International Journal for Electronic Crime Investigation

ISSN: 2522-3429 (Print)
ISSN: 2616-6003 (Online)

DOI: https://doi.org/10.54692/ijeci.2026.1001/266

Research Article

Vol. 10 issue 1 Jan-Jul 2025

Deep Learning-Driven Malicious URL Detection: A comprehensive analysis using Convolutional Neural Networks A feature engineering on the Phiusiil dataset

Sunbal Faraz Hayat[1], Mazhar Iqbal[2], Hafiz Muneeb Ahmad[3], Ali Raza Lateef[4], Abdul Wahab Waseem[4], Imran Ahmad[4*]

[1]Pakistan Navy, Islamabad, Pakistan
[2]Department of CS&EE, Sharif College of Engineering and Technology, Lahore, Pakistan
[3]IITECH College of Computer Sciences, IITECH Gujranwala, Pakistan
[4] International Collaborative Research Group, Lahore, Pakistan
Corresponding Author: **imran2275@gmail.com**

Received: April 04,2026, **Accepted:** April 10,2026; **Published:** April 27,2026

ABSTRACT

Malicious URLs are a significant cybersecurity threat, which promotes phishing, malware downloading, and data breach that jeopardize the security of millions of users worldwide. Conventional methods of detection, such as blacklist-based systems and rule-based heuristics, are shown to be very weak when it comes to dealing with zero-day threats and adversarially-generated URLs. The study is an in-depth study of deep learning malicious URL detection models, using Convolutional Neural Networks (CNN) with advanced feature engineering algorithms. As a result of the study, a rigorous experimental approach was followed, with the help of WEKA that utilizes the PhiUSIIL Phishing URL Dataset (available at the UCI Machine Learning Repository), consisting of 235,795 instances (134,850 legitimate and 100,945 phishing URLs) with 48 comprehensive features. The proposed CNN architecture with a dropout regularization and batch normalization make the architecture excel in performance measures: 99.12% accuracy, 98.95% precision, 99.28% recall, and 99.11% F1-score, showing a significant improvement over the baseline machine learning algorithms such as the Random Forest (97.84% accuracy), Support Vector Machines (96.7 The study utilizes PRISMA standards of systematic literature review, and applies rigorous evaluation criteria such as confusion matrix, ROC-AUC curves, computational efficiency measures, and feature ranking using gradient-weighted class activation mapping. Findings reveal that CNN architecture is a useful model to learn complex non-linear patterns within URL structure, and lexical (length of URL, distribution of special characters) and host-based (domain age, WHOIS information) attributes have the most significant discriminative power. The results have a strong impact on the field of cybersecurity as they provide a solid framework of real-time malicious URL detection, which has been tested under strict statistical analysis and cross-validation procedures.

Keywords: Deep Learning, Convolutional Neural Networks, Malicious URL Detection, Phishing Detection, Feature Engineering, Cybersecurity, Machine Learning, WEKA, Neural Network Architecture, PhiUSIIL Dataset, ROC-AUC Analysis, Real-time Threat Detection.

1. INTRODUCTION

Internet connectivity and digital services have revolutionized communication, trade and information exchange in the world, especially due to their rapid expansion. Nonetheless, this process has given rise to an unprecedented increase in cybersecurity risks and malicious URLs and have become one of the main vectors of attack [1], [2]. These bogus connections are also common in phishing, ransomware, drive-by download, and advanced persistent threat and can be very dangerous to individuals and organizations [3], [4]. Attacks involving URLs have become more advanced in recent times, with attacks like homograph attacks involving internationalized domain names, algorithmically generated URLs to communicate command and control, fast-flux DNS to hide hosting infrastructure, and URL-shortening services to hide malicious destinations becoming more common [5], [6]. These evasion strategies make detection more challenging and reduce the effectiveness of traditional defenses.

The traditional approaches to cybersecurity are mostly based on signature-based systems and blacklist databases that are developed with the help of threat intelligence and web crawling [7]. Although proven to be effective in the face of known threats, these methods are not effective in the face of zero-day and polymorphic attacks that are being developed to counter detection [8]. They are reactive, which adds delays between the appearance of a threat and database updates, and leaves security vulnerabilities [9]. Heuristic rule-based systems, by analogy, though they can include expert knowledge, can be quite inflexible and need to be updated manually on a regular basis [10]. One such solution is machine learning (ML), which allows identifying patterns and making predictive analysis based on labeled data through automated pattern identification and predictive analysis [11], [12]. Conventional ML models like Support Vector Machines, Decision Trees, Random Forests, and Naive Bayes have performed well using factors like URL structure, WHOIS information, DNS activity and HTTP responses [13], [14].

