a threat to cybersecurity defense systems. Social engineers, phishers, and computer hackers use social engineering, which is a dynamic and rapidly evolving process, based more on psychological tricks than technical vulnerability [1]. It affects people, organizations, and business entities, coercing them into disclosing personal info and doing activities that serve cyber offenders. Social engineering is the fastest and most malicious attacks method, despite advance defenses for instance firewalls, intrusion detection systems, encryption, and antivirus. Social engineering is a huge global cybersecurity issue, as the scheme aims to get past the defense systems used by fraudsters and access personal information [2]. Among the several types of social engineering, phishing attacks can be transmitted via a range of paths, such as malicious URLs, emails, DNS spoofing, domain spoofing, and network exploitation [3]. The past few years have seen the use of DL being a huge success in dealing with the current complexities of cybersecurity, especially for identifying the sophisticated and dynamically changing varieties of cyberattacks such as phishing.

Unlike conventional machine learning approaches, where feature extraction takes time, DL models can learn hierarchical/abstract representations directly from raw data itself [4]. Owing to this strength, such methods can be applied to phishing detection, wherein the pattern can be weak or present in the form of strings, such as webpages or URLs. Although CNN, MLP architectures have excellent pattern recognition capability, the LSTM/GRU model is exceptionally effective in modeling relationships that are sequential. These DL models can prove to be an effective tool for cybercrime initiatives as these tools can be used for the development of improved phishing detection systems by reducing false positives with the help of DL models [5].

The rising sophistication of phishing attacks raises a great threat to the long-standing security measures, as those security methods fail to identify and prevent these sophisticated phishing attacks effectively. There is an urgent need for advanced and accurate methods for fast detection of phishing URLs and other phishing attack vectors prior to causing damage [6], [7].

To enhance detection, this research work will compare and analyze the efficacy of various DL models in detecting phishing URLs using advanced FS techniques. Through the provision of an accurate framework that can effectively determine phishing URLs, this research work will contribute significantly to research in cybersecurity. Through the focus on using sequential DL models along with an appropriate feature selection, this research work will provide tangible guidelines on developing intelligent systems that can effectively resist phishing attacks. Finally, this research will aim to safeguard individuals as well as businesses from phishing attacks that result in economic losses, data theft, and reputation crises.

The reminder of the paper is organized in following sections: Section II discuss key research gap identified various research work. In section III the complete methodological pipeline of the proposed study is discussed. Section IV interprets key findings from experiments and discussion. Section V concludes main findings and provides necessary future directions.

2. Related Work

Over two million URLs, a research study [8] indicated that a CNN-based phishing URL detector using raw URLs without any predefined criteria reached an accuracy of approximately 96%. However, as phishing activities continue to target IoT devices due to their widespread usage as well as their handling of confidential data.

An optimization of feature selection for an IoT with a combination of a random forest classifier, reached an accuracy of 99.57% in detection as discussed in [9]. Phishing activities continue to rise as an evolving source [10]. A thorough categorization of the detection of phishing using HTML and URL-related features has been implemented. This describes the characteristics of deep learning techniques in terms of preprocessing, feature selection, architecture, as well as performance, illustrating the loopholes and strengths of existing techniques [11]. Since phishing activities target user deception in terms of web pages, detection techniques continue to develop using DL techniques, irrespective of the ordering of words. However, this approach fails in considering the semantic aspects. This research [12] intended to develop a novel method of detecting phishing using the text aspect of web pages, instead of URLs, using NLP as well as GloVe embedding techniques. Among the tested four deep learning architectures, Bidirectional Gated Recurring Units resulted in highest detection accuracy of 97.39%, considering semantic aspects. Advanced phishing activities target confidential information using URLs. Since these activities are sophisticated, imitating actual URLs. To overcome this problem, a novel approach for Bidirectional Encoder Representations from Transformers (BERT) technique in combination with a deep CNN architecture in classification was introduced. When applied on a phishing site prediction dataset of 549,346 URLs in three scenarios, this research resulted in a detection rate of 96.66%, outperforming previous techniques [13]. Although previous research on email phishing detection techniques do exist, these techniques fail in terms of high detection accuracy as well as fewer FPR. To overcome this problem, this research introduces novel techniques in terms of a1D-CNNPD-based model, as well as its expansions using recurring layers in terms of LSTM, L-Bi, L-GRU, as well as L-Bi GRU. The performance evaluation of these extended models over the phishing Corpus dataset as well as the Spam Assassin dataset showed improvement over the base model. The bi-GRU variant among these models performed best in terms of accuracy (99.68%), precision (100%), F1-score (99.66%), and recall (99.32%). These results demonstrate the aptness of hybrid CNN-RNN models in identifying phishing mails effectively and support their adoption in practical applications of security relevant to phishing attacks [14]. The growing number of phishing sites is causing a rise in cyber threats to people and organizations in today's world. Due to this, there is an emerging need for more efficient phishing detection (PD) solutions. The authors propose a CNN-based model that accurately distinguishes phishing websites using URL attributes from the PhishTank dataset.

