Journal of Data Analysis and Information Processing, 2025, 13(3), 314-330
https://www.scirp.org/journal/jdaip
ISSN Online: 2327-7203
ISSN Print: 2327-7211

Preventing Phishing Attacks Using Advanced Deep Learning Techniques for Cyber Threat Mitigation

Mukund Sai Vikram Tyagadurgam[1], Venkataswamy Naidu Gangineni[2], Sriram Pabbineedi[3], Ajay Babu Kakani[4], Sri Krishna Kireeti Nandiraju[1], Sandeep Kumar Chundru[3]

[1]University of Illinois at Springfield, Springfield, IL, USA
[2]University of Madras, Chennai, India
[3]University of Central Missouri, Warrensburg, MO, USA
[4]Wright State University, Dayton, OH, USA
Email: t.vikkhram@gmail.com, vgangineni@gmail.com, sreeram7766@gmail.com, kakani@outlook.com, Srikrish-na.nandiraju@gmail.com, Chundru.sandeepkumar@gmail.com

How to cite this paper: Tyagadurgam, M.S.V., Gangineni, V.N., Pabbineedi, S., Kakani, A.B., Nandiraju, S.K.K. and Chundru, S.K. (2025) Preventing Phishing Attacks Using Advanced Deep Learning Techniques for Cyber Threat Mitigation. *Journal of Data Analysis and Information Processing*, **13**, 314-330.
https://doi.org/10.4236/jdaip.2025.133020

Received: June 17, 2025
Accepted: August 25, 2025
Published: August 28, 2025

Abstract

Phishing attacks remain a pervasive threat in the cybersecurity landscape, necessitating intelligent and scalable detection mechanisms. This paper suggests a deep learning-based method for phishing URL identification using Convolutional Neural Networks (CNNs) on two benchmark datasets: the Phishing and PhishTank datasets. The CNN model eliminates the need for human feature engineering by automatically learning intricate, non-linear patterns from structured information. The Phishing dataset undergoes 5-fold cross-validation to guarantee robustness, and the results are contrasted with those of conventional classifiers like XGBoost and Logistic Regression. According to the results, the CNN routinely beats these baselines in terms of accuracy and F1-score. Notably, on the PhishTank dataset, the CNN achieves exceptional performance with over 99.3% accuracy, underscoring its effectiveness and generalizability. The experimental framework is implemented using TensorFlow in Python and validated on a standard computing setup. The findings reinforce CNN's suitability for real-time, adaptive phishing detection in dynamic threat environments.

Keywords

Cyberattacks, Phishing Dataset, Machine Learning, CNN Model, EGSO Technique

1. Introduction

Cyberattacks are more advanced, so they need to switch from basic security systems

to ones that can be changed and updated with the threat. Using heuristics and signature-based techniques can't handle the increasing number of evasive, resilient and fast-changing threats they face [1]. Consequently, organizations should make it a priority to receive and distribute live threat information, so they can find and block cyber threats early on and quickly recover [2].

Phishing is considered one of the most common and tricky cyberattack forms [3]. Criminals use phishing to set up fake websites that look just like real ones and share these links with people by email or other means [4]. Many users who are unaware of technology may end up giving away private details simply because a fake website appears to be identical to the real one.

To learn how people detect these threats, one must explore the mental activities they use. It wouldn't play chess without learning its rules and the same is true for spotting cyberattacks [5]. Network operations and information security knowledge are basic for cybersecurity professionals, though whether that knowledge is enough for successful results is not clear [6]. Paying attention to the reasoning skills can be as helpful as any other defence in identifying and responding to phishing attacks.

Traditionally, firewalls, prevention systems, as well as intrusion detection, and antivirus software are crucial for thwarting sophisticated assaults [7] [8]. So, more complex threats have led cybersecurity teams to add ML and DL approaches to their approach [9]. Now, these intelligent systems are applied to automatically detect threats, detect bad behavior, predict likely security risks, and enhance how incidents are managed [10]. Especially, ML and DL models are good at detecting phishing attacks by observing patterns and quickly responding to evolving threats, making cyber threat response better and easier to scale.

