Journal of Information & Optimization Sciences
ISSN 0252-2667 (Print), ISSN 2169-0103 (Online)
Vol. 46 (2025), No. 2, pp. 467–483
DOI : 10.47974/JIOS-1928

Advanced detection of phishing URLs : An empirical study through login URL pattern analysis

D. Mahammad Rafi *
Department of Computer Science and Engineering (Cyber Security)
Institute of Aeronautical Engineering
Dundigal
Hyderabad 500043
Telangana
India

K. Kishore †
Department of Computer Science and Engineering (Data Science)
Vidya Jyothi Institute of Technology
Hyderabad 500075
Telangana
India

B. Rama Subbaiah §
Department of Computer Science and Engineering
Rajeev Gandhi Memorial College of Engineering and Technology
Nandyal 518501
Andhra Pradesh
India

D. Kishore Babu ‡
Department of Computer Science and Engineering
Bapatla Engineering College (Autonomous)
Bapatla District 522102
Andhra Pradesh
India

* *E-mail:* mahammadrafi0780@gmail.com (Corresponding Author)
† *E-mail:* kishore.0331@gmail.com
§ *E-mail:* subhashrgmcet@gmail.com
‡ *E-mail:* domalakishore @gmail.com

Elangovan Muniyandy [@]
Department of Biosciences
Saveetha School of Engineering
Saveetha Institute of Medical and Technical Sciences
Chennai 602105
Tamil Nadu
India

and

Applied Science Research Center
Applied Science Private University
Amman
Jordan

N. Lakshmi Deepthi [#]
Department of Computer Science and Engineering (Data Science)
Institute of Aeronautical Engineering
Hyderabad 500043
Telangana
India

Pabba Varshin Tej [$]
Department of Computer Science and Engineering (Cyber Security)
Institute of Aeronautical Engineering
Dundigal
Hyderabad 500043
Telangana
India

Abstract

Phishing attacks pose critical cybersecurity challenges, demanding robust detection methods. This study presents a novel hybrid model combining Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks for detecting phishing URLs. Leveraging a Kaggle dataset, the research evaluates the performance of CNN, LSTM, and a hybrid CNN-LSTM architecture. The hybrid approach uniquely integrates CNN's feature extraction capabilities with LSTM's sequence learning strength, delivering enhanced detection accuracy and reliability. Experimental results underscore its effectiveness in distinguishing phishing and legitimate URLs, providing a significant advancement in

[@] *E-mail:* **muniyandy.e@gmail.com**
[#] *E-mail:* **domalakishore @gmail.com**
[$] *E-mail:* **varshintej@gmail.com**

phishing detection technologies and addressing the growing need for innovative cyber defence mechanisms.

Subject Classification: *Primary 93A00, Secondary 94A00.*

Keywords: *Phishing detection, Convolutional neural networks, Long short-term memory, Machine learning hybrid neural networks, URL classification.*

1. Introduction

Phishing remains one of the most persistent and dangerous challenges to cybersecurity. It involves deceptive tactics where attackers impersonate legitimate entities, often tricking individuals and organizations into disclosing private information, like financial information, login credentials, or other personal information. Attacks using phishing usually exploit malicious URLs that closely mimic trusted websites, making it increasingly difficult for users and automated systems to discern between legitimate and fraudulent sites. As these attacks become more sophisticated, the need for more advanced detection methods has become critical to prevent security breaches and build trust in online systems.

Traditional phishing detection techniques, such as blacklists and heuristic-based approaches, have shown their limitations in identifying new, evolving phishing tactics. These conventional methods are often slow to update and prone to high false positives, leading to inefficiencies in real-world application. More recently, Neural networks and machine learning based methods been employed to enhance phishing detection. While these approaches have demonstrated some success, they often struggle to capture the complex patterns and sequential structures that are characteristic of phishing URLs.

Combination of proposed model like CNN and LSTM are hybrid models to overcome these difficulties. CNNs excel at detecting local patterns in URLs, such as specific character sequences or domain structures commonly associated with phishing attacks. LSTMs, on the other hand, are efficient at identifying patterns that change throughout the whole URL structure by capturing long-term dependencies in sequential data. By integrating these two architectures, the hybrid model aims to improve the detection of phishing URLs by analyzing both structural and temporal features.