A balanced dataset of 10,000 benign and 10,000 phishing URLs were tested the model using typical metrics. The DL model with seven layers consisting of convolutional, pooling, and dense layers performed with 98.77% accuracy, outperforming other models. Comparative outcomes indicate that CNN performs better than K-Nearest Neighbor (KNN) (87%), Natural Language Processing (NLP) (97.98%), Recurrent Neural Network (RNN) (97.4%), and Random Forest (RF) (94.26%), owing to deeper structure, bigger training data, and richer feature extraction [15]. Phishing is a major global cybercrime targeting users' sensitive information through websites, cloud platforms, and government portals. Many users remain unaware of such threats while browsing. Existing phishing detection methods, especially for email attacks, remain limited. This study proposes a three-step content-based approach using URL, web traffic, and content features to detect phishing attacks.

Data collected from recent phishing attacks enhances accuracy. Three classification techniques, namely SVM, Neural Network (NN), and Random Forest (RF), have been tested, obtaining a success rate of 95.18%, 85.45%, and 78.89%, respectively. The efficiency of machine learning in phishing detection is proved by the experimental outcome [16].

3. Methodology

This section provides an elaborate description of the overall pipeline used for phishing URL detection, which includes dataset pre-processing, feature engineering and selection, data splitting and imbalance management, deep learning model definitions, and the fusion ensemble strategy as depicted in Figure 1.

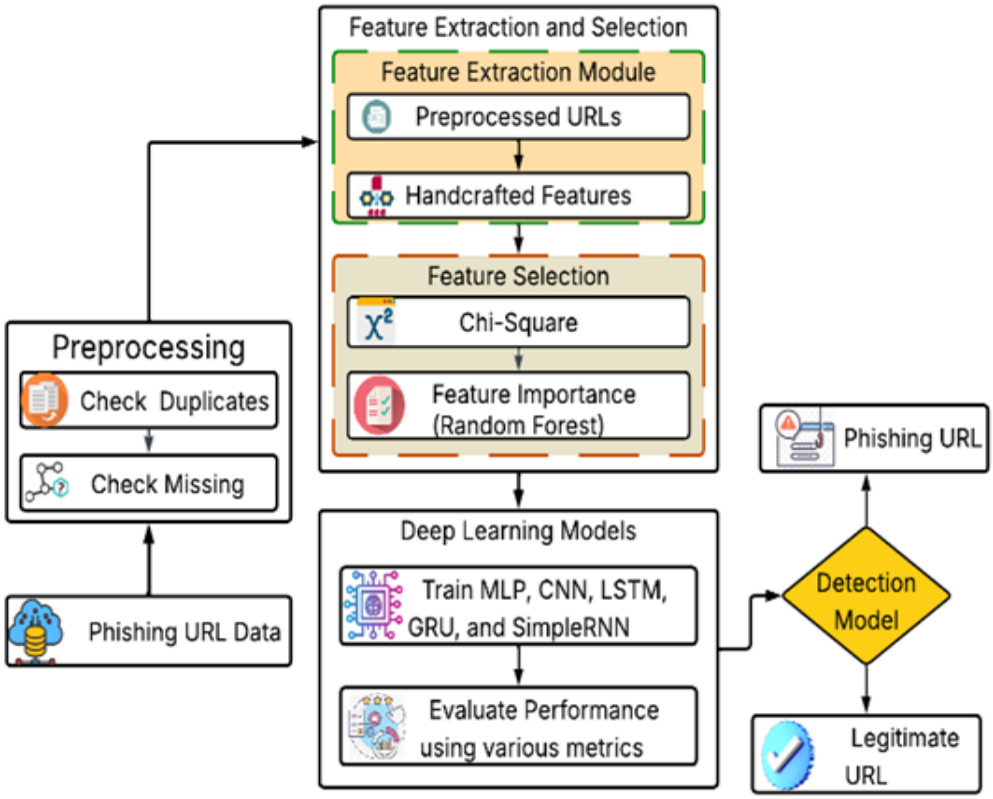

Figure 1. Proposed research methodology

3.1. Dataset and Pre-Processing

For the experimental process, a publicly available dataset containing a list of URLs, some of which were labeled as legitimate or phishing, was employed. Prior to the commencement of the analysis, the list of URLs underwent preprocessing.

3.2 Feature Engineering and Selection

To transform raw URLs into discriminative features for classification, hand-crafted lexical features that capture structural and security-related patterns are extracted (Table 1). These features include:

• Simple URL Structure: URL Length, Domain Length, and Number of Subdirectories are used as proxies for URL complexity.

• Security Indicators: HTTPS Indicator reflects the presence of encryption, typically associated with legitimate sites. IP Address Indicator flags the use of raw IPs, which are more common in phishing URLs.

• Obfuscation Indicators: Dot Count excessive dots (e.g., example.tk.tk) can indicate domain manipulation. The symbol is rarely used in legitimate URLs, often to mislead users. Shortening Service Usage hides the actual destination URL, commonly seen in malicious redirection [17]. These features are lightweight and faster to compute compared to more resource-intensive NLP-based methods (e.g., tokenization, embeddings). While effective for capturing structural patterns, lexical features may fail to detect semantic cues embedded in URLs. However, offer a favourable trade-off between performance and interpretability.