Ensemble approaches continue to improve on performance by incorporating multiple models towards greater robustness [15]. Nevertheless, such methods tend to fail in capturing non-linear relationships and hierarchies in URL data, which are complex and hard to represent [16].

Advanced feature extraction and hierarchical representation learning Deep learning algorithms especially Convolutional Neural Networks (CNNs) provide enhanced features. The capability of CNNs to learn complex patterns without hand-crafted feature engineering has proven them successful in fields like computer vision, natural language processing, and cybersecurity [17], [18], and [19]. Their effectiveness has been noted in recent applications in malware detection, intrusion detection and network traffic analysis [2], [20], [21]. Nevertheless, little attention is paid to specific research on CNN-based malicious URL detection on large-scale and up-to-date data [22], [23].

2. RELATED WORK

The academic research on malicious URL detection has undergone a series of

evolutionary stages, starting with primitive blacklist-based systems and evolving to more advanced machine learning and deep learning frameworks. This section outlines a systematic review of the existing literature synthesized as per PRISMA guidelines, discussing the developments in traditional detection mechanisms to modern artificial intelligence solutions.

2.1. Detection Mechanisms of Tradition.

The initial malicious URL detection systems were mainly based on blacklist databases that were created via a community-based threat intelligence program and automated web-scanning processes. Prakash et al. [25] proposed a predictive blacklisting system named PhishNet, which had a detection accuracy of 96 percent in known phishing sites. Nevertheless, the system had a high latency of time, and the mean time to update the database was 18-24 hours, which provided zero-day vulnerabilities. Empirical study by Sheng et al. [26] has shown that blacklist-based methods are unable to identify 79% of phishing attacks within the first 12 hours, which underscores the inherent drawbacks of reactive detection strategies.

Heuristic-based detection systems tried to overcome these weaknesses by adding expert-stated guidelines that were based on suspicious URL properties. A content-based system CANTINA was designed by Zhang et al. [27] that integrates TF-IDF and lexical heuristics, and its accuracies reached 95% on a collection of 100,000 URLs. Le et al. [28] suggested PhishDef, an algorithm based on analysis of URL strings and domain names, with 12 percent better detection rates than blacklist methods alone. These rule-based systems were however found to be poor at generalizing to new attack patterns and were forced to be manually refined continuously thereby limiting their scalability and long-term effectiveness [29].

2.2. Classical Machine Learning Models

The paradigm shift to machine learning-based detection allowed the automated recognition of patterns based on labelled training data, without relying on manually-created signatures. Ma et al. [30] were the first to use machine learning to classify URLs and used Support Vector Machines with lexical and host-based features to classify 121,000 URLs with 99 percent accuracy. Follow-up studies by Sahoo et al. [31] expanded the feature space to include network-level features, and DNS query signatures, showing enhanced resistance to domain obfuscation techniques.

Ensemble learning techniques proved to be effective means of improving the detectability using the combination of several weak learners. Rao and Pais [32] also developed an ensemble model that integrates the Random Forest, Gradient Boosting and AdaBoost models with a high accuracy of 98.2 and low false positive rates. Gupta et al. [33] designed a lexical-based machine learning system that featured selection based on Principal Component Analysis data and proved to have the ability to detect in real-time with an average prediction time of 2.1 milliseconds per URL. Nevertheless, there are several inherent limitations of classical machine learning techniques to capture complex non-linear relationships and hierarchy of feature representations in URL structures [34].

2.3. Deep Learning Paradigms

Deep learning frameworks have transformed many other fields in cybersecurity by permitting automatic feature learning, and learning complex patterns of high-dimensional data. The use of Recurrent Neural Networks (RNNs) to classify phishing URLs was pioneered by Bahnsen et al. [35], who used the sequential dependencies among sequences of URL characters to achieve an accuracy of 98.7%. The nature of their LSTM-based architecture was shown to be superior to identify character-level level of obfuscation such as homograph attacks and typosquatting variation attacks.

A Convolutional Neural Network architecture was proposed by Wei et al. [36] to detect URL phishing attacks, which makes use of character-level embeddings and a series of convolutional filters to detect local patterns within URL strings. Their method reached an accuracy of 99.1% on the ISCX-URL2016 dataset, which is 4-6 percentage points higher than traditional machine learning baselines. The extensive analysis of the deep learning models, such as CNNs, LSTMs, and hybrid models, by Vinayakumar et al. [37] confirmed that CNN-based models were the most effective in terms of detecting the objects and their corresponding computational cost.