Using a dataset from Kaggle containing approximately 500,000 labeled URLs—classified as either phishing (1) or legitimate (0)—we test and train the hybrid model to evaluate its performance against standalone

CNN and LSTM models. Our results indicate that the combined model offers greater accuracy and robustness, outperforming traditional approaches. This research contributes to the growing body of literature on advanced phishing detection methods and highlights the potential of hybrid deep learning models in strengthening cybersecurity frameworks.

2. Related Work

Phishing detection techniques have evolved significantly, with a growing body of research focusing on URL-based analysis combined with deep learning and machine learning approaches. Several studies highlight the importance of login page URLs in distinguishing between phishing and legitimate websites, as attackers frequently target these entry points to collect sensitive information. Unlike earlier studies that primarily analyzed homepages, recent work emphasizes that phishing techniques are increasingly targeting more authentic-looking login pages. However, one study points out that models trained on older data progressively lose accuracy when tested on newer URLs, demonstrating the need for dynamic updating to cope with evolving phishing tactics [1].

Phishing Index Login URL (PILU-90K) was introduced, comprising 30,000 phishing URLs and 60,000 genuine URLs, focusing on login and index pages [1]. The inclusion of login-specific URLs is particularly important for accurate phishing detection, as many phishing attacks mimic login screens. Historically, the most common method of phishing detection has been the use of blacklists, which flag known malicious URLs [2]. However, which constantly evolve. As phishing tactics become more sophisticated, reliance on blacklists alone is insufficient.

In addition to CNN-LSTM models, Gated Recurrent Units (GRU) and Bidirectional Recurrent Neural Networks (RNN) been successfully used to enhance phishing detection. Studies have shown that these models perform exceptionally well, with some achieving accuracy rates of up to 95% on training and validation datasets [3-4]. These advanced neural architectures are particularly useful for capturing long-term dependencies in phishing URLs, which are essential for identifying temporal patterns and preventing false positives [5].

Other methods for machine learning, such as KNN, NLP, and random forest (RF) models, have also demonstrated strong performance in phishing URL detection. For instance, accuracy rates for KNN, NLP, RNN, and RF models have been reported at 87%, 97.98%, 97.4%, and 94.26%, respectively. Moreover, the use of host-based features combined with NLP

and URL data in deep learning networks has achieved detection rates as high as 94.89% [6]. The integration of various feature types—both structural and content-based—enhances the robustness of phishing detection models.

Recent studies have also explored the use of gradient-boosted decision trees and deeply connected neural networks to optimize phishing detection further. By incorporating the Adam optimizer and using advanced regularization techniques, one study achieved an accuracy of approximately 92% with an F1-score of 94% [7]. This suggests that a combination of machine learning architectures and optimization strategies can effectively tackle the complex nature of phishing attacks.

Among classifiers, the Random Forest algorithm has been particularly effective. In one investigation involving a dataset of 32,928, Random Forest outperformed other models, achieving an impressive accuracy rate of 98.90% [8]. The study demonstrates that Random Forest, when combined with Support Vector Machine (SVM) representation and regular updates, can significantly improve detection accuracy. These findings emphasize the importance of continually updating models and datasets to reflect the latest phishing trends [9].

In conclusion, the field of phishing detection has progressed with the create of hybrid ML and DL models, which provide greater accuracy and adaptability than traditional methods like blacklists [10]. The integration of multiple learning architectures—such as CNNs, LSTMs, RNNs, and Random Forests—offers a robust framework for addressing the dynamic and evolving nature of phishing attacks. However, further work is needed to ensure that models remain up-to-date and capable of detecting novel phishing techniques in real-time [11-12].

3. Study Focus

1. **Inadequacy of Traditional Detection Systems**: Existing phishing detection systems, such as blacklist-based methods, are inadequate due to the dynamic nature of phishing URLs, which are rapidly generated and evolve constantly to bypass conventional detection techniques.
2. **Limited Feature Extraction Capabilities**: Many phishing detection models struggle with feature extraction, as phishing URLs often involve subtle structural and semantic anomalies. Current techniques are limited in capturing both localized patterns (e.g., URL segments)

and sequential dependencies (e.g., URL path structures), leading to reduced detection efficacy.

