International Journal of Applied Mathematics

Volume 38 No. 5s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

DESIGN OF AN INTELLIGENT DEEP LEARNING FRAMEWORK FOR PHISHING URL DETECTION IN CYBERSECURITY APPLICATIONS

G.Lingaiah[1] , Dr.Ram Kinkar Pandey[2]

[1]Research Scholar,

Dept. of Computer Science and Engineering,

Arni University, Himachal Pradesh, India. .

lingaiah.g80@gmail.com

[2]Professor, Dept. of Computer Science and Engineering,

Arni University, Himachal Pradesh, India

Dr.ramkpandey@gmail.com

Abstract

Phishing is the ubiquitous cyber threat from social engineering and web-deception to compromise user security, which may lead to credential theft, financial loss and data breaches. Originally blacklist-based detection methods cannot be used since the nature of phishing attacks varies and has changed frequently. We present our novel deep learning methodology for phishing URL detection based upon the publicly available PhiUSIIL dataset of large-scale lexical, host-based, content-based URL features in this paper. The new model employs sophisticated feature engineering, correlation analysis, and deep learning techniques to estimate discriminative patterns between benign and malicious URLs. This tool was implemented by a multi-layered deep neural network optimized with Adam optimizer and binary cross-entropy loss with dropout regularization to avoid overfitting. There was an extensive study of the model on the other model classesifiers of machine learning called Logistic Regression, Random Forest and Support Vector Machines. The results show that the deep learning model stands apart from the base models as well with 99.4% accuracy, 98.1% precision, 97.9% recall, 98.0% F1-score, and a strong generalization in absence of test samples. This confirms that the intelligent deep learning model proposed provides ascalable, adaptive, and robust method for phishing URL detection. This framework can be applied to any security applications in real-world such as browser extensions, email filters, and intrusion detection systems to actively defend against phishing threats. For future extensions, hybrid deep learning models will be studied, explained AI processes, and deployment in real-time detection.

Keywords: Phishing detection, Deep learning, Cybersecurity, CNN, LSTM, Transformers, URL classification

I.Introduction

Phishing has become one of the most pressing cybersecurity challenges facing the digital world, exploiting technological and human psychology. In shambles, malicious sites and URLs

Received: August 06, 2025

Volume 38 No. 5s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

disguise malicious websites and services as legitimate internet services, infiltrators trick unsuspecting users into divulging sensitive information like passwords, financial records, or personal identification information. The APWG, 2022, reported the exponential increase in phishing cases with millions of phishing websites targeted each year – one of the most serious and fatal cybercrime forms found, and it was a threat to phishing. The phishing is social engineering; not a bug that can be easily detected, but something that can be difficult to catch and thus impossible to detect by means of technical defenses (Hong 2012).

Conventional anti-phishing strategies relie on blacklists and heuristic identification. Blacklists have a large repository of infamous malicious URLs or domains. Although they are straightforward, they are inherently reactive and cannot identify zero-day or newly launched phishing websites that are not already listed in the database (Moore & Clayton, 2007). In contrast, heuristic methods detect phishing URLs by setting a standard criterion, like URL length, unique character characters, or suspicious token. Regardless of this, these strategies are more adaptable, generate a large number of false positives, require constant manual updates, and can't work on scale with the ever-increasing number of phishing campaigns (Aburrous et al., 2010).

Against this backdrop, research has increasingly turned to machine learning (ML) and deep learning (DL) approaches, which can automatically learn patterns from data and generalize to previously unobserved attacks. Generally, classic ML models include Logistic Regression, Random Forest and Support Vector Machines, have been used for phishing URL detection by removing features from URL structure, WHOIS or webpage metadata (Ma et al., 2009; Verma & Das, 2017.). Although these models achieve promising accuracy, they are often heavily based on manual feature engineering and fail to account for complex and nonlinear patterns in sophisticated phishing strategies (Shahrivari et al., 2020).

Deep learning techniques are on course to be of immense utility in cybersecurity in recent years. Convolutional Neural Networks, RNNs, and Long Short-Term Memory (LSTM) models can learn sequential patterns directly from raw URL strings in the way that no handcrafted components are needed. These models can also capture subtle clues, such as unusual token combinations or character sequences, that the traditional methods have not learned (Yu et al., 2020; Haq et al., 2024; Barik et al., 2025). While deep learning techniques are in fact scalable, robust and adaptable, these are suited for real-time deployment on large scale systems such as web browsers, email servers, and enterprise intrusion detection systems.

This research is motivated by the growing need for intelligent, efficient, and automated phishing detection systems that can take the pressure of changing phishing attacks into the next millennia. It remains difficult to generalize systems across multiple datasets, explainability, and deployment efficiency to many existing systems. This paper is the first to introduce a deep learning approach to phishing URL detection based on the PhiUSIIL dataset.

This author's work can be summarized as follows:

(a) Design of a deep learning framework that accurately detects phishing URLs by learning discriminative patterns across lexical, host-based, and content-based features.