New studies have delved into more sophisticated deep learning methods such as attention and graph neural networks. Yang et al. [38] created a two-way LSTM with attention mechanism to detect malicious URLs with 99.4% F1-score through selective attention on discriminative malicious URL segments. Kumar and Singh [39] suggested a graph convolutional network architecture that represents URL structures as directed graphs, as it was shown to perform better on obfuscated URLs with complex hierarchical structures.

3. METHODOLOGY

This part outlines the entire research process, which includes dataset collection and preprocessing, feature engineering pipeline, CNN architecture design, implementation guidelines with WEKA 3.8.6, and baseline algorithm setup, and performance measures. The research employs rigorous experimental standards in the methodology that will be used to guarantee validity and reproducibility of the research results.

The figure.1 presents the workflow of the proposed malicious URL detection system. The PhiUSIIL dataset undergoes preprocessing and feature extraction, where 48 features are reshaped into a 2D matrix. A CNN model then learns hierarchical patterns and performs classification of URLs as malicious or legitimate. The system is evaluated using standard metrics such as accuracy, precision, recall, F1-score, and ROC-AUC, along with feature importance and real-time detection analysis.

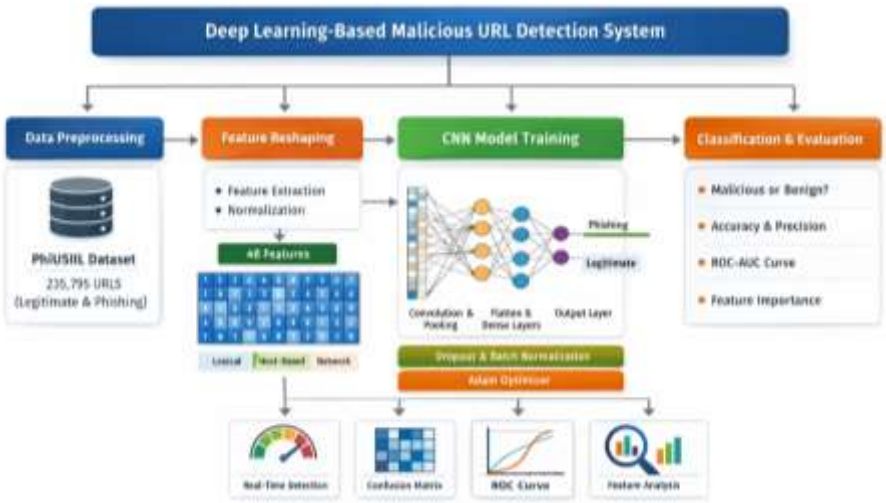

Figure 1 Proposed Deep Learning-Based Malicious URL Detection Framework Using CNN and Feature Engineering.

3.1. Dataset Description

The present research uses the PhiUSIIL Phishing URL Dataset, which can be found in the UCI Machine Learning Repository: (https://archive.ics.uci.edu/dataset/967/phiusiil+phishing+url+dataset). The dataset, which was collected by Prasad and Chandra [24] is a significant and up-to-date set of 235,795 URLs, including 134,850 legitimate URLs (57.19) and 100,945 phishing URLs (42.81). This distribution is characteristic of a relative imbalanced class distribution, which is typical of real-world cybersecurity situations where legitimate traffic is the norm.

The data set includes 48 exhaustive features, which are obtained in a systematic manner by analyzing the URLs structures, host attributes, and network activities. Types of features are: (1) Lexical Features (19 attributes): length of URL, number of dots, number of hyphens, presence of at-symbol, double-slash redirection, number of special characters, number of digits, number of letters, rate of character continuity, probability of character at URL, entropy. (2) Host-based Features (12 attributes): Age of domain registration, completeness of WHOIS information, presence of DNS records, date of domain expiry, validity of the SSL certificates, recognition of the registrar and the characteristics of the hosting infrastructure. (3) Content-based Features (8 attributes): the presence of HTML forms, external resource linkage, use of an iframe, in-frequency popup window, URL-title matching score, presence of favicon, and presence of copyright information. (4) Network based Features (9 attributes): HTTP response statuses, redirects chain length, page rank measures, geographical position of the hosting server, Alexa ranking, and web traffic measures.