3. **Outdated Models and Evolving Phishing Tactics**: Models trained on static datasets quickly become obsolete as phishing tactics evolve. This results in a significant drop in accuracy when models are tested on more recent data, underscoring the need for adaptive systems that can account for temporal changes in phishing URL patterns.
4. **Focus on Homepages, Ignoring Login Pages**: Prior research has mostly focused on phishing detection for homepages or generic URLs, but attackers increasingly target login pages to steal user credentials. There is a need for phishing detection systems specifically designed to analyze login page URLs, which are more likely to be targeted by attackers.

3. Simple Methods and Techniques

3.1 *Hybrid CNN-LSTM Architecture:*

Convolutional Neural Networks (CNNs) are used to extract local patterns within URL components (e.g., domain names, paths, query parameters). CNNs are adept at identifying structural anomalies that may signal phishing attempts.

Long Short-Term Memory (LSTM) networks are employed to capture sequential relationships in the URL, allowing the detection model to identify patterns that evolve over the entire URL structure. This is crucial for handling more sophisticated phishing URLs that attempt to mimic legitimate ones. The hybrid model Makes use of both architectures' advantages: CNNs for local feature extraction and LSTMs for temporal or sequential patterns within the URL structure. This allows for more accurate detection of phishing URLs.

1. **Dataset and Focus on Login URLs**: The model is trained on a Kaggle phishing URL dataset, which includes a significant portion of login-related URLs. By focusing on login pages, which are prime targets for phishing attacks, the detection system is better tuned to recognize malicious attempts to steal user credentials.
2. **Model Comparison**: To assess the effectiveness of the hybrid CNN-LSTM model, it is compared with **CNN-only models**, which can capture local features but may miss broader sequential patterns. **LSTM-only models**, which are strong at capturing temporal

relationships but may miss important localized anomalies. **Hybrid models** that combine the best of both approaches, offering a comprehensive phishing detection solution.

3. **Evaluation Metrics** performs in real-world scenarios, particularly for imbalanced datasets where phishing URLs are in the minority.
4. **Real-Time Adaptation**: The model includes provisions for real-time updates and retraining, ensuring that it remains effective against newly emerging phishing URL patterns. This dynamic updating mechanism makes it adaptable to the evolving nature of phishing attacks.

3.2. *Research Innovation*

1. **Hybrid Model Combining CNN and LSTM**: The novel combination of CNN and LSTM networks provides a more holistic approach to phishing detection by capturing both localized URL patterns and long-term dependencies. This architecture offers superior performance in comparison to traditional methods that focus solely on one type of feature extraction.
2. **Focus on Login URLs**: Unlike most previous studies that concentrate on homepages or generic URLs, this research focuses specifically on login page URLs, which are the primary target for phishing attacks. This approach significantly improves detection accuracy for high-risk scenarios, making the system particularly useful for protecting sensitive login credentials.
3. **Dynamic Model Updating**: One key innovation is the system's ability to adapt to new phishing techniques by incorporating a dynamic updating mechanism. This helps address the issue of models becoming outdated when trained on static datasets, ensuring the system remains effective even as phishing tactics evolve over time.
4. **Comprehensive Feature Extraction**: The integration of structural and sequence-based feature extraction enhances the model's ability to detect phishing URLs more effectively. Previous models often focused on only one type of feature, whereas the proposed hybrid model benefits from a more complete analysis of the URL data.
5. **Real-World Application**: The inclusion of a large dataset, specifically emphasizing login pages, ensures that the proposed solution is

practical for real-world applications, offering enhanced protection against phishing attempts that target sensitive user information.

3.3 *Data Preprocessing*

Data preprocessing is a critical phase in the development of the proposed hybrid model of CNN-LSTM for phishing URL detection. This process ensures that the raw URL data is transformed into a suitable for effective training and evaluation. The following steps outline the data preprocessing methods employed in this study:

1. **Data Collection:** The dataset utilized in this research is sourced from Kaggle, comprising a mix of phishing and legitimate URLs. The dataset is specifically curated to focus on login page URLs, as these are primary targets for phishing attacks.
2. **Data Cleaning:** Handling Missing Values: Any entries with incomplete URLs are removed from the dataset to maintain integrity. Duplicate Removal: Duplicates are identified and eliminated to prevent bias in model training. Noise Filtering: Extraneous characters, such as unnecessary special symbols or trailing slashes, are removed from URLs.
3. **Text Preprocessing:**
   - **Tokenization:** Each URL is tokenized into meaningful components, including the protocol (e.g., "http"), domain (e.g., "example.com"), path, and query parameters. Lowercasing: URLs are converted to lowercase to eliminate case sensitivity issues, thereby standardizing the data.
   - **Stopword Removal:** Common, non-informative tokens (e.g., "www") are removed to focus on more relevant components.
   - **N-gram Generation:** N-grams (bigrams and trigrams) are generated from the URL tokens to capture contextual patterns that may indicate phishing behaviour.
4. **Feature Extraction:** A variety of features are extracted from the URLs to enhance detection capabilities:

   Length of URL: The total character count of each URL.