Table 1. Hand-engineered lexical features

Features	Description
URL Length	Total length in characters.
IP Address Presence	Tells if the URL has an IP address.
Dot Count	Number of "." characters within the URL.
At Symbol	If the "@" symbol is utilized.
HTTPS Presence	Binary flag for secure protocol presence.
Subdirectory Count	Based on occurrences of "/".
Domain Length	The length of the netloc in the URL.
Shortening Service Usage	Marks utilization of services such as bit.ly, t.co, etc.

3.3 Feature Selection

To improve model performance and limit overfitting, two statistical feature selection methods were used to select only the most discriminant features: the Chi-Square test and Random Forest feature importance. Both methods picked the top 5 features from their respective criterion.

1. ***Chi-Square Test:*** The Chi-Square test is a statistical hypothesis test used to assess the independence between categorical variables (attributes) and the target class. It assesses whether there is a significant difference between the observed frequency distribution of the categorical attributes and their expected distribution under an independent model. The chi-square statistic for feature f is calculated using equation (1).

$$X^2 = \sum_{i=1}^{k} \sum_{j=1}^{c} \frac{(O_{ij} - E_{ij})^2}{E_{ij}} \quad (1)$$

Where,

- O_{ij}: Represents the observed frequency count of category i of features f with class labels j
- E_{ij}: Represents the expected count of frequency assuming independence.
- k: Depicts the number of feature categories
- c: Shows the number of class labels.

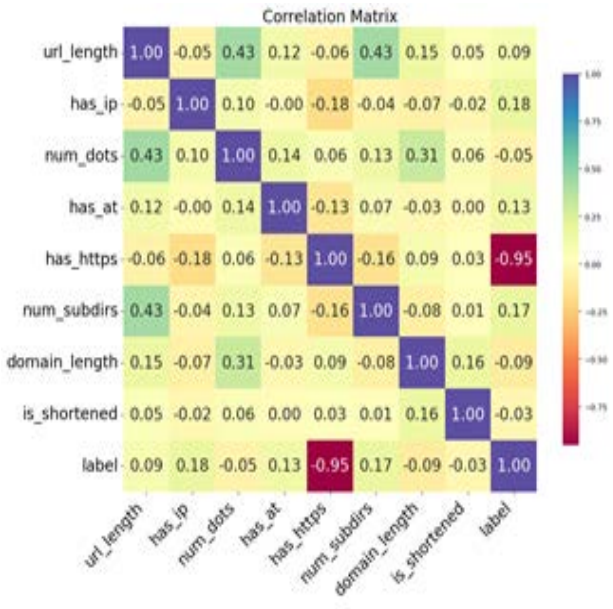

Figure 2. Correlation matrix for best feature

Figure 2 depicts the correlation matrix that reveals linear pairwise relationships between URL-based features and the target class. The existence of strong negative correlation is found for has_https and the target variable (-0.95), establishing its high discrimination ability. Moderate positive correlations are found for has_ip (0.18) and num_subdirs (0.17), whereas other features like url_length and num_dots show low but significant correlations. The Chi-Square Test used these dependencies and selected the statistically significant features for model training based on their properties of being non-linear or categorical, like has_https, as opposed to purely linear correlations.

2. ***Random Forest Importance:*** A Random Forest algorithm is a type of ensemble algorithm, which employs the use of decision trees. Random Forest calculates the importance of the features based on certain impurity measures, Gini Index, or Information gained. Suppose there is a certain set of features F, where the importance score can be calculated based on the cumulative reduction in impurity at the nodes brought about by the splits within the feature F. Mathematically, the importance score I(F) of features set F can be given by equation (2).

$$I(F) = \frac{1}{T} \sum_{t=1}^{T} \sum_{n \epsilon N_{t(F)}} \frac{W_{i \cdot \Delta i_n}}{\sum_{n \epsilon N_t} W_n} \quad (2)$$

Where,

- T: denotes the trees
- $N_{t(F)}$: shows the set of nodes where feature set F is used for splitting in tree t
- Δi_n: Represents the impurity decrease at node n
- W_n: Denotes the weighted number of samples reaching node n.

Random Forest models nonlinear and variable interactions and thus can be regarded as a strong wrapper technique to uncover the effective predictive capability of the variable. This technique can handle both continuous and categorical variables and does not suffer from the presence of noisy or irrelevant features as in the case of Chi-Square.