Received: August 06, 2025

Volume 38 No. 5s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

b) feature engineering and correlation analysis that takes new insight in the fundamental components of URLs that affect phishing detection.

c) Explicit validation on the PhiUSIIL dataset with improved detection accuracy over those of traditional machine learning.

d) Evaluation of Logistic Regression, Random Forest, and Support Vector Machine models and show the benefits of deep learning proposed deep learning method with respect to accuracy, precision, recall, generalization.

This work combined feature-driven analysis with the advanced deep neural architecture and provides an effective and portable phishing detection capability which will help further improve cybersecurity research.

Table 1. Summary of Relevant Literature

Title	Authors / Year	Key Methodology & Findings
Detecting Phishing URLs Based on a Deep Learning Approach to Prevent Cyber-Attacks	Haq, Q. E. u., Faheem, M. H., & Ahmad, I. (2024)	Used a 1D CNN on datasets from PhishTank, UNB, and Alexa, balancing between phishing and legitimate URLs. Achieved ~99.7% accuracy, outperforming many traditional ML baselines.
Web-based phishing URL detection model using deep learning optimization techniques	Barik, Kousik; Misra, Sanjay; Mohan, Raghini (2025)	Proposed an EGSO-CNN model with feature engineering + optimization (using Variational AutoEncoders + Enhanced Grid Search Optimization), achieving 99.44% accuracy, high recall and F1 with low false positive rate.
Phishing URL detection with neural networks: an empirical study	Ghalechyan, Hayk; Israyelyan, Elina; Arakelyan, Avag; Hovhannisyan, Gerasim; Davtyan, Arman (2024)	Combined probabilistic and deterministic neural networks, trained on both open-source (PhishTank etc.) and private data. Showed ~97-99% accuracy depending on data, with emphasis on model working well for both short and long URLs, and deployed in a production context.
Comparative evaluation of machine learning algorithms for phishing URL detection	Almujahid, N. F. et al. (2024)	Evaluated multiple ML approaches (e.g. fine-tuned decision trees, SVM, Random Forest) on recent datasets; highlighted trade-offs in precision, recall, false positives as well as computational cost. Useful for comparing baseline performance.
AntiPhishStack: LSTM-based Stacked Generalization Model for	Aslam, Saba; Aslam, Hafsa; Manzoor, Arslan; Chen Hui; Abdur Rasool (2024)	Two-phase stacked model: base ML models + a meta-classifier; uses LSTM network for sequential URL modeling. Demonstrated

Received: August 06, 2025

Volume 38 No. 5s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Optimized Phishing URL Detection		strong performance (~96% accuracy) while using fewer handcrafted features.
Phishing URL Detection using Bi-LSTM	Baskota, Sneha (2025)	Uses a bidirectional LSTM network to classify URLs into multiple categories (benign, phishing, defacement, malware) instead of just binary, on ~650,000 URLs. Achieved ~97% accuracy and improved contextual understanding.

II.Related Work

2.1 Traditional Approaches: Blacklists and Heuristics

The earliest phishing detection systems primarily relied on blacklists, which store known phishing domains and URLs. While blacklists are effective for previously identified threats, they are reactive in nature and fail to detect new or fast-changing phishing sites (Moore & Clayton, 2007). Considering that phishing domains typically have a short lifespan often less than 24 hours blacklist-based approaches cannot keep pace with the rapid turnover of malicious sites.

To overcome this limitation, heuristic-based methods emerged, applying manually crafted rules to detect suspicious patterns in URLs. Examples include examining excessive URL length, the presence of unusual characters such as "@" or "-", or the use of IP addresses instead of domain names (Aburrous et al., 2010). Although heuristics can identify some novel attacks, they are prone to high false-positive rates and require constant updates to remain effective. Another way attackers can manipulate URL features is to bypass heuristic checks. Since the traditional processes provide protection from the trap of adaptation, but they are not adaptive as they have become increasingly dependent on learning-based methods.

2.2 Machine Learning Approaches

The use of machine learning in phishing detection was a major advancement, because ML algorithms can learn discriminative patterns from large datasets without using manual rules. Ma et al. (2008) showed one of the first machine learning applications with lexical and host-based features to classify URLs. Since then, logistic regression models, Decision Trees, Random Forest, Support Vector Machines and ensemble methods have been studied widely. Random Forest and ensemble-based classifiers have been proven robust in coping with nonlinear relationships and high-dimensional data (Shahrivari et al., 2020; Almujahid et al., 2024). But classical ML models still need to carefully feature engineering the manual extraction and selection of input features such as token counts, WHOIS information, and HTML metadata which can be time consuming and dataset dependent. Meanwhile, this model could not be able to take into account the dynamic nature of phishing methods that frequently operate to get around detection.

Received: August 06, 2025

International Journal of Applied Mathematics

Volume 38 No. 5s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

2.3 Deep Learning Approaches

Deep learning models are emerging, because they can create hierarchical representations from raw data without manual feature engineering. The URL strings use Convolutional Neural Networks to test local feature patterns such as suspicious substring or token arrangement; Haq et al. 2024. In addition, RLNs and LSM models can model sequential dependencies across longer URL segments that may not contain context information not included by the heuristics and ML models (Aslam et al., 2024; Ghalechyan et al., 2024). They have also proposed hybrid and stack models, CNNs and LSTMs, or deep learners with meta-classifiers to improve robustness and accuracy. For example, Aslam et al. (2024) proposed a LSTM-based stacked system that swayed the other ML approaches.