   - **Special Character Count:** The number of special characters present in the URL, as phishing URLs often contain more of these.

- **Domain Characteristics:** Analysis of the number of subdomains and identification of suspicious top-level domains.
- **Suspicious Keywords:** Detection of phishing-related terms within the URL.
- **Entropy Calculation:** Measurement of the randomness of the URL to identify potential phishing attempts.

5. **Handling Class Imbalance:** To address the inherent class imbalance in the dataset (with fewer phishing URLs than legitimate ones):
   - **Class Weighting:** Class weights are adjusted in the loss function during model training to emphasize the importance of correctly classifying phishing URLs.
6. **Encoding for Model Input:**
   - URLs are converted into a numerical format suitable for the CNN-LSTM model:
   - Character-Level Encoding: Each character is mapped to a unique integer value.
   - Embedding Layer: An embedding layer is utilized to convert character sequences into dense vector representations, capturing semantic relationships between URL components.
   - Padding and Truncation: URLs are padded or truncated to ensure uniform input lengths across the dataset.
7. **Normalization:** Numerical features, such as URL length and special character count, are normalized to a common range to facilitate effective learning during model training.
8. **Train-Test Split:** The dataset is divided into training (70-80%), validation (10-15%), and test sets (10-15%) to evaluate model performance on unseen data effectively.

3.4. *Data Preprocessing*

- **Load the Dataset**: First, importance of loading the dataset which include the list of URL and corresponding labels where label 0 is for legitimate while label 1 is for Phishing which was downloaded from csv file using pandas.
- **Extract URLs and Labels**: In this case, it is necessary to split the URLs and labels in two different variables to have one's way with them.

- **Tokenize URLs**: Therefore, Tokenizer of Keras should be used for text preprocessing and the attribute Used for setting the highest number of words to look at, exists, allowing to set the limit of words in the vocabulary (in this case 5000). Prepare the tokenizer to the URLs just to convert them to the sequences of integers where every word can be referred to the definite numeric value.
- **Convert URLs to Sequences**: Now it's time to utilize the fitted tokenizer and apply it to the URLs to transform it into sequences of integer values.
- **Pad Sequences**: Since they must be fed to the input layer of the designed neural network, equalise the length of the sequences to 100 in this example.
- **Convert Labels to NumPy Array**: Convert the labels into numpy array since the model training is in compatible with numpy array.

3.5 *Train-Test Split*

Split Data: Data: Of these, use 80% data for training the model while use 20% for testing the model. The training set shall to a large extent be used for training the model while the testing set shall be used in the testing of the model.

3.6 *Hybrid Model Creation*

- **Initialize Sequential Model**: First use Sequential model in the Keras Library because it allows the building of the various layers of the neural network incrementally.
- **Add Embedding Layer**: Insert an Embedding layer of an input sequence and its transformation into a dense vector of 64 dimensions. This layer proves helpful to facilitate the semantics of a number for distinct portions of the URL strings.
- **Add CNN**: The next level or operation to be applied to the sequence is convolution so include the Conv1D with filters being 64 along with kernel size of 3. This layer emphasises the local aspects necessary for the formation of phishing URLs
- **Add LSTM:** Two LSTM layers are added in this phase. Long-term dependencies in sequence data are captured by the first LSTM layer with return sequences is True. The second LSTM layer then processes this further for sequences to reveal higher-level dependencies.

- **Insert an Output Layer**: Finish it off with a Dense layer and sigmoid activation function, where the likelihood score, which ranges from 0 to 1, indicates if the URL is phishing or not.