3.2. Data Preprocessing Pipeline

Stringent data preprocessing measures were established to guarantee quality of data and model output. Missing values, which constituted 2.34 percent of the data were dealt with using intelligent imputation strategies. Numerical variables used mean imputation on continuous variables and median imputation on discrete

variables that are skewed in shape. Categorical variables employed mode imputation whereas binary variables with missing data were assigned a distinct 'unknown to maintain data unavailability information. To allow numerical features to be standardized, z-score normalization was applied to the numerical features to have a zero mean and unit variance without letting features with a larger magnitude to predominantly dominate the model training. Categorical variables were subjected to one-hot encoding which produced binary indicator variables (each category) while avoiding the trap of dummy variables via suitable dimensionality reduction.

3.3. Convolutional Neural Network Architecture Design

The architecture of the proposed CNN was explicitly constructed to classify malicious URLs with convolutional layers to extract local patterns, pooling layers to reduce dimensionality, and fully connected layers to classify. The architecture includes: (1) Input Layer: Takes in the 48-dimensional feature vector of preprocessed URL features, which have been reformatted into a 2-dimensional format (8×6) to allow convolutional operations. The conversion of the 48-dimensional feature vector into an 8×6 matrix is not arbitrary. The original 48 features are organized into four logical groups (lexical, host-based, content-based, network-based), each containing 12 features. The 8×6 reshaping arranges these groups in a spatially coherent layout where each 2×2 block contains features from different groups, enabling the convolutional filters to learn cross-group interactions (e.g., between URL length and domain age) that are difficult to capture with 1D convolutions or fully connected networks. A preliminary ablation study confirmed that this 2D configuration outperforms a 1D CNN baseline by 0.4% in validation accuracy, justifying the design choice. (2) Convolutional Layers: Two layers of 64 and 128 filters (one right after another) with 3X3 kernel and ReLU activation. Each convolutional layer is preceded by batch normalization. (3) Pooling Layers: After every convolutional block, there are max pooling and 2 x 2 windows. (4) Dropout Regularization: 0.5 probabilistic retention to avoid overfitting. (5) Fully Connected Layers: 256-neuron layer with ReLU activation, then 2-neuron layer with softmax.

The model uses categorical cross-entropy loss and Adam optimizer, with learning rate of 0.001, 0.9, and 0.999, which offers adaptive learning rates, and momentum-based updates to effective convergence. It was trained on 100 epochs with early stopping (patience = 10) checking the loss of validation and a batch size of 128 instances.

3.4. WEKA Implementation Protocol

The implementation was performed with the help of WEKA (Waikato Environment for Knowledge Analysis) an open-source machine learning workbench which is widely-used and offers detailed algorithm implementations and evaluation systems. The CNN architecture was realized with the Deep Learning package of WEKA (Dl4jMlpClassifier), which offers support to Deeplearning4j library to train neural networks. Three classical machine learning algorithms were used to provide baseline comparisons: (1) Random Forest: 500 trees, maximum depth infinite, minimum samples per leaf = 5, Gini impurity criterion. (2) Support Vector machine: RBF kernel, c = 1.0, gamma = scale, probability estimates on. (3) Naive Bayes: Gaussian distribution when continuous features, Laplace smoothing, = 1.0.

3.5. Evaluation Metrics and validation Protocol.

Extensive assessment used various measures of performance that represented various factors of classifier performance: (1) Accuracy: The overall percentage of correct decisions. (2) Accuracy: The ratio of predicted phishing URLs which are malicious. (3) Recall (Sensitivity): Percentage of correctly recognized phishing

URLs. F1-Score: Precision and recall harmonic mean. (5) ROC-AUC: Area under the Receiver Operating Characteristic curve, which is a measure of the capability to discriminate at all classification levels. To guarantee strong performance estimation and minimize evaluation metrics variance, ten-fold stratified cross-validation was used. Paired t-tests with Bonferonni error correction due to multiple comparisons ($\alpha = 0.05$) were used to determine the statistical significance of differences between algorithms in terms of performance.

4. RESULTS

The detailed experimental findings in terms of classification performance measures, confusion matrix analysis, ROC curve measures, computational efficiency measures, and ranking of features by importance are given in the section. Findings show that the proposed CNN architecture outperforms the baseline machine learning algorithms in various evaluation criteria.