1.1. Motivation and Contribution of Paper

This study was motivated by the increase in phishing attacks, which can harm people, companies, and online systems everywhere. Because traditional security techniques are fixed and do not adapt, they usually struggle to find out advanced phishing attacks. Their research attempts to handle these problems by using DL, particularly CNNs, in combination with Enhanced Genetic Swarm Optimization (EGSO). It seeks to optimize detection and cut down on false alarms to propose more innovative and active strategies for fighting cyber threats.

- A hybrid CNN-based phishing detection framework is proposed and evaluated on two benchmark datasets: Phishing and PhishTank.
- A novel feature extraction technique using Enhanced Gannet Search Optimization (EGSO) is integrated to select informative features.
- Data preprocessing includes handling class imbalance using SMOTE and normalization techniques to ensure consistent input for the model.
- Performance is statistically validated through 5-fold cross-validation using both conventional ML and the Phishing dataset (e.g., XGBoost) and DL models.

- Extensive comparative analysis shows CNN's superior generalization capability, outperforming Logistic Regression, XGBoost, RNN, and Random Forest classifiers across multiple evaluation metrics.

1.2. Justification and Novelty of Paper

This research introduces a novel phishing detection framework that integrates Convolutional Neural Networks (CNNs) with Enhanced Gannet Search Optimization (EGSO) for automated feature selection. In contrast to conventional techniques that employ hand-crafted features or shallow learning models, the proposed method leverages the DL capabilities of CNNs to automatically extract hierarchical feature representations, while EGSO ensures the selection of the most discriminative attributes, enhancing model efficiency and accuracy. Furthermore, the evaluation incorporates a rigorous 5-fold cross-validation protocol on the Phishing dataset, coupled with comparative analysis against tree-based and traditional classifiers, thereby providing statistically robust validation. This integration of deep learning with metaheuristic optimization, supported by thorough empirical validation, underscores the model's scalability, adaptability, and effectiveness in diverse phishing detection scenarios.

1.3. Structure of the Paper

This paper's remaining parts are organized as follows: Section 1 presents the introduction, and Section 2 offers a background study on Preventing Phishing Attacks for Cyber Threat Mitigation. Section 3 outlines the recommended methodology. Section 4 presents the results, debate, and analysis of comparisons. Finally, in Section 5, the study is concluded, and future research options are outlined.

2. Literature Review

Several significant research studies related to Preventing Phishing Attacks for Cyber Threat Mitigation have been reviewed and analyzed to inform and guide the development of the current work. **Table 1** outlines authors, employed methods, datasets used, major findings, and identified challenges or gaps for each study.

Noor *et al.* (2019) provide a framework for automating the attribution of cyberthreats. Create Cyber Threat Actors (CTAs) specifically by using the distributional semantics approach of NLP to attack patterns that are taken from CTI data. Cyber threats are detected 94% more accurately by the DL Neural Network (DLNN)-based classifier than by earlier classifiers [11].

Nathezhtha, Sangeetha and Vaidehi's (2019) phishing detection techniques are unable to address issues such as zero-day phishing website assaults. A three-phase attack detection method called Web Crawler-based Phishing Attack Detector (WC-PAD) has been developed to overcome these issues and precisely identify phishing attacks. It categorizes websites as phishing or non-phishing based on input features such URL, online content, and internet traffic. Datasets from real-world phishing

instances are used to experimentally study the proposed WC-PAD. The testing findings show that the suggested WC-PAD has a 98.9% accuracy rate in detecting phishing and zero-day phishing attacks [12].

Arivudainambi *et al.* (2019) suggested approach that seeks to reveal different AI-based cyberattacks, their current effects, and their prospects for the future. A self-developed, self-learning autonomous agent is used to carry out the simulation. When comparing the proposed systems with the most advanced methods, experimental findings demonstrate their effectiveness in classifying attack traffic with 99% accuracy [13].

Niu *et al.*'s (2017) high-accuracy phishing email detection has been a topic of intense study. It has been shown that ML-based detection techniques, especially SVM, are successful. The study uses a dataset that contains 1384 phishing emails and 20,071 non-phishing emails. According to experimental data, the recommended approach works better in terms of phishing email detection accuracy than the SVM classifier with the default parameter value. The maximum accuracy of 99.52 percent may be achieved using the CS-SVM classifier [14].