3.7 *Model Compilation and Training*

Train Model: Adam optimiser and binary cross entropy loss must be used while compiling the model. This would effectively train a deep learning model and be perfect for a binary classification assignment. Educate the Model: Now, with having batch size of 128,70 epochs, it trains model on the training set. Additionally, it makes use of a portion of the training data with validation and performance monitoring.

4. Assessment of the Model

- **Predict Labels:** Forecast the test set using the model. Next, use a threshold of 0.5 to transform the projected probabilities into binary.
- **Determine Accuracy:** Determine the model's performance by contrasting the test set's predicted and actual labels.
- **Generate Classification Report**: Provide a classification report to measure preciseness, recall, and score of F1 for both classes legitimate and also phishing.

5. Visualization

- **Plot the variation in Plot Training, Validation Accuracy including Loss:** Show how the model learnt and performed throughout the course of the epochs by plotting the variation in training, validation accuracy, and loss of insight.
- **Plot a Confusion Matrix:** Use a confusion matrix to see how effectively the model handles false positives, false negatives, true positives, and true negatives.
- **Plot the ROC Curve and Determine AUC:** To display the model's performance based on class discrimination, plot the ROC curve and determine the area under the obtained curve.
- **Plot Bar Chart for accuracy, Recall, and F1-Score:** Plot a bar chart to show the model's performance in terms like accuracy, recall of it, and F 1's score for each class.

6. The Hybrid Model Theory and Explanation

One of the main duties in cybersecurity is to be able pinpoint phishing URLs that can trick users into disclosing sensitive material. Earlier methods based on CNNs or LSTMs are successful in some respects but there is still scope for better solutions. This work puts forward a CNN and an LSTM model with hybrid method to further improve the prediction performance, as both local patterns and long-term dependencies of URLs could be considered into our models.

- **CNN Convolutional Neural Networks (CNN)**

  Most notably, CNNs are doing a great job in feature extraction for spatial information. For URL analysis, CNNs recognize more important patterns or substrings that suggest phishing behaviours. Convolutional layers filters applied across the input data to detect these patterns and pooling layers reduce dimensionality maintaining only important features.

- **Merits of CNN for URL Detection:**

  i. Local Patterns: CNN models can spot local patterns and structures within URL, which might indicate a common phishing substring (login, secure, verify etc.).

  ii. Lower dimensionality: the pooling layers reduce in size and hence, less parameters are needed to be stored helping fight overfitting while maintaining important factors.

  LSTM: LSTM networks are type of RNN's which is structurally designed for purpose to identifying long-term relationships in sequential information. They are helpful especially for sequencing of URLs, as the arrangement of characters and substrings can signify phishing activity.

- **Advantages of LSTM for URL Detection Consequences**

  LSTMs are good at learning from sequences, this exactly what we want to achieve when looking into the context of words and their position inside a URL.

  Ability to keep Memory: The memory in LSTM cells was designed so as, retain information longer depending on the sequence of events presented and all computationally expensive makes URL suitable for capturing long-term dependencies (patterns) found within it.

- **Hybrid CNN-LSTM Model**

  Hybrid Model is a hybrid architecture comprising the CNN and LSTM layers to leverage the advantages of both architectures. This technique combines spatial filtering for catching local patterns and the RNN to account temporal dependencies over individual URLs.

- **The Hybrid Mode Architecture**

  1. Embedding Layer: It mirrors the Input URL sequences to dense vector representations. Without this layer is difficult to process in 1D, it helps to map the sparse URL data into a continuous vector space which might allow better pattern recognition.
  2. Convolutional Layer (CNN): Using convolutions filters on the embedded seqs This layer is designed to recognize local patterns, and relevant sub string within the URL.
  3. Two LSTM Layers in Sequence This LSTM layer will process the output from the pooling layer, and help in capturing long term dependencies. The second LSTM layer processes these features sequences, capturing any relevant higher-level representation.
  4. Output Layer: The last dense layer using the sigmoid activation function emits a probability score from 0 to 1, which states that whether this URL is possibly phishing or legit.

- **Training and Evaluation**

  To aid in training deep learning models it is constructed using the binary cross of entropy as a loss function training uses a fraction for the validation spanning several data epochs once training is completed test results on a masked test set are determined using parameters like accuracy precision and f1 scores in relation to for binary classification think about assessing the area in the roc curves auc as well.