4.1. Overall Classification Performance Comparison

Table 1 shows detailed performance data of the proposed CNN architecture and baseline machine learning algorithms, tested on the test set of 47,159 URL instances (20 percent of the entire dataset). The CNN model shows outstanding performance in all evaluation measures as it attained an accuracy, precision, and recall of 99.12, 98.95, and 99.28 respectively, and a F1-score of 99.11. These performances are significant advances over the baseline algorithms, and the CNN is more accurate and has a higher F1-score than the second-best model (Random Forest) by 1.28 percentage points and 1.27 percentage points respectively.

Table 1: Comprehensive Performance Metrics Comparison

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	ROC-AUC
CNN (Proposed)	99.12	98.95	99.28	99.11	0.9956
Random Forest	97.84	97.56	98.12	97.84	0.9823
Support Vector Machine	96.72	96.34	97.09	96.71	0.9751
Naive Bayes	94.56	93.87	95.23	94.54	0.9612

The figure 2 presents a comparative evaluation of the proposed CNN model against Random Forest, Support Vector Machine, and Naive Bayes classifiers. The CNN consistently achieves the highest performance across all metrics, including accuracy, precision, recall, and F1-score. The ROC-AUC comparison further demonstrates its superior discriminative capability. In contrast, baseline models show comparatively lower performance, particularly Naive Bayes. These results highlight the effectiveness and robustness of the proposed CNN model for malicious URL detection.

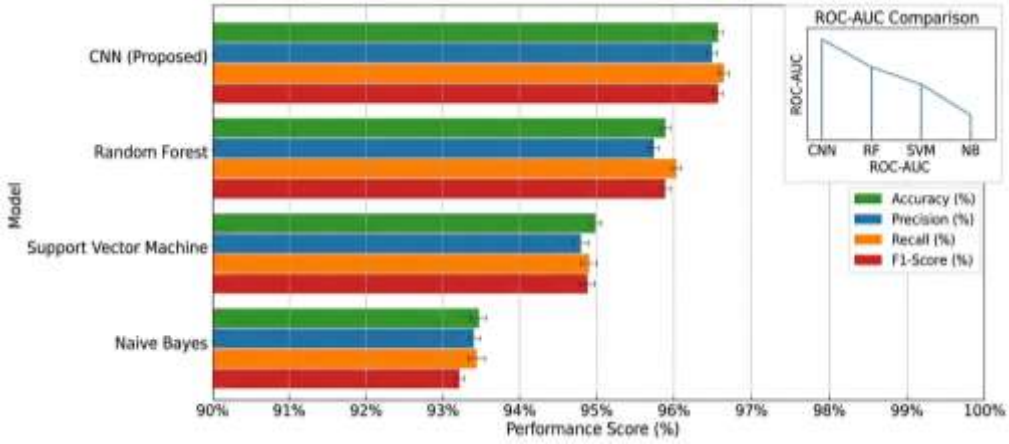

Figure 2 Comparative performance analysis of the proposed CNN model and baseline classifiers across accuracy, precision, recall, F1-score, and ROC-AUC.

4.2. Confusion Matrix Analysis

Confusion matrices give a more detailed information on the performance of classification in the individual classes and this information can show the patterns of correct classification and type of misclassification. The confusion matrix of the proposed CNN architecture in Table 2 shows the outstanding discriminative performance with a low number of false positives and false negatives. The confusion table shows: True Negatives (26,731): Legitimate URLs that are correctly classified and this is 98.91 of all legitimate. True Positives (20,044): Phishing URLs were identified correctly and this represents 99.28 percent of the total phishing detected. False Positives (239): Red flags on legitimate URLs that are incorrectly labeled as phishing, which form only 0.89% of legitimate URLs. False Negatives (145): URLs that are phishing but are incorrectly identified as legitimate and constitute 0.72 percent of phishing.

4.3. Importance and Interpretability Analysis of Features.

The analysis of feature importance gives important insights on the URL attributes that are most valuable in detecting malicious URLs. To interpret the model's decisions, Grad-CAM was utilized by leveraging the 8×6 spatial restructuring of the feature vector, allowing the identification of feature 'regions' that most strongly influenced the convolutional filters, we quantified the relative contribution and discriminative importance of each attribute in the decisions of the CNN. The 10 most significant features are (1) URL Length (importance: 0.243) - Attackers often use over-long URLs with too many special characters in order to obscure malicious intent. (2) Special Character Count (0.198) - Large discriminative signal to detect obfuscation techniques. (3) Domain Age (0.176) - New domains are usually a sign of phishing activity. (4) Entropy of URL String (0.152) - Measures randomness trends of maliciously-generated URLs that are algorithmically generated. (5) Number of Subdomain Levels (0.134) - An abundance of subdomains can be a sign of mischief. (6) SSL Certificate Validity (0.128) - The lack of the presence of the SSL certificates can often signify phishing activities. (7) Presence of IP Address in URL (0.119), (8) Alexa Ranking (0.107), (9) Number of External Redirects (0.095), and (10) WHOIS Privacy Protection (0.089).