Table 2. Feature importance

Features	Importance
has_https	0.752566
num_dots	0.147577
url_length	0.034313
domain_length	0.030011
num_subdirs	0.016054
has_ip	0.014145
has_at	0.004602
is_shortened	0.000733

Table 2 highlights the list of ranked features based on the importance scores. Importantly, the most discriminative feature, based on an importance score of 0.753, involves *has_https*; this indicates that URLs using HTTPS are mostly benign. Moreover, the number of dots in URLs, as indicated in num_dots (0.148 importance score), also plays an important role, since many dots in URLs could indicate that the URL is attempting to hide its actual nature. Then, url_length with an importance score of 0.034 follows as one of the subsequent crucial elements. Features with importance scores less than 0.03 do not contribute significantly or at all in the classification stage. Features that receive importance scores of less than 0.03, such as has_at and is_shortened, do not, in practice, contribute significantly in classifying URLs as benign or malicious. Feature importance scores can be obtained from tree-based classifier importance, such as in the case of Random Forest classifiers that work effectively in handling nonlinear data.

3.4 Data Splitting and Imbalance Handling

The data was split into training (70%}, validation (15%}, and testing (15%) data sets, stratified to preserve the original class distribution, as shown in Table 3.

Table 3. Division of samples

Training Samples	Testing Samples	Validation Samples
315111	67527	67538

The class distribution prior to the use of the Synthetic Minority Over-Sampling (SMOTE) technique is presented in Figure 3.

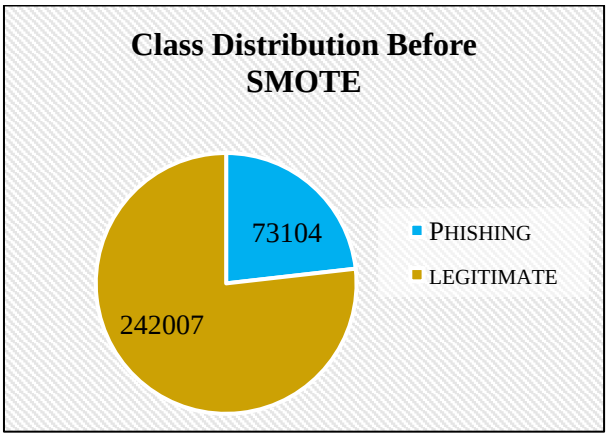

Figure 3. Class distribution before SMOTE

Figure 3 shows the class distribution. It is an imbalanced dataset, as there are 73,104 phishing samples against 242,007 legitimate samples, yielding a proportion of approximately 1:3.3. The ones that result from such a model are biased towards the majority class; this inflates the overall accuracy and damagingly compromises the model's objective of detecting phishing attacks. Because of this, phishing recall often declines, entailing a higher rate of false negatives in which actual phishing attempts go undetected. The problem of class imbalance can be significantly handled through the SMOTE technique. SMOTE was used to synthetically generate 169,000 new phishing samples, which further led to a perfectly balanced dataset comprising 242,000 instances of each class. A balanced class distribution reduces model bias toward the majority class and significantly improves the model's ability to detect phishing attacks by improving recall without sacrificing precision.

3.5 Deep Learning Architectures

Five deep learning architectures were developed and evaluated using the selected features:

• **Multilayer Perceptron (MLP):** A fully connected feedforward neural network comprising two hidden layers.

• **Convolutional Neural Network (CNN):** A one-dimensional convolutional model applied to feature sequences, designed to capture local temporal patterns.

• **Simple Recurrent Neural Network (RNN):** A basic recurrent model that captures short-term dependencies in sequential data.

• **Long Short-Term Memory (LSTM):** An enhanced RNN variant capable of learning long-range temporal dependencies through memory cells and gating mechanisms.

• **Gated Recurrent Unit (GRU):** A streamlined alternative to LSTM that maintains comparable performance with fewer parameters.

All models using 50 epochs with Adam optimizer and binary cross-entropy loss function were trained. Performance metrics were monitored on the validation set to guide training and assess generalization.

4 Results

The experiments in this study were performed on Windows 10. Table 4 interprets the performance evaluation of five different DL models.

Table 4. Performance evaluation

DL Models	Accuracy	Precision	Recall	F1 Score
MLP	98.63%	94%	93%	93%
CNN	98.73%	92%	91%	91%
SimpleRNN	98%	96%	95%	95%
LSTM	98.56%	95%	96%	95%
GRU	98%	96%	97%	96%

The MLP achieved high accuracy (97.63%) but had slightly lower recall (93%) due to its inability to capture temporal dependencies. Similarly, CNN reached 97.73% accuracy but struggled with sequential patterns, resulting in a lower recall (91%). SimpleRNN improved recall (95%) and accuracy (98%) by processing sequential data, though it remained limited in learning long-range dependencies. LSTM outperformed all models with 98.56% accuracy and 96% recall, owing to its gated architecture that effectively captures long-term patterns crucial for reducing false negatives. GRU matched LSTM's performance (98% accuracy, 97% recall) while being computationally more efficient. Overall, LSTM and GRU were best suited for phishing detection, with GRU offering a strong balance between effectiveness and efficiency.