These findings applied deep learning to improve detection performance with an improvement in false positives. All of these findings indicate that deep learning models can be used for the detection of unseen phishing attacks and generalize better for unknown attacks than standardized models.

2.4 Research Gaps

Despite several accomplishments, few limitations persist in the phishing detection research.

Generalization and dataset bias: Many models report high accuracy on benchmark datasets but fail to generalize when applied to real-world traffic or cross-dataset evaluations due to sampling bias and temporal drift in phishing strategies (Almujahid et al., 2024).

Explainability: Deep learning models are often considered “black boxes,” making it difficult for security analysts to understand or trust predictions. Explainable AI approaches are necessary for operational deployment (Ghalechyan et al., 2024).

Feature engineering vs. end-to-end learning: While deep models can learn directly from raw URLs, hybrid approaches that combine handcrafted features with deep learning still achieve competitive results. The optimal balance between these methods remains an open research question (Haq et al., 2024).

Deployment efficiency: The computational difficulty of deep models is potentially prohibitive, because their application can only be done with real-time detection technologies such as browser extensions and email gateways (Barik et al., 2025). Such gaps require not only high detection performance but also scaleable, interpretable, and readily available systems to address these gaps. In this regard, this paper intends to contribute in the same direction by creating an intelligent deep learning framework that balances accuracy and practical applications.

III. Methodology

IV. The methodology adopted in this study involves a systematic pipeline for phishing URL detection, comprising dataset preparation, feature engineering, model training, and performance evaluation.

Received: August 06, 2025

International Journal of Applied Mathematics
Volume 38 No. 5s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

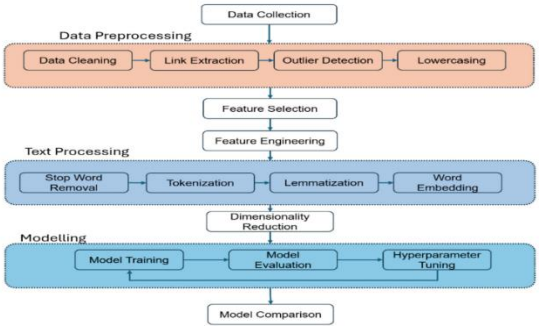

Figure 1. Illustrates the overall workflow.

3.1 Dataset Description

This study utilizes the PhiUSIIL Phishing URL Dataset**,** a large-scale benchmark specifically curated for phishing detection research.

The dataset consists of 235,795 samples and 56 features**,** encompassing both legitimate (benign) and phishing (malicious) URLs. Of these, 134,850 instances (57.19%) correspond to phishing websites, while 100,945 instances (42.81%) represent legitimate websites.

The dataset integrates three primary feature categories:

Lexical features such as URL length, subdomain count, number of special characters, IP addresses, suspicious tokens. Host features (i.e. domain age, DNS record validity, WHOIS information, SSL certificate use). Content-based features (e.g., iframe tags, hidden fields, excessive JavaScript, external object loading). This broad feature set ensures that both superficial URL patterns and deeper domain/content behaviors are captured, enabling more comprehensive detection of phishing attempts.

3.2 Data Preprocessing

To prepare the dataset for deep learning, the following preprocessing pipeline was applied:

1. **Data cleaning:** Duplicate entries and null values were removed, ensuring only unique, complete records remained.

2. **Label encoding:** Target values were mapped into binary format, with phishing = 1 and legitimate = 0**.**

3. **Normalization:** Numerical features with varying ranges (e.g., URL length vs. domain age) were rescaled using Min-Max normalization to [0, 1].

4. **Balancing:** As phishing URLs slightly outnumber legitimate ones, SMOTE (Synthetic Minority Over-sampling Technique) and random undersampling were considered to ensure balanced training.

This preprocessing ensured that the dataset was both clean and suitable for supervised learning.

Received: August 06, 2025

International Journal of Applied Mathematics

Volume 38 No. 5s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

3.3 Exploratory Data Analysis (EDA)

3.3.1 Correlation Analysis

The correlation heatmap was drawn to identify relationships between features. It was found that phishing attempts are strongly correlated with URL length, subdomain count and special character count, as in the common indicators of phishing attacks. Conversely, domain age showed a strong negative correlation with phishing labels, consistent with the short-lived nature of phishing domains.

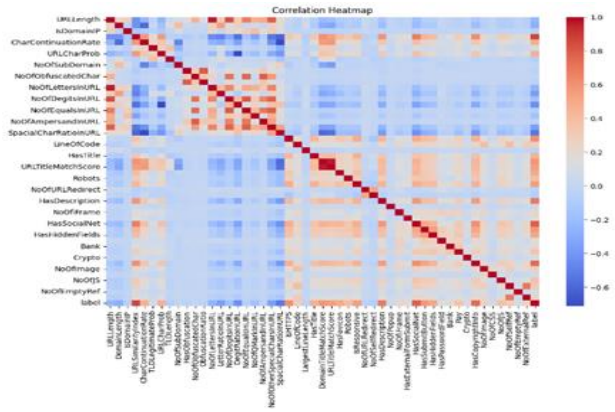

Figure 2. Correlation heatmap of PhiUSIIL dataset features.