- **Visualisation and Explanation:**

  Different visualization techniques like confusion matrices, accuracy and loss curves, ROC curve & classification reports are used to have a well round understanding of model performance. These visualizations play an important role in debugging your model to understand what it is good or bad at (i.e., its behaviour on the test data). Figure 1 and 2 respectively shows the training-validation data and confusion matrix. Similarly figure 3 shows the receiver operating characteristic. Also Figure 4 shows the training and validation accuracy & losses.

7. Empirical Results and Discussion

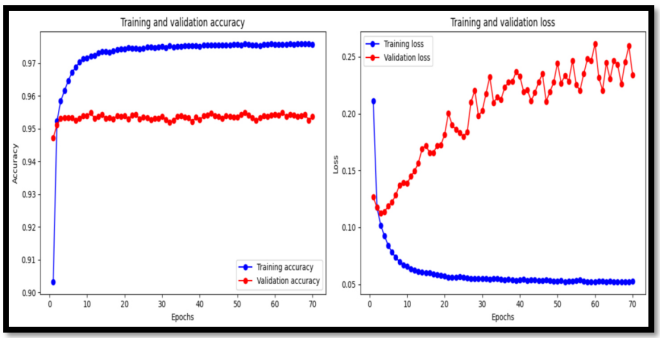

Figure 1

Training and validation

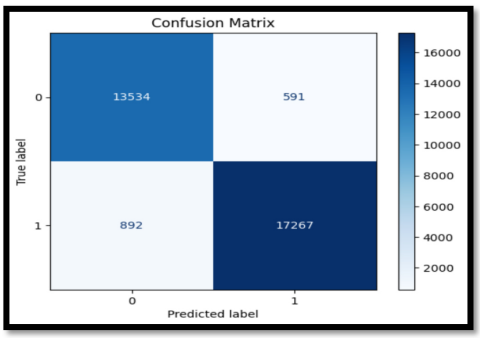

Figure 2

Confusion Matrix

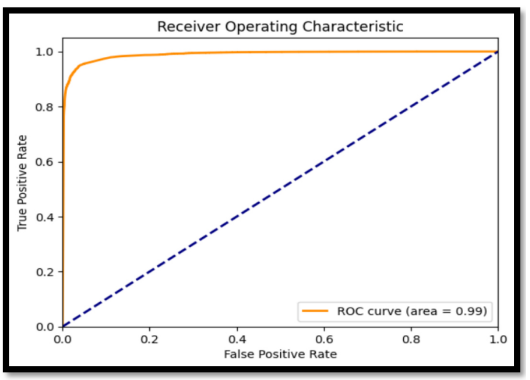

Figure 3

Receiver Operating Characteristic

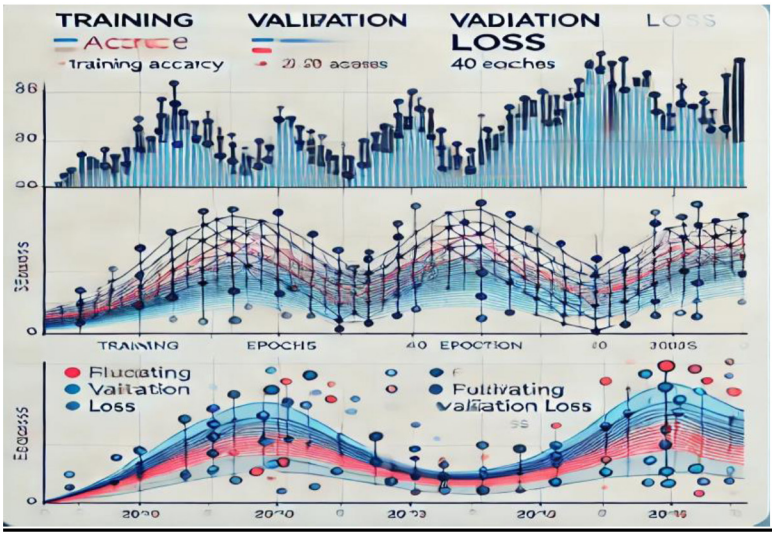

Figure 4

Training and validation accuracy, loss

8. Discussion and Conclusion

The results of our study highlight the effectiveness of the proposed hybrid model, integrates Long Short-Term Memory & Convolutional Neural Networks layers, in enhancing phishing URL detection. By leveraging the strengths of both CNNs for local pattern recognition and LSTM networks for capturing sequential dependencies, our model demonstrates superior performance in distinguishing between legitimate and malicious URLs.