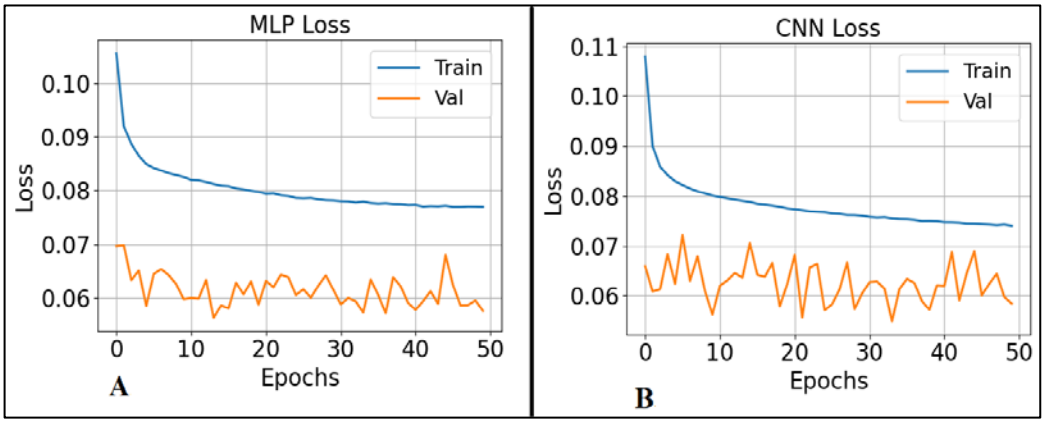

Figure 4. Loss graph for DL Models

Figure 4 shows the training and validation loss of DL. Figure 4 (A) depicts the performance of the MLP model which shows a gradual reduction in training and validation loss, approaching ~0.06 with negligible variance, reflecting good generalization without overfitting. Smooth optimization implies well-optimized hyperparameters (e.g., regularization, learning rate) and appropriateness for non-spatial data. While in Figure 4 (B) the performance of the CNN model is shown.

In contrast, the CNN exhibits sudden initial convergence close to 0.06 in a few epochs by its inductive biases (weight sharing, local connectivity) for structured data such as images. The narrow train-val gap supports good generalization. Both models obtain low variance and bias, but the faster convergence of CNN emphasizes its strength for grid-like inputs, while the slow optimization of MLP is well matched with easier tasks. Neither overfits, indicating proper capacity nor regularization.

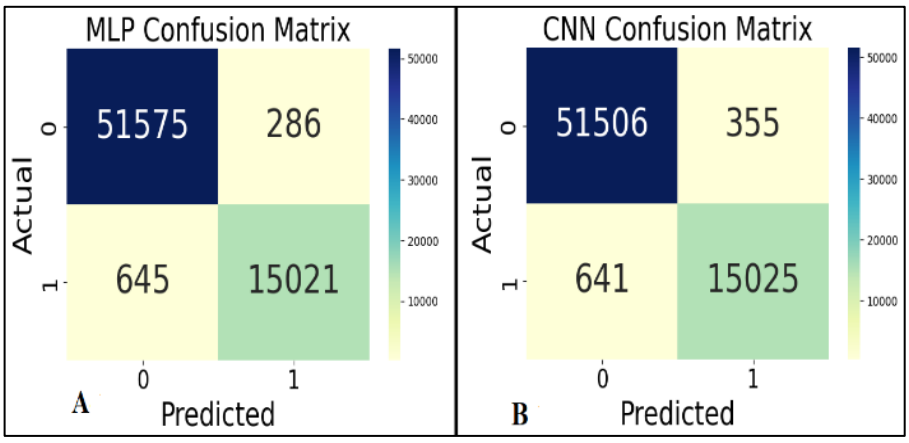

Figure 5. Confusion Matrix for DL models

Figure 5 represents the performance of MLP and CNN models over unseen test data using confusion matrices. In Figure 5 (A) the MLP model correctly classified 66,596 out of total samples (67527) with 931 misclassifications demonstrating high predictive accuracy. Similarly, Figure 5 (B) shows the CNN model's confusion matrix, with 66,531 correct predictions and 996 misclassifications.

Both models show strong generalization and balance performance across classes. MLP shows slightly better overall accuracy with more conservative boundary, while CNN is marginally better at detecting positive case.