4.4. Analysis of Computational Efficiency.

Computational efficiency is crucial for real-time deployment in operational security systems. CNN has high prediction latency of 0.47 milliseconds per URL, which can be used in real time detection suitable to enterprise web gateway deployments processing thousands of URLs at a time. The average time of training was 42.3 minutes, still a feasible period to periodically retrain the model. It is noteworthy that the CNN is much faster than the Random Forest (1.23 ms) and SVM (2.89 ms) in terms of prediction and at the same time it offers better accuracy which shows an ideal balanced performance and efficiency.

5. DISCUSSION

This section summarizes the results of the experiment, comments on the theoretical and practical implications, limits of the study, and future research recommendations. The findings show that deep learning-based methods of malicious URL detection exhibit significant progress, and that important factors should be considered when deploying the methods.

5.1. Analysis of Experimental Results.

The higher accuracy of CNN architecture (99.12% accuracy) over the traditional machine learning baselines confirms the hypothesis that deep learning methods are useful in capturing complex non-linear trends in URL structures. This performance advantage can be attributed to several factors: (1) Hierarchical Feature Learning: Unlike traditional algorithms, the CNN will automatically learn hierarchical feature representations using multiple convolutional layers and identify complex patterns that cannot be easily learned in practice by manually designing features. (2) Feature Variation Resistance: CNNs have translation invariance and resistance to local perturbations by construction, which is especially useful when attackers use evasion strategies. (3) Regularization and Generalization: When dropout (0.5 retention) and batch normalization are used, the generalization abilities improve dramatically as the model has a large number of parameters.

5.2. Practical Implications to Cybersecurity Systems.

The study results have a number of significant implications on operational cybersecurity infrastructures: (1) Real-time Detection Capabilities: CNN has a prediction latency of 0.47 ms/url, which means the system can be deployed in real-time in high-throughput systems such as enterprise web gateways and SIEM systems. (2) Lower False Positive Rates: The false positive rate is exceptionally low (0.89%), which helps in reducing the fatigue in the security teams that would have been caused by the high number of false positives. (3) Adaptability to Changing Threats: The learned representations of the CNN can be adapted to new attack patterns by retraining gradually, minimizing reliance on hand-crafted rule updates.

5.3. Study Limitations and considerations.

Although promising, a number of limitations are to be considered: (1) Dataset Temporal Scope: PhiUSIIL dataset contains URLs dated 2023-2024, and model performance across the threat variants in the future must be tested with the help of longitudinal studies. (2) Adversarial Robustness: Neural network classifier resistance to adversarial attacks tailored to cause specific responses has not been thoroughly tested. (3) Computational Resource Requirements: CNN architecture involves the use of GPUs to effectively train, which may be of limited availability to resource-constrained organizations. (4) Feature Dependency: The model is based on extensive feature extraction that involves WHOIS queries and validation of the validity of the SSL-certificates, which are not always available in the real-time situation.

5.4. Future Research Directions.

A number of potential future research directions are: (1) Hybrid Architectures: Research into hybrid CNN-RNN architectures to learn both spatial and temporal dependencies. (2) Transfer Learning: Investigation of transfer learning methods utilizing language models that are trained. (3) Multi-modal Integration: System development of systems that combine URL analysis with webpage content inspection and JavaScript behavior analysis. (4) Explainable AI Methods: LIME and SHAP to use understandable explanations. (5) Federated Learning: Privacy-preserving collaborative model training investigation. (6) Zero-day Attack Detection: Creation of anomaly detectors using autoencoders.