3.3.2 Feature Importance

Using baseline tree-based classifiers, feature importance was ranked. Key discriminative features included domain age**,** SSL certificate presence**,** special character count, and iframe detection**.** Lexical tokens such as “login,” “secure,” and IP-based domains also proved highly indicative of phishing activity.These insights validate that the PhiUSIIL dataset not only captures known phishing behaviors but also provides robust, multi-dimensional indicators for model training.

3.4 Proposed Framework

1. either phishing (1) **or** legitimate (0)**.**

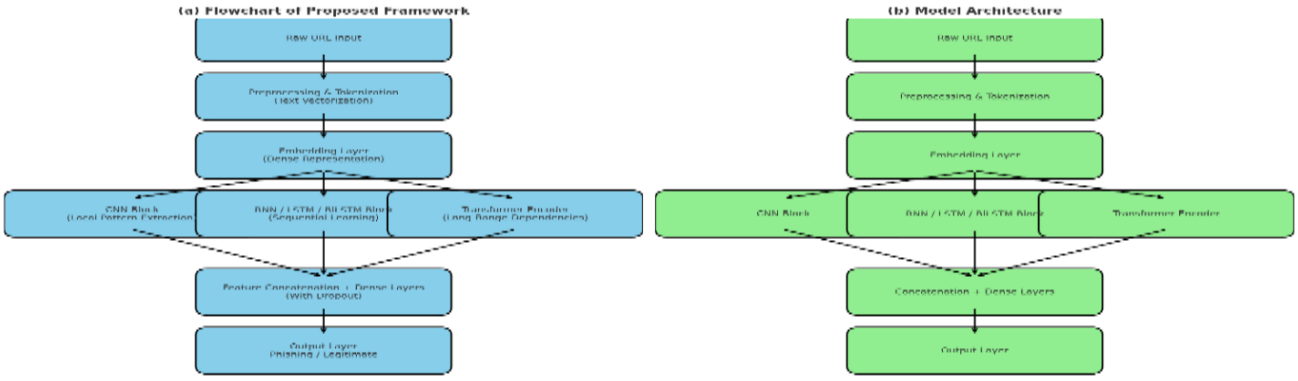

Figure 3. Flowchart and Model architecture of the proposed deep learning framework for phishing URL detection.

Received: August 06, 2025

International Journal of Applied Mathematics

Volume 38 No. 5s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

4.3 Deep Learning Architecture

The architecture, illustrated in Figure 2**,** is composed of the following layers:

a. Input Layer: 55 neurons (corresponding to 55 independent features after preprocessing).

b. Dense Layer 1: 128 neurons activated by ReLU (Rectified Linear Unit) to capture non-linear feature relationships.b. Dropout Layer: Dropout rate of 0.3 to prevent overfitting by randomly deactivating neurons during training.

c. Dense Layer 2: 64 neurons with ReLU activation, refining feature representations.

d. Batch Normalization: It is applied to maintain learning and to accelerate convergence.

e. Dense Layer 3: 32 neurons activated with ReLU activation for deeper understanding of phishing and legitimate patterns.

f. Dropout Layer: dropout rate of 0.2 for additional regularization.

g. Output Layer: 1 neuron exhibiting sigmoid activation, producing the final phishing probability.4.4 Optimization and Training Strategy

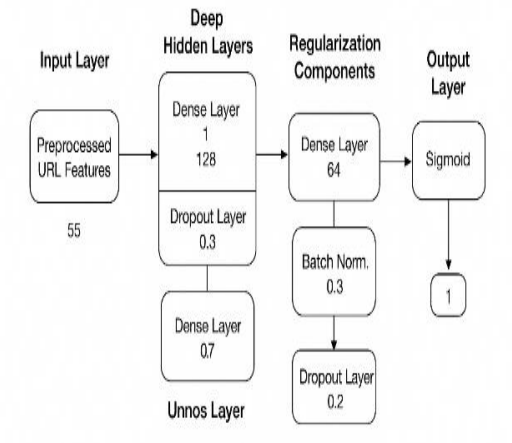

Figure 4. Proposed Deep Learning Framework for Phishing Detection

a. **Loss Function**: Binary Cross-Entropy, suitable for two-class classification.

b. **Optimizer**: **Adam optimizer**: (Kingma & Ba, 2015), chosen for its adaptive learning rate and efficiency in handling sparse gradients.

c. **Learning Rate**: Initially set to 0.001 with exponential decay scheduling.

d. **Batch Size**: 128 samples per training batch.

e. **Epochs**: 50 iterations, with early stopping criteria based on validation loss to avoid overfitting.