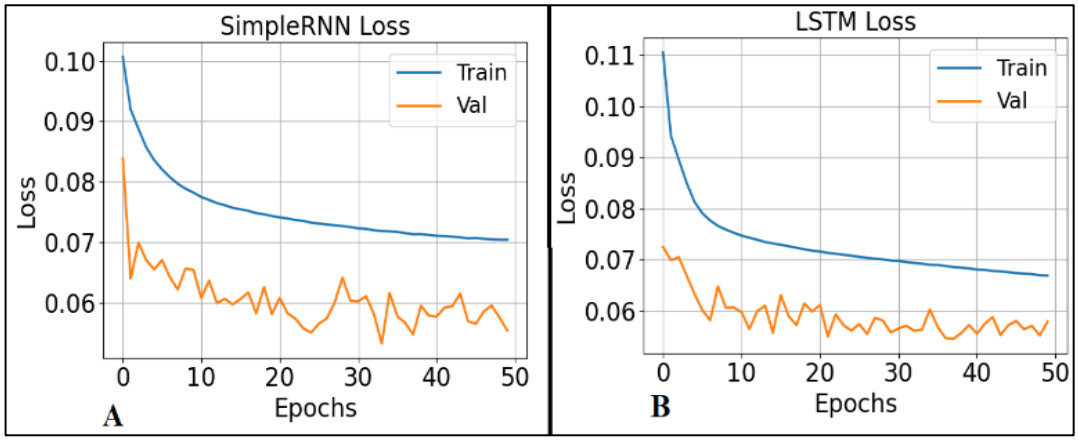

Figure 6. Loss graph for DL models

Figure 6 illustrates the training and validation loss trends for the SimpleRNN and LSTM models. In Figure 6 (A), the SimpleRNN loss curves show a steady decrease across epochs, with validation loss consistently lower than training loss indicating good generalization and the absence of overfitting. The smooth training curve further reflects stable model optimization.

In contrast, Figure 6 (B) presents the LSTM model's loss curves, which also decline smoothly but begin with a slightly higher initial loss. Overall, both models perform effectively, but the LSTM demonstrates greater learning efficiency and generalization capability.

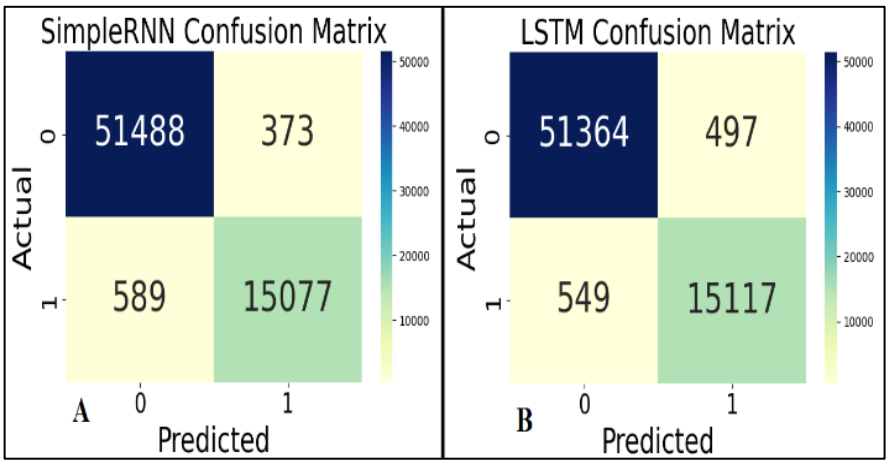

Figure 7. Confusion Matrix for DL models

Figure 7 presents the confusion matrices of the SimpleRNN and LSTM models on the unseen test data. In Figure 7 (A), the SimpleRNN model achieves 66,565 correct predictions out of 67,527 samples, with only 962 misclassifications indicating strong classification accuracy and robust generalization. Similarly, Figure 7 (B) illustrates the LSTM model's results, yielding 66,481 correct classifications and 1,046 misclassifications.

While both models demonstrate effective performance and balanced prediction across classes, the SimpleRNN marginally outperforms LSTM in terms of total correct predictions. This suggest that SimpleRNN's simpler architecture leads to more conservative decision boundary and fewer classifications, while LSTM's higher capacity may cause overfitting when temporal dependencies are simpler.

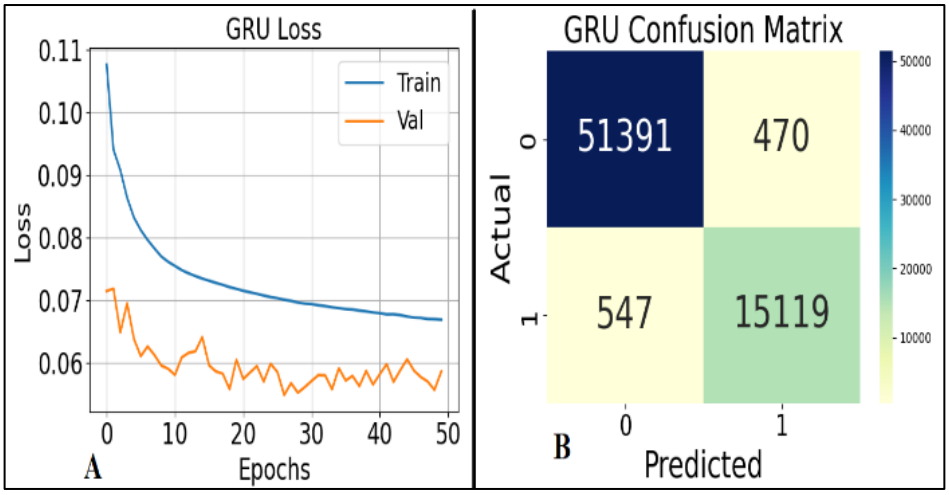

Figure 8. Performance evaluation of GRU

Figure 8 demonstrates the model learning process as well as prediction accuracy of the GRU model. In Figure 8 (A), there is steady improvement in model performance as measured on both training and validation sets. It is also important to note that there is an indication of lower validation loss than in training over all epochs, thus less overfitting. The model learning curve also appears quite regular in Figure 8 (A), indicating an optimized learning process. Figure 8 (B) presents an indication of performance as measured using an unrevealed test dataset in which the GRU model was able to predict correctly 66,510 out of a total of 67,527 samples while failing only 1,017 times.