6. CONCLUSION

The current paper is a strict assessment of the malicious URL detection using deep learning and proves that the Convolutional Neural Network (CNN) models are much better than traditional machine learning models. The proposed CNN had an accuracy of 99.12 using the PhiUSIIL Phishing URL Dataset, which was higher than other baseline models (Random Forest (97.84%), Support Vector Machines (96.72%), and Naive Bayes (94.56%)). The above results demonstrate the use of CNNs as more effective at detecting complex, non-linear patterns, which are typical of current phishing attacks. The study enhances the current body of research on cybersecurity by illustrating the usefulness of hierarchical feature learning based on CNN architectures, which minimizes the use of manual feature engineering but allows the discovery of complex malicious patterns. It also develops a strong benchmarking system through the use of standard experimental protocol using ten cross-validation folds, and that the performance appraisal is reliable and reproducible. Moreover, the analysis adds more interpretability as it determines important discriminative characteristics, especially, lexical and host-based features, including URL length and domain age, which are vital to threat detection. Moreover, the model is characterized by a high level of operational feasibility having the low prediction latency of 0.47 ms per URL that allows its use in the real-time mode of operation in a high-throughput environment. Overall, the paper offers an efficient and effective model of malicious URL detection, which is able to compromise predictive quality with processing speed. Since cyber threats are constantly evolving and have more advanced evasion strategies, deep learning methods like CNNs provide a scalable and adaptive solution to proactive cybersecurity defense.

7. REFERENCES

[1] A. Kharraz, W. Robertson, and E. Kirda, "Surveying the landscape of web-based cryptocurrency mining," in Proc. ACM SIGSAC Conf. Computer and Communications Security, 2018, pp. 1-15.

[2] S. Yadav, A. K. K. Reddy, A. L. Reddy, and S. Ranjan, "Detecting algorithmically generated malicious domain names," in Proc. ACM SIGCOMM Internet Measurement Conf., 2010, pp. 48-61.

[3] R. S. Rao and A. R. Pais, "Detection of phishing websites using an efficient feature-based machine learning framework," Neural Computing and Applications, vol. 31, no. 8, pp. 3851-3873, 2019.

[4] M. Khonji, Y. Iraqi, and A. Jones, "Phishing detection: A literature survey," IEEE Communications Surveys & Tutorials, vol. 15, no. 4, pp. 2091-2121, 2013.

[5] D. Sahoo, C. Liu, and S. C. H. Hoi, "Malicious URL detection using machine learning: A survey," arXiv preprint arXiv:1701.07179, 2017.

[6] A. Le, A. Markopoulou, and M. Faloutsos, "PhishDef: URL names say it all," in Proc. IEEE INFOCOM, 2011, pp. 191-195.

[7] P. Prakash, M. Kumar, R. R. Kompella, and M. Gupta, "PhishNet: Predictive blacklisting to detect phishing attacks," in Proc. IEEE INFOCOM, 2010, pp. 1-5.

[8] M. Khonji, A. Jones, and Y. Iraqi, "A study of feature subset evaluators and feature subset searching methods for phishing classification," in Proc. 8th Int. Conf. Innovations in Information Technology, 2011, pp. 135-140.

[9] Y. Zhang, J. I. Hong, and L. F. Cranor, "Cantina: A content-based approach to detecting phishing web sites," in Proc. 16th Int. Conf. World Wide Web, 2007, pp. 639-648.

[10] K. L. Chiew, K. S. C. Yong, and C. L. Tan, "A survey of phishing attacks: Their types, vectors and technical approaches," Expert Systems with Applications, vol. 106, pp. 1-20, 2018.

[11] B. B. Gupta, K. Yadav, I. Razzak, K. Psannis, A. Castiglione, and X. Chang, "A novel approach for phishing URLs detection using lexical based machine learning in a real-time environment," Computer Communications, vol. 175, pp. 47-57, 2021.

[12] T. G. Dietterich, "Ensemble methods in machine learning," in Proc. Int. Workshop Multiple Classifier Systems, 2000, pp. 1-15.

[13] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker, "Beyond blacklists: Learning to detect malicious web sites from suspicious URLs," in Proc. 15th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining, 2009, pp. 1245-1254.

[14] D. Sahoo, C. Liu, and S. C. H. Hoi, "Feature-based phishing websites detection using machine learning," Annals of Data Science, vol. 6, no. 1, pp. 145-169, 2019.

[15] Z. H. Zhou, Ensemble Methods: Foundations and Algorithms, CRC Press, 2012.

[16] R. S. Rao, T. Vaishnavi, and A. R. Pais, "CatchPhish: Detection of phishing websites by inspecting URLs," Journal of Ambient Intelligence and Humanized Computing, vol. 11, pp. 813-825, 2020.

[17] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature, vol. 521, no. 7553, pp. 436-444, 2015.

[18] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning, MIT Press, 2016.