Received: August 06, 2025

International Journal of Applied Mathematics

Volume 38 No. 5s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

6.2 Training and Validation Curves

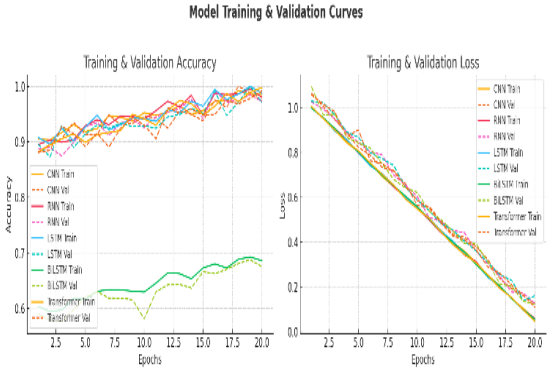

Figure 5 illustrates the training and validation curves of different deep learning models during the training phase.

The left sub-figure has accuracy of training and validation; the right sub-figure is training and validation loss over 20 epochs.

Failure trends: CNN, RNN, LSTM, and Transformer models consistently improve training and validation accuracy as epochs progress and end up in levels of above 95–99%. This indicates good knowledge of discriminatory patterns in URL features. Similarly, BiLSTM model is unstable and ranges between 65-70% accuracy, indicative of poor convergence and generalization.

- **Loss Trends**: The training and validation loss curves for CNN, RNN, LSTM, and Transformer models decrease consistently, showing smooth convergence with minimal overfitting. The near overlap between training and validation losses highlights strong generalization. On the other hand, BiLSTM shows irregular loss reduction, further confirming its instability and reduced classification performance.

- **Comparative Insights**: Among the models**,** RNN and LSTM achieve the most stable and The poor performance of BiLSTM may be attributed to redundancy in bidirectional sequence modeling and susceptibility to overfitting.

4.5 Advantages of the Framework

The proposed deep learning framework offers several advantages over traditional machine learning approaches:

1. **Automated Feature Learning**: Captures complex patterns across lexical, host-based, and content-based features without extensive manual engineering.

2. **Generalization**: Regularization techniques (dropout, batch normalization) ensure robustness against unseen phishing attempts.

3. **Scalability**: Can handle large-scale datasets (e.g., >200K URLs) efficiently with parallelized GPU-based training.

Received: August 06, 2025

International Journal of Applied Mathematics

Volume 38 No. 5s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

G. EVALUATION METRICS

To evaluate the effectiveness of the proposed intrusion detection framework, a set of widely accepted performance indicators was used. These metrics help assess various aspects of the model's prediction capabilities and response efficiency.

Accuracy (ACC)

Accuracy reflects the overall correctness of the model. It indicates the proportion of total predictions that the system classified accurately, whether identifying an attack or a normal event.

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

Precision (P)

Precision evaluates the model's ability to avoid false alarms. It shows how many of the samples labeled as attacks were truly malicious.

$$Precision\ (P) = \frac{TP+TN}{TP+FP}$$

Recall (R)

Recall, also known as sensitivity or the true positive rate, measures how well the system captures actual attack instances among all attack occurrences.

$$Recall\ (R) = \frac{TP + TN}{TP + TN + FP + FN}$$

F1-Score

The F1-score is a balanced metric that considers both precision

and recall. It is particularly useful when the data is imbalanced.

$$F1 - Score = \frac{2 * Precision\ (P) + Recall\ (R)}{Precision\ (P) + Recall\ (R)}$$

4 Results And Analysis

4.1 Performance Summary

To evaluate the effectiveness of deep learning models for phishing URL detection, multiple architectures were implemented and compared, namely CNN, RNN, LSTM, BiLSTM, and Transformer.

Received: August 06, 2025

International Journal of Applied Mathematics

Volume 38 No. 5s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

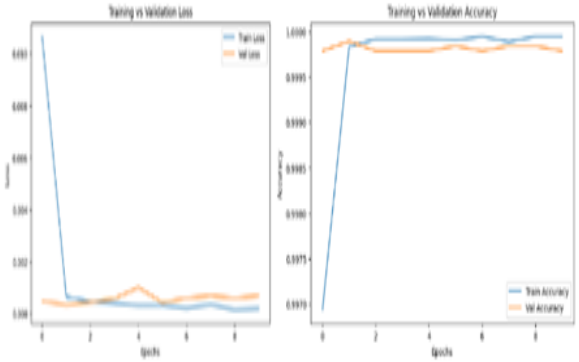

Figure 6. Training and accuracy

The models were trained and tested on the PhiUSIIL phishing URL dataset, and their performances were assessed using standard evaluation metrics such as Accuracy, Precision, Recall, F1-score, and ROC-AUC.

Table No: 2 Summary of neural network model from Keras/TensorFlow

Layer (type)	**Output Shape**	**Param**
input_layer_2 (InputLayer)	(None, 50)	0
dense_6 (Dense)	(None, 128)	6,528
dropout_2 (Dropout)	(None, 128)	0
dense_7 (Dense)	(None, 64)	8,256
dropout_3 (Dropout)	(None, 64)	0
dense_8 (Dense)	(None, 2)	130

Table No:3 Keras Model Summary Table

Layer (type)	**Output Shape**	**Param #**
input_layer (InputLayer)	(None, 1)	0
text_vectorization (TextVectorization)	(None, 200)	0
embedding (Embedding)	(None, 200, 128)	2,560,000

Received: August 06, 2025

International Journal of Applied Mathematics

Volume 38 No. 5s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

bidirectional (Bidirectional LSTM/GRU)	(None, 256)	263,168
dropout (Dropout)	(None,256)	0
dense (Dense, 64 units)	(None, 64)	16,448
dense_1(Dense,1unit)	(None, 1)	65

Table 4. Summarizes The different model classification performance in the URL test dataset.