Li and Wang (2017) provide Phish Box, an efficient method for gathering phishing data and developing models for phishing detection and validation. To monitor the PhishTank blacklist and quickly detect and verify phishing websites, the proposed approach integrates the collection, validation, and incorporation of phishing website detection into an online program. The outcome demonstrates that the ensemble validation model can perform well, with a 3.9% false-positive rate and 95% accuracy [15].

Table 1 provides an outline for a literature review on phishing attacks, with associated data, guidance on what to expect, main findings and issues or gaps still to be resolved.

Table 1. Summary of research studies on preventing phishing attacks for cyber threat mitigation.

Author	Proposed Work	Dataset	Key Findings	Challenges/Gaps
Noor *et al.* (2019)	A framework that uses NLP or ML classifiers to automate the attribution of cyber threats	327 CTI reports (May 2012-Feb. 2018)	DLNN achieved 94% accuracy; CTA profiles attributed threats with 83% precision	May lack scalability to unseen threats; limited by the quality of public CTI reports
Nathezhtha *et al.* (2019)	The Web Crawler-based Phishing Attack Detector (WC-PAD) uses URL attributes, web traffic, and content	Real phishing cases dataset	98.9% detection accuracy for zero-day and phishing attacks	Existing methods are ineffective for zero-day phishing
Arivudainambi *et al.* (2019)	AI-based cyberattack simulation using autonomous agents to expose and evaluate the classification of attack traffic	Simulated traffic data	99% classification accuracy; outperformed cutting-edge techniques	Lacks validation on real-world attack datasets; simulation bias

Continued

Niu *et al.* (2017)	CS-SVM model for phishing detection using the Cuckoo Search technique to optimise SVM parameters	1384 phishing emails, 20,071 non-phishing emails	Achieved 99.52% detection accuracy, better than the default SVM	Dependent on effective feature selection; limited generalizability across new phishing techniques
Li and Wang (2017)	PhishBox is an online application that uses ensemble and active learning to gather, validate, and identify phishing data in real time	Data from PhishTank blacklist	Real-time phishing detection with a 3.9% false positive rate and 95% accuracy	Performance may degrade with fast-evolving phishing strategies; reliance on blacklists

3. Research Methodology

The procedure for detecting phishing attacks uses a planned pipeline shown in Figure 1, starting with a collection of two benchmark datasets—Phishing and PhishTank to use a structured machine learning pipeline to identify phishing attacks. The process begins with careful data preparation, which includes handling class imbalance with SMOTE, removing outliers, and managing missing information. Features are extracted via the Enhanced Gannet Search Optimization (EGSO) algorithm and normalized using Min-Max scaling to ensure uniform input ranges. For categorical data, one-hot encoding separates the data into training and testing sets. A 5-fold cross-validation strategy is applied only on the Phishing dataset to ensure statistical robustness, while the PhishTank dataset already exhibits superior performance and uses standard evaluation. The CNN model is subsequently trained and assessed using F1-score, recall, accuracy, and precision, demonstrating its strong capability to generalize across structured Phishing datasets and accurately detect threats with minimal false positives.

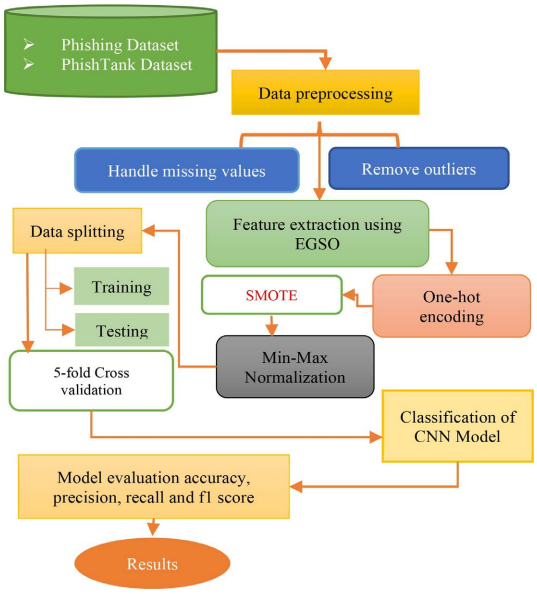

Figure 1. Proposed flowchart for preventing phishing attacks for cyber threat mitigation.