GRU's gating mechanism technically allows it to learn effectively from relevant temporal relationships in their models more than LSTMs while using less architecture.

4.1 Compartive Study and Discussion

One of the objectives of this study is to compare the performance of models in terms of accuracy and compared the performance of proposed models with the existing models (Table 5).

Table 5. Comparison between existing and proposed models

Study	DL Models	Dataset	Accuracy
[18]	CNN	Phishing URLs	98%
[19]	MLP	Phishing Websites/URLs	98.4%
[20]	CNN, LSTM, IPDS (CNN+LSTM)	Phishing URLs	CNN = 92.55% LSTM = 92.79% IPDS = 93. 28%
Proposed Study	CNN, MLP, LSTM, GRU, and SimpleRNN	Phishing URLs	CNN =98.73% MLP =98.63% GRU =98% LSTM =98.80% SimpleRNN =98%

The performance of DL models is explained through Figure 9, which clarifies the effectiveness of optimized models of Sequential Data in phishing scenarios.

The highest accuracy for this experiment was demonstrated by LSTM at 98.73%, clearly asserting its ability to learn long-term dependencies using Gated Memory Units - an integral requirement in learning temporal properties in phishing datasets like URLs and texts. GRU also closely trailed behind with an accuracy of 98.63%.

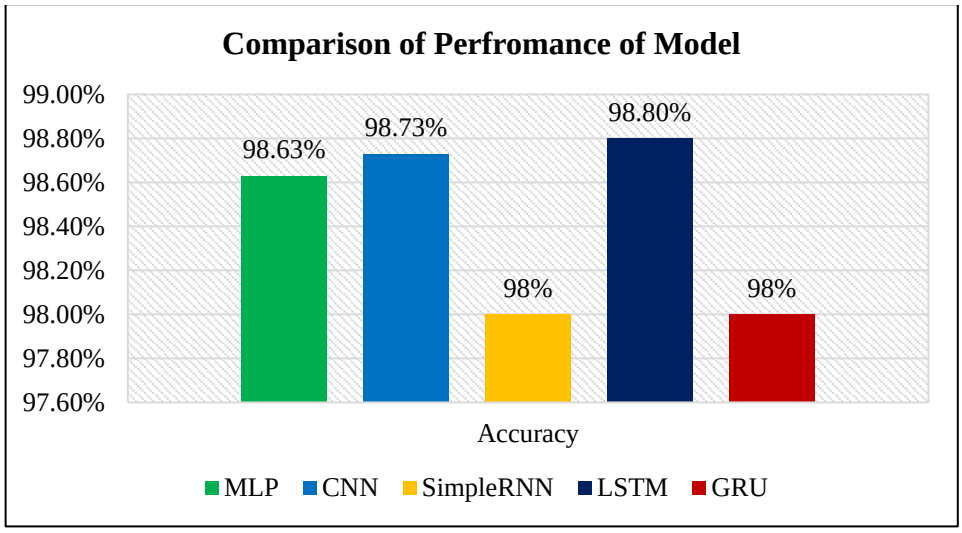

Figure 9. Comparative analysis of the DL model

Although only slightly less in terms of accuracy than LSTMs, GRU is more computationally optimized in terms of model effectiveness, which is an important criterion for implementation in resource-depleted scenarios. SimpleRNN reported an accuracy of 98%, which although fair, clearly depicted some inadequacies in learning complicated temporal relationships in phishing datasets possibly due to an optimized architecture scheme devoid of Gated Units. Non-sequential based models like CNN (Accuracy = 97.80%) and MLP (Accuracy = 97.60%) reported slighly lower accuracies.

Although very capable in learning spatial representations through CNNs and dense learning across MLPs, these models clearly depicted inadequacies in learning time-dependent properties that play an important role in phishing scenarios. The choice between them should consider the trade-off between maximum accuracy and computational efficiency, especially in large-scale or real-time security applications.

Furthermore, the proposed models were compared with the state-of-the-art models as depicted in Table 5. Several research studies have explored different DL models for phishing URL detection. Study [18] employed a CNN and achieved a notable accuracy of 98% using a phishing URL dataset. Study [19] used an MLP on phishing websites/URLs, achieving a slightly higher accuracy of 98.4%. Study [20] implemented a combination of CNN, LSTM, and a hybrid model named IPDS (CNN + LSTM) on a phishing URL dataset. The reported accuracies were CNN = 92.55%, LSTM = 92.79%, and IPDS = 93.28%.

This line of inquiry is furthered in the proposed study, which compares the performance of five DL models: CNN, MLP, LSTM, GRU, and SimpleRNN using a phishing URL dataset. Compared to previous studies, the suggested models showed a considerable improvement in accuracy, with LSTM achieving 98.80% across all architectures.

This demonstrates how recurrent models are more successful at identifying sequential patterns that are pertinent to phishing detection.