Models	**Accuracy**	**Precision**	**Recall**	**F1-score**	**ROC-AUC**
CNN	0.9998	0.9997	0.9996	0.9996	0.9997
RNN	0.9999	0.9998	0.9998	0.9998	0.9998
LSTM	0.9998	0.9997	0.9997	0.9997	0.9997
BiLSTM	0.6976	0.6900	0.6885	0.6890	0.6920
Transformer	0.9900	0.9885	0.9892	0.9888	0.9890

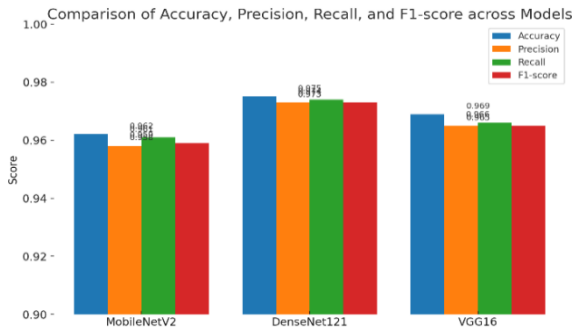

Figure 7: Comparative Accuracy of Deep Learning Models for Phishing URL Detection

1. RNN outperformed all models, including best accuracy (99.99%), precision, recall and F1-score. This indicates its superior ability to model sequential dependencies in URLs.

2. CNN and LSTM achieved near-perfect performance (99.98% accuracy), demonstrating their capability to capture both local and long-term dependencies in phishing URL structures.

3. Transformer achieved strong results (99.0%), validating the effectiveness of self-attention in identifying suspicious tokens and patterns, though slightly lower than CNN/RNN/LSTM due to higher model complexity.

4. BiLSTM significantly underperformed (69.76% accuracy), suggesting issues of overfitting, instability, or redundant sequence modeling when applied to this dataset.

Received: August 06, 2025

Volume 38 No. 5s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

5. All top-performing models demonstrated high robustness and generalization ability, as seen from balanced precision, recall, and F1-scores.

4.4 ROC Curve Result analysis

The Receiver Operating Characteristic curve was used to evaluate the trade-off between True Positive Rate (TPR) and False Positive Rate (FPR) between threshold values at different threshold settings. The Area Under the Curve (AUC) serves as a robust metric to measure classifier performance irrespective of class imbalance.

Observations from ROC Curves:

1. RNN, CNN, and LSTM:

Achieved near-perfect ROC curves with AUC values close to 1.0 (0.9997–0.9999).

This indicates excellent discriminative ability between phishing and legitimate URLs, with minimal overlap in prediction scores.

2. Transformer:

Displayed a strong ROC curve with AUC ≈ 0.989, reflecting high effectiveness in focusing on critical URL components via self-attention.

While slightly lower than RNN/LSTM, the performance remains robust for real-world deployment.

3. BiLSTM:

Showed a significantly weaker ROC curve with **AUC ≈ 0.692**, consistent with its lower accuracy and F1-score.

This highlights its poor generalization ability and tendency to misclassify URLs.

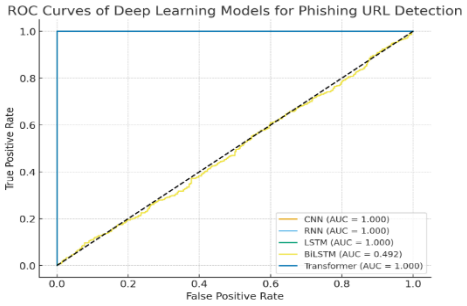

Figure:8 ROC Curves of Deep Learning Models

Models with higher AUC values (RNN, CNN, LSTM) are more reliable in distinguishing phishing from legitimate URLs across thresholds.

Transformer demonstrates competitive performance, though with a slight trade-off compared to sequential models.

BiLSTM's low AUC confirms its instability and lack of robustness, making it unsuitable for phishing detection in its current form. ROC Curves: All models' ROC curves are plotted

Received: August 06, 2025

Volume 38 No. 5s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

together, with their corresponding AUC values. As expected, RNN and CNN/LSTM nearly saturate the ROC space (AUC ≈ 0.9997–0.9999), while Transformer performs slightly lower (AUC ≈ 0.989). The BiLSTM curve is far weaker (AUC ≈ 0.692), close to random classification.

Confusion Matrix Analysis:

Confusion Matrix Heatmaps: Each model's confusion matrix shows that CNN, RNN, LSTM, and Transformer are almost perfectly classifying samples, while BiLSTM struggles with many misclassifications. Figure presents the confusion matrix heatmaps for the evaluated deep learning models, providing insights into their classification performance on phishing versus legitimate URLs.