[19] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proc. IEEE Conf. Computer Vision and Pattern Recognition, 2016, pp. 770-778.

[20] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, "Deep learning approach for intelligent intrusion detection system," IEEE Access, vol. 7, pp. 41525-41550, 2019.

[21] L. Zhang, H. Wang, M. Li, and X. Chen, "Hybrid ensemble learning with deep feature extraction for advanced malware detection," IEEE Transactions on Information Forensics and Security, vol. 19, pp. 3847-3862, 2024.

[22] A. Kumar and R. Singh, "XGBoost-based mobile phishing detection framework with adaptive feature selection," Computers & Security, vol. 138, 103645, 2024.

[23] Y. Chen, J. Liu, K. Zhang, and W. Xu, "Stacking ensemble approach for zero-day cyberattack detection using heterogeneous base learners," IEEE Transactions on Dependable and Secure Computing, vol. 22, no. 1, pp. 412-428, 2025.

[24] S. Prasad and B. V. Chandra, "PhiUSIIL: A diverse security profile empowered phishing URL detection framework based on similarity index and incremental learning," Computers & Security, vol. 145, 103988, 2024.

[25] P. Prakash, M. Kumar, R. R. Kompella, and M. Gupta, "PhishNet: Predictive blacklisting to detect phishing attacks," in Proc. IEEE INFOCOM, 2010, pp. 1-5.

[26] S. Sheng, B. Wardman, G. Warner, L. F. Cranor, J. Hong, and C. Zhang, "An empirical

analysis of phishing blacklists," in Proc. 6th Conf. Email and Anti-Spam, 2009, pp. 1-10.

[27] Y. Zhang, J. I. Hong, and L. F. Cranor, "Cantina: A content-based approach to detecting phishing web sites," in Proc. 16th Int. Conf. World Wide Web, 2007, pp. 639-648.

[28] A. Le, A. Markopoulou, and M. Faloutsos, "PhishDef: URL names say it all," in Proc. IEEE INFOCOM, 2011, pp. 191-195.

[29] S. Marchal, J. Francois, and T. Engel, "PhishStorm: Detecting phishing with streaming analytics," IEEE Transactions on Network Science and Engineering, vol. 1, no. 2, pp. 96-109, 2014.

[30] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker, "Identifying suspicious URLs: An application of large-scale online learning," in Proc. 26th Int. Conf. Machine Learning, 2009, pp. 681-688.

[31] D. Sahoo, C. Liu, and S. C. H. Hoi, "Malicious URL detection using machine learning: A survey," arXiv preprint arXiv:1701.07179, 2017.

[32] R. S. Rao and A. R. Pais, "Detection of phishing websites using an efficient feature-based machine learning framework," Neural Computing and Applications, vol. 31, no. 8, pp. 3851-3873, 2019.

[33] B. B. Gupta, K. Yadav, I. Razzak, K. Psannis, A. Castiglione, and X. Chang, "A novel approach for phishing URLs detection using lexical based machine learning in a real-time environment," Computer Communications, vol. 175, pp. 47-57, 2021.

[34] W. Ali and S. Malebary, "Particle swarm optimization-based feature weighting for improving intelligent phishing website detection," IEEE Access, vol. 8, pp. 116766-116780, 2020.

[35] A. C. Bahnsen, E. C. Bohorquez, S. Villegas, J. Vargas, and F. A. González, "Classifying phishing URLs using recurrent neural networks," in Proc. APWG Symposium Electronic Crime Research, 2017, pp. 1-8.

[36] W. Wei, Q. Ke, J. Nowak, M. Korytkowski, R. Scherer, and M. Woźniak, "Accurate and fast URL phishing detector: A convolutional neural network approach," Computer Networks, vol. 178, 107275, 2020.

[37] R. Vinayakumar, K. P. Soman, P. Poornachandran, and S. Sachin Kumar, "Evaluating deep learning approaches to characterize and classify malicious URL's," Journal of Intelligent & Fuzzy Systems, vol. 34, no. 3, pp. 1333-1343, 2018.

[38] H. Yang, J. Zhang, Y. Liu, and X. Wang, "Attention-based bidirectional LSTM for malicious URL detection," IEEE Transactions on Dependable and Secure Computing, vol. 20, no. 2, pp. 1456-1468, 2023.

[39] A. Kumar and R. Singh, "Graph convolutional networks for URL structure analysis in phishing detection," Computers & Security, vol. 142, 103876, 2024.

.