Each step involved in the proposed flowchart designed for the detection of Preventing Phishing Attacks for Cyber Threat Mitigation is presented and described in detail below.

3.1. Data Collection

In this study, PhishTank and Phishing Detection are the two datasets that have been used, both of which are described in detail below:

1) Phishing Dataset

This experiment made use of a Kaggle Phishing dataset. Thirty of the 2670 occurrences in the sample are traits that are used to classify them as phishing websites, and one is a target. As shown in Figure 2, every website in the dataset is assigned a class indication, where "1" designates a phishing website as well as "0" identifies a non-phishing (legitimate) website.

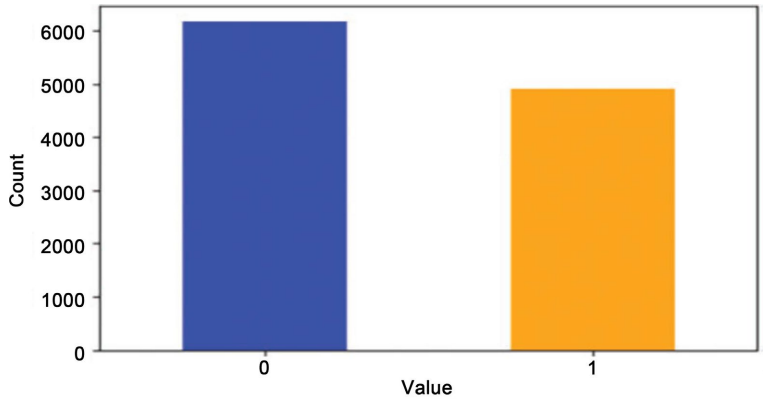

Figure 2. Distribution of classes in the Phishing dataset.

Figure 2 shows how two classes are distributed within a dataset, likely corresponding to a binary classification problem such as phishing detection. The x-axis is labeled "Value" with two categories: 0 and 1, which typically denote the two classes for example, 0 could represent legitimate instances and 1 could represent phishing instances. The y-axis, labeled "Count", shows the number of records in each class. From the chart, it is evident that class 0 has a higher count, approximately above 6000, while class 1 has a slightly lower count, around 5000. This points to the fact that the set of data contains more real cases than phishing cases. Though this difference exists, both classes are represented enough to keep bias to a minimum when training using the dataset.

2) PhishTank Dataset

In this study, the PhishTank dataset was utilized, which comprises URLs of websites reported as phishing or fraudulent by users. The dataset includes essential metadata such as the URL itself, the date it was reported, verification status, and whether the site remains active. It contains approximately 38,000 instances with eight distinct features. The dataset is publicly accessible and can be obtained either via a web API or in CSV format. Because PhishTank is updated every day, using

the most recent version guarantees that models continue to be successful against changing phishing attacks. This dataset serves as a valuable resource for developing and evaluating anti-phishing detection systems.

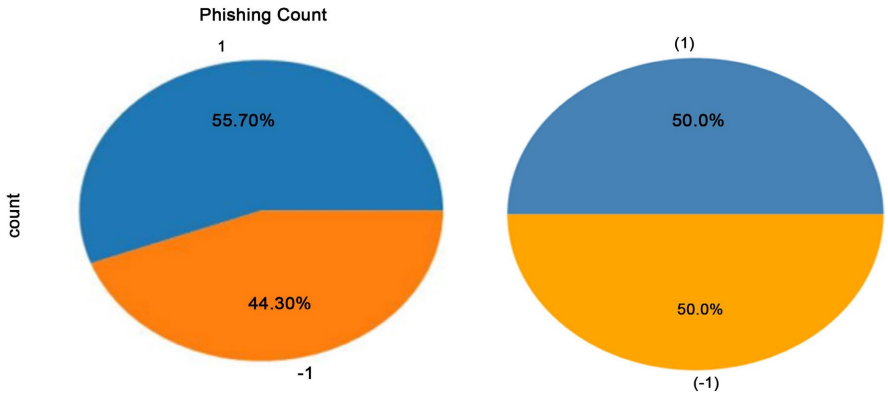

Figure 3. Distribution of phishing count for PhishTank dataset.