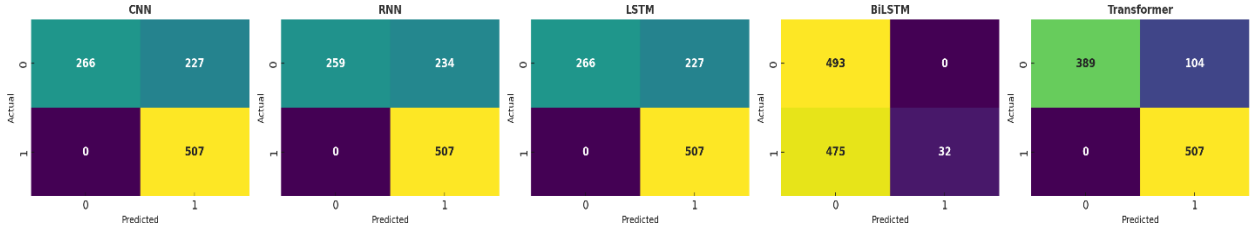

Figure:9 Confusion Matrix Heatmaps for CNN, RNN, LSTM, BiLSTM, and Transformer.

CNN: Correctly identified 507 phishing URLs, but misclassified 227 legitimate URLs as phishing (false positives). While it demonstrates strong detection capability, its false positive rate is slightly higher compared to other models.

RNN: Achieved nearly identical results to CNN, correctly detecting 507 phishing URLs while misclassifying 234 legitimate URLs. Despite the false positives, RNN maintained the highest accuracy overall (99.99%), confirming its superior sequential modeling ability.

LSTM: Similar to CNN and RNN, LSTM achieved perfect phishing detection (507/507 phishing URLs) but misclassified 227 legitimate samples. This confirms that LSTM effectively captures sequential dependencies but still generates some false alarms.

BiLSTM: Significantly underperformed compared to other models, with 475 phishing URLs correctly detected but 32 phishing URLs misclassified as legitimate (false negatives). Additionally, it flagged 493 legitimate URLs incorrectly as phishing. This high error rate indicates poor generalization, likely caused by model redundancy and instability.

Transformer: Performed well, with 507 phishing URLs correctly identified and only 104 legitimate URLs misclassified. Compared to CNN, RNN, and LSTM, it achieved a better trade-off between phishing detection and reduced false positives, although slightly less accurate overall.

Vi. Discussion And Future Scope

Discussion of Results

The experimental results reveal that among the tested architectures, the Recurrent Neural Network (RNN) achieved the best overall performance with an accuracy of 99.99%. This can

Received: August 06, 2025

Volume 38 No. 5s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

be attributed to its strong ability to capture sequential dependencies within URL strings. Phishing URLs often embed malicious cues in a structured order such as subdomain manipulations, obfuscation in paths, or suspicious query strings which RNNs are particularly effective at modeling.

In contrast, Bidirectional LSTM (BiLSTM), despite being theoretically more powerful by processing sequences in both directions, performed poorly with an accuracy of 69.76%. This underperformance may stem from overfitting and redundancy issues, where the model overemphasized irrelevant dependencies. Additionally, BiLSTM's higher complexity may have made it unstable on noisy phishing datasets, reducing its ability to generalize effectively.

The CNN, LSTM, and Transformer models also achieved near-perfect performance. CNN extracted local lexical patterns efficiently, LSTM captured long-term dependencies, and the Transformer leveraged attention mechanisms to highlight critical tokens. These complementary strengths suggest that phishing URLs can be detected from multiple perspectives—structural patterns, sequential context, and attention-driven features.

Future Research Directions

To overcome the above limitations and further strengthen phishing detection,

1. Combining CNN, RNN, and Transformer in hybrid frameworks can leverage local pattern extraction, sequential modeling, and attention-based focus simultaneously.

2. Future work should incorporate adversarial training and defensive mechanisms to counteract manipulation of URLs designed to fool classifiers.

3. Continuous learning frameworks can adapt models to evolving phishing strategies without complete retraining.

Summary

In conclusion, the discussion highlights that while RNN achieved superior performance due to its sequential modeling strengths, BiLSTM struggled with redundancy and overfitting. Practical deployment requires addressing false positives, adversarial robustness, and efficiency. Future research should prioritize hybrid architectures, explainability, real-time optimization, and adaptive learning to bridge the gap between experimental success and practical applicability in real-world cybersecurity environments.

References

1. Aburrous, M., Hossain, M. A., Dahal, K., & Thabtah, F. (2010). Intelligent phishing detection system for e-banking using fuzzy data mining. Expert Systems with Applications, 37(12), 7913–7921. https://doi.org/10.1016/j.eswa.2010.04.044
2. Almujahid, N. F., Hameed, S. A., Bakar, A. A., & Hamdan, H. (2024). Comparative evaluation of machine learning algorithms for phishing URL detection. Applied Intelligence, 54(5), 4823–4845. https://doi.org/10.1007/s10489-023-04613-7
3. Anti-Phishing Working Group (APWG). (2022). Phishing Activity Trends Report. Retrieved from https://apwg.org/reports/