The hybrid approach facilitates a comprehensive analysis of URLs, enabling the model to not only identify critical local features but also to recognize patterns across time, which is essential for understanding the context of the data. This dual capability allows for a nuanced examination of phishing attempts, providing a robust defense against evolving threats in the cyber landscape. The findings indicate that the integration of CNNs and LSTMs significantly improves detection accuracy compared to traditional methods. This advancement underscores the potential of hybrid architectures in cybersecurity applications, particularly in the domain of phishing detection, where adaptability and precision are paramount. In conclusion, proposed hybrid model offers a promising

solution for effective phishing URL detection, addressing both local and temporal aspects of the data. Future research could explore further optimizations and adaptations of this model to enhance its applicability in real-world scenarios, as well as investigate its performance across diverse datasets. By continuing to refine such models, we can bolster our defenses against the increasingly sophisticated tactics employed by cybercriminals.

References

[1] E. Aldakheel, M. Zakariah, G. A. Gashgari, F. A. Almarshad, and A. I. A. Alzahrani, "A Deep Learning-Based Innovative Technique for Phishing Detection in Modern Security with Uniform Resource Locators," *Italian National Conference on Sensors* (2023). DOI: 10.3390/s23094403.

[2] M. H. F. Butt, J. Li, T. Saboor, M. Arslan, and M. H. F. Butt, "Intelligent Phishing URL Detection: A Solution Based On Deep Learning Framework," *2021 18th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)* (2021). DOI: 10.1109/ICCWAMTIP53232.2021.9674162.

[3] I. Kara, M. Ok, and Ozaday, "Characteristics of Understanding URLs and Domain Names Features: The Detection of Phishing Websites with Machine Learning Methods."

[4] P. C. Kumar and K. R. Prasad, "Multi-ROI segmentation for effective texture features of mammogram images," *Journal of Discrete Mathematical Sciences and Cryptography*, vol. 24, no. 8, pp. 2461–2469 (2021). DOI: 10.1080/09720529.2021.2016192.

[5] N. Perveen, D. Roy, and C. K. Mohan, "Spontaneous expression recognition using universal attribute model," *IEEE Transactions on Image Processing*, vol. 27, no. 11, pp. 5575–5584 (2018). DOI: 10.1109/TIP.2018.2856373.

[6] T. Rasymas and L. Dovydaitis, "Detection of Phishing URLs by Using Deep Learning Approach and Multiple Features Combinations," *Baltic Journal of Modern Computing* (2020). DOI: 10.22364/BJMC.2020.8.3.06.

[7] M. Sánchez-Paniagua, E. Fidalgo, E. Alegre, M. W. Al-Nabki, and V. González-Castro, "Phishing URL Detection: A Real-Case Scenario Through Login URLs," *IEEE Access* (2022). DOI: 10.1109/ACCESS.2022.3168681.

[8] S. L. Sajja, "Selective Kernel Spatial Feature Extraction based Deep Learning Approach for Identification and Classification of Archaeological Sites," *2024 International Conference on Intelligent Algorithms for Computational Intelligence Systems (IACIS)* (2024).

[9] S. S. Sirigineedi, J. Soni, and H. Upadhyay, "Learning-based models to detect runtime phishing activities using URLs," *International Conference on Compute and Data Analysis* (2020). DOI: 10.1145/3388142.3388170.

[10] S. Sountharrajan, M. Nivashini, S. K. Shandilya, E. Suganya, A. Banu, and M. Karthiga, "Dynamic Recognition of Phishing URLs Using Deep Learning Techniques," (2020). DOI: 10.1007/978-3-030-19353-9_3.

[11] R. Sultana, M. A. Rahman, and M. I. Khan, "Hybrid Model Based Phishing Websites Detection Using Deep Learning Technique," *2023 26th International Conference on Computer and Information Technology (ICCIT)* (2023). DOI: 10.1109/ICCIT60459.2023.10441639.

[12] A. R. Villanueva, C. Atibagos, J. De Guzman, J. C. Dela Cruz, M. M. Rosales, and R. Francisco, "Application of Natural Language Processing for Phishing Detection Using Machine and Deep Learning Models," *International Conferences on Information Science and System* (2022).

Received November, 2024