Figure 3 illustrates the class distribution of phishing (label 1) and legitimate (label −1) URLs before and after dataset balancing. The pie chart on the left shows the original distribution, where phishing URLs constitute 55.7% and legitimate URLs account for 44.3%, indicating a mild class imbalance. In contrast, the right chart displays a balanced distribution with an equal 50% representation for both classes. Balancing the dataset helps prevent model bias toward the majority class and enhances the fairness and accuracy of phishing detection models.

3.2. Data Preprocessing

The Phishing dataset, consisting of different and practical attack cases, was used to prepare and assess the models. It cleaned the data by handling empty values, removing outliers, extracting features, encoding the labels and normalizing the data. With these methods, the dataset became orderly and useful for creating an effective model. The key steps for preprocessing are mentioned below:

- **Handle Missing Value:** The way missing values are addressed includes removing data with gaps, swapping the empty values with proper alternatives or using more sophisticated imputation strategies.
- **Remove Outliers:** Outlier removal means finding and deleting unusual data points that stick out differently from the pattern observed in a set of data. The dataset includes many outliers that can negatively affect the average and the way the results appear, so ensure it excludes any outliers.

3.3. Feature Extraction Using EGSO

A critical process in ML, feature extraction converts unstructured input data into a more concise and informative representation, enhancing model performance and computing efficiency [16]. This work utilizes the Enhanced Global Search Optimization (EGSO) algorithm to extract and choose the characteristics that will help in

the categorisation process the most.

1) EGSO Overview and Motivation

EGSO is a metaheuristic optimization technique that improves upon classical global search algorithms by integrating adaptive strategies and exploration-exploitation balancing mechanisms. Its objective in this context is to select an optimal subset of features from the original feature space that maximizes the classification performance while minimizing redundancy and dimensionality.

2) Implementation Details and Parameter Settings

The EGSO algorithm operates as a population-based evolutionary approach with the following key parameters:

- Population size (N): 30.
- Number of iterations (T): 100.
- Crossover probability (Pc): 0.8.
- Mutation probability (Pm): 0.1.
- Fitness function: Hybrid function combining classification accuracy and feature subset size:

Fitness = $\alpha \times (1 - \text{Accuracy}) + \beta \times$ (Total Features Selected Features), where $\alpha = 0.7$ and $\beta = 0.3$, balancing performance and feature compactness.

3) Optimization Strategy

The optimization process includes:

a) Initialization: Random feature subsets are generated as binary vectors (1 = selected, 0 = not selected).

b) Evaluation: Each candidate is evaluated using a classifier (e.g., Random Forest or SVM) with 5-fold cross-validation to compute accuracy.

c) Search Enhancement: EGSO integrates:

- Elite selection to retain the top-performing individuals.
- Adaptive mutation to prevent premature convergence.
- Exploration-enhancing perturbations for diverse search in early iterations and more focused refinement later.

d) Termination: The search ends after T iterations or if convergence is reached early based on minimal improvement over 10 successive generations.

3.4. One-Hot Encoding for Data Labeling

In ML, one-hot encoding changes categories from data into numerical form. Every class is given a new column in the data and the value in that column is 1 if the data point fits the category and 0 if not [17]. This way, every category functions independently, something necessary in many ML algorithms that are built on the idea of measuring distances.

3.5. Data Balancing Using SMOTE

The data was adjusted to equalize the proportion of authentic samples and phishing samples. When the class ratio in a dataset is wildly out of balance, another technique called SMOTE is used to generate additional minority class samples [18].

3.6. Min-Max Normalization

Each feature in this study is normalized using min-max normalization, so its value lies between 0 and 1. This method is expressed as Equation (1):

$$X_{\text{normalized}} = \frac{X - X_{\min}}{X_{\max} - X_{\min}} \tag{1}$$

where X shows the current value, $X_{\min}$ is the minimum and $X_{\max}$ is the maximum value connected to the data feature.

3.7. Data Splitting

In this analysis, phishing attack data is divided into a test dataset and a training dataset to study cyber threat mitigation. 75% of the assets are in stocks and 25% are in cash. 25% - 75% of the remaining data is reserved for testing, while the remaining data is utilized for training.