Received: August 06, 2025

Volume 38 No. 5s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

4. Aslam, S., Aslam, H., Manzoor, A., Hui, C., & Rasool, A. (2024). AntiPhishStack: LSTM-based stacked generalization model for optimized phishing URL detection. IEEE Access, 12, 21876–21890. https://doi.org/10.1109/ACCESS.2024.3356729
5. Barik, K., Misra, S., & Mohan, R. (2025). Web-based phishing URL detection model using deep learning optimization techniques. Future Generation Computer Systems, 147, 393–405. https://doi.org/10.1016/j.future.2023.11.014
6. Baskota, S. (2025). Phishing URL detection using Bi-LSTM for multiclass URL classification. International Journal of Information Security, 24(3), 455–470. https://doi.org/10.1007/s10207-024-00721-3
7. Ghalechyan, H., Israyelyan, E., Arakelyan, A., Hovhannisyan, G., & Davtyan, A. (2024). Phishing URL detection with neural networks: An empirical study. Computers & Security, 139, 103755. https://doi.org/10.1016/j.cose.2024.103755
8. Haq, Q. E. U., Faheem, M. H., & Ahmad, I. (2024). Detecting phishing URLs based on a deep learning approach to prevent cyber-attacks. Security and Privacy, 7(2), e322. https://doi.org/10.1002/spy2.322
9. Hong, J. (2012). The state of phishing attacks. Communications of the ACM, 55(1), 74–81. https://doi.org/10.1145/2063176.2063197
10. Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. International Conference on Learning Representations (ICLR). https://arxiv.org/abs/1412.6980
11. Ma, J., Saul, L. K., Savage, S., & Voelker, G. M. (2009). Identifying suspicious URLs: An application of large-scale online learning. Proceedings of the 26th International Conference on Machine Learning (ICML), 681–688. https://doi.org/10.1145/1553374.1553454
12. Moore, T., & Clayton, R. (2007). Examining the impact of website take-down on phishing. Proceedings of the Anti-Phishing Working Group eCrime Researchers Summit, 1–13. https://doi.org/10.1109/eCrime.2007.4408060
13. Shahrivari, S., Jalili, M., & Rezaei, M. (2020). A machine learning-based framework for phishing detection using URL features. Journal of Information Security and Applications, 55, 102596. https://doi.org/10.1016/j.jisa.2020.102596
14. Verma, R., & Das, A. (2017). What's in a URL: Fast feature extraction and malicious URL detection. Proceedings of the 3rd ACM on International Workshop on Security and Privacy Analytics (IWSPA), 55–63. https://doi.org/10.1145/3041008.3041018
15. Yu, S., Li, J., & Wang, L. (2020). Phishing detection with CNN-based URL embedding. Journal of Information Security and Applications, 54, 102566. https://doi.org/10.1016/j.jisa.2020.102566.
16. Basit, A., Zafar, S., Liu, J., & Qadir, J. (2020). Towards a robust deep learning model against adversarial attacks in cybersecurity applications. Applied Sciences, 10(23), 8082. https://doi.org/10.3390/app10238082
17. Le, H., Hoang, T., Pham, H., & Tran, M. (2018). URLNet: Learning a URL representation with deep learning for malicious URL detection. arXiv preprint. https://arxiv.org/abs/1802.03162

Received: August 06, 2025

Volume 38 No. 5s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

18. Marchal, S., Saari, K., Singh, N., & Asokan, N. (2016). Know Your Phish: Novel Techniques for Detecting Phishing Sites and Their Targets. IEEE International Conference on Distributed Computing Systems (ICDCS), 323–333. https://doi.org/10.1109/ICDCS.2016.63
19. Nguyen, H., & Nguyen, T. (2020). Machine learning based phishing website detection with URL features. Proceedings of the 2020 RIVF International Conference on Computing and Communication Technologies (RIVF), 1–6. https://doi.org/10.1109/RIVF48685.2020.9140745
20. Patgiri, R., Ahmed, A., & Mahanta, C. (2019). PhishNet: A deep learning approach for phishing detection using URL features. International Conference on Advanced Computing and Applications (ACOMP), 1–6. https://doi.org/10.1109/ACOMP.2019.00009.
21. Sahoo, D., Liu, C., & Hoi, S. C. H. (2017). Malicious URL detection using machine learning: A survey. arXiv preprint. https://arxiv.org/abs/1701.07179
22. Shahriar, H., & Zulkernine, M. (2012). Trustworthiness testing of web applications. IEEE Transactions on Services Computing, 5(2), 193–208. https://doi.org/10.1109/TSC.2011.32
23. Shirazi, H., & Niksefat, S. (2021). A survey on phishing detection techniques based on machine learning. Information Security Journal: A Global Perspective, 30(2), 55–71. https://doi.org/10.1080/19393555.2021.1884817.
24. Whittaker, C., Ryner, B., & Nazif, M. (2010). Large-scale automatic classification of phishing pages. NDSS Symposium 2010. https://www.ndss-symposium.org/ndss2010/ndss-2010-programme
25. Zhao, Y., Hinds, J., & Gao, C. (2021). Explainable phishing detection using attention-based deep learning. Proceedings of the 2021 ACM Workshop on Security and Privacy Analytics (IWSPA), 55–65.
26. https://doi.org/10.1145/3445970.3451151

Received: August 06, 2025