5 Conclusion

This work presents a comprehensive comparison of DL models, including CNN, MLP, LSTM, GRU, and SimpleRNN, for the detection of phishing URLs. The proposed approach successfully focused on the most important features of the phishing URLs by employing effective feature selection techniques, including Chi-Square analysis and feature importance from a Random Forest model. This preprocessing step was the main factor behind the outstanding performance of sequential models, especially LSTM, which topped them with an accuracy of 98.80%. These results confirm the importance of incorporating deep learning with robust feature engineering methods to develop accurate and efficient anti-phishing systems. Clearly, the experimental results confirm that a sequential model is highly appropriate in extracting temporal and contextual patterns encoded in the URLs and ignored by typical machine learning methods. By confirming the efficiency of both GRU and SimpleRNN architectures, the contribution of temporal pattern recognition has been further emphasized for anti-phishing applications. These results clearly lead to the conclusion that the combination of robust feature engineering methods with deep learning architectures will definitely contribute to the development of accurate and efficient anti-phishing tools.

The proposed approach offers higher detection accuracy, fewer false warning, and adaptability to evolving phishing tactics, but also has some limitations. Future research could investigate hybrid architectures and field deployments to further enhance detection. The research mostly used features that are non-real-time related to URLs and is in an offline environment. This could be improved in the future for added robustness against potential threats by considering the use of real-time data streams. Another approach for improvement may be the conduct of real-world implementations for further evaluation.

References:

[1]. Chetioui, K., et al. (2022). Overview of social engineering attacks on social networks. *Procedia Computer Science*, *198*, 656-661.

[2]. Akeiber, H. J. (2025). The evolution of social engineering attacks: A cybersecurity engineering perspective. *Al-Rafidain Journal of Engineering Sciences*, 294-316.

[3]. Alghenaim, M. F., et al.. (2022). Phishing attack types and mitigation: A survey. *The International Conference on Data Science and Emerging Technologies*, 131-153. Springer Nature Singapore.

[4]. Benavides, E., et al. (2019). Classification of phishing attack solutions by employing deep learning techniques: A systematic literature review. *Developments and Advances in Defense and Security: Proceedings of MICRADS 2019*, 51-64.

[5]. Ahmad, S., et al. (2024). Across the spectrum in-depth review AI-based models for phishing detection. *IEEE Open Journal of the Communications Society*, *6*, 2065-2089.

[6]. da Silva, C. M. R., Feitosa, E. L., & Garcia, V. C. (2020). Heuristic-based strategy for Phishing prediction: A survey of URL-based approach. *Computers & Security*, *88*, 101613.

[7]. Sánchez-Paniagua, M., et al. (2022). Phishing URL detection: A real-case scenario through login URLs. *IEEe Access*, *10*, 42949-42960.

[8]. Al-Alyan, A., & Al-Ahmadi, S. (2020). Robust URL Phishing Detection Based on Deep Learning. *KSII Transactions on Internet & Information Systems*, *14*(7).

[9]. Bustio-Martínez, L., et al. (2022). A lightweight data representation for phishing URLs detection in IoT environments. *Information Sciences*, *603*, 42-59.

[10]. Alabdan, R. (2020). Phishing attacks survey: Types, vectors, and *technical* approaches. *Future internet*, *12*(10), 168.

[11]. Asiri, S., et al. (2023). A survey of intelligent detection designs of HTML URL phishing attacks. *IEEe Access*, *11*, 6421-6443.

[12]. Benavides-Astudillo, E., et al. (2023). A phishing-attack-detection model using natural language processing and deep learning. *Applied Sciences*, *13*(9), 5275.

[13]. Elsadig, M., et al. (2022). Intelligent deep machine learning cyber phishing url detection based on bert features extraction. *Electronics*, *11*(22), 3647.

[14]. Altwaijry, N., et al. (2024). Advancing phishing email detection: A comparative study of deep learning models. *Sensors*, *24*(7), 2077.

[15]. Aldakheel, E. A., et al. (2023). A deep learning-based innovative technique for phishing detection in modern security with uniform resource locators. *Sensors*, *23*(9), 4403.

[16]. Mohamed, G., et al. (2022). An effective and secure mechanism for phishing attacks using a machine learning approach. *Processes*, *10*(7), 1356.

[17]. Haynes, K., Shirazi, H., & Ray, I. (2021). Lightweight URL-based phishing detection using natural language processing transformers for mobile devices. *Procedia Computer Science*, *191*, 127-134.

[18]. Singh, S., Singh, M. P., & Pandey, R. (2020). Phishing detection from URLs using deep learning approach. *2020 5th international conference on computing, communication and security (ICCCS)*, 1-4.

[19]. Saha, I., et al. (2020). Phishing attacks detection using deep learning approach. *2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT)*, 1180-1185.

[20]. Adebowale, M. A., Lwin, K. T., & Hossain, M. A. (2023). Intelligent phishing detection scheme using deep learning algorithms. *Journal of Enterprise Information Management*, *36*(3), 747-766.