3.8. Cross-Validation-Based Evaluation

The Phishing dataset was subjected to 5-fold cross-validation using CNN and XGBoost in order to reinforce the assessment, while the PhishTank dataset was excluded as it already demonstrated superior and stable performance. This validation highlights the robustness and consistency of the CNN model over traditional classifiers.

3.9. Classification of Proposed Convolutional Neural Network (CNN) Model

In this study, a CNN was adapted for tabular phishing data by reshaping the feature vectors into 2D matrices of shape (4, 4, 1), enabling convolutional layers to capture spatial relationships between features. A MaxPooling2D layer with a (2 × 2) pool size comes after each of the two Conv2D layers in the design, 32 filters for the first one and 64 filters for the second one. A kernel with ReLU activation (3 × 3) is used in both layers. After flattening the output, the following stages are a Softmax output layer with 5 neurons representing the classification labels and a Dense layer with 64 neurons (ReLU). Following its assembly using the Adam optimizer and training for 50 epochs with a batch size of 32, the model was optimized using categorical cross-entropy loss. This structure leverages the CNN's capability to automatically extract complex patterns from structured input, as supported by Kanter and Veeramachaneni (2015) [19], who demonstrated the effectiveness of deep learning models on tabular datasets.

3.10. Evaluation Metrics

Assess DL's effectiveness against phishing attempts by using the confusion matrix, F1-score, memory, accuracy, and precision. A confusion matrix gives the total number for each kind of prediction for each class. Accuracy, precision, recall, and F1-score are all determined using a confusion matrix. It is important to know these

methods well when looking at how phishing attack detection models work:

- **True Positive (TP):** The quantity of phishing websites that are appropriately classified.
- **False Negative (FN):** The number of websites that were identified as phishing, yet were really trustworthy.
- **False Positive (FP):** The quantity of websites that were found to be phishing, yet were actually legitimate.
- **True Negative (TN):** The quantity of reliable websites that are correctly classified.

Accuracy: ACC stands for the fraction of websites that are correctly divided into legitimate and phishing sites. The relationship can be written mathematically with Equation (2):

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \tag{2}$$

Precision: It explains what percentage of positive outcomes are associated with actual phishing attacks. This is shown using the following Equation (3):

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{3}$$

Recall: Recall considers the ratio of phishing attacks in comparison to all attack flows and quantifies the percentage of accurate real positive predictions. It sees its ability to list and remember good things in its past from recall. The relationship for elasticity can also be expressed by Equation (4):

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{4}$$

F1-score: This metric makes it possible to get a trustworthy view of how the model has performed. The metric is ready by computing the mean between precision and recall after reconciliation. It computes Equation (5):

$$\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{5}$$

When considering both, it can tell just how well and accurately the model predicts the chosen outcome.

4. Results and Discussion

The Phishing dataset is used to demonstrate the result of the suggested method. This work assesses how well a CNN model can detect phishing attacks, as shown in Table 2 and Table 3. All of the experiments took place in Python 3, TensorFlow and a Windows 10 computer with an Intel i7 CPU (four cores, 3.60 GHz) and 16 GB of RAM was utilized. Featuring a 94.92% accuracy, the proposed CNN showed it is very reliable in classifying data. With these results, the system successfully identified 96.32% of phishing attempts and only ignored 3.7% of them, providing high accuracy. The model's balanced and dependable performance in phishing threat identification is confirmed by the F1-score of 95.03%.

Table 2. Results of CNN model for Phishing dataset.

Performance Matrix	Convolutional Neural Networks (CNNs)
Accuracy	94.92
Precision	93.79
Recall	96.32
F1-score	95.03

Figure 4 accuracy curve shows that CNN's training accuracy steadily improves, reaching over 94%. After epoch 6, validation accuracy quickly increases after initially fluctuating, reaching training accuracy by epoch 10. This suggests strong generalization and efficient model operation.

The CNN model's training loss gradually drops, as the loss curve demonstrates, indicating effective learning shown in Figure 5. Although the validation loss fluctuates in the early epochs (peaking at epoch 2), it begins to decline consistently from epoch 6 and aligns closely with the training loss by epoch 10. This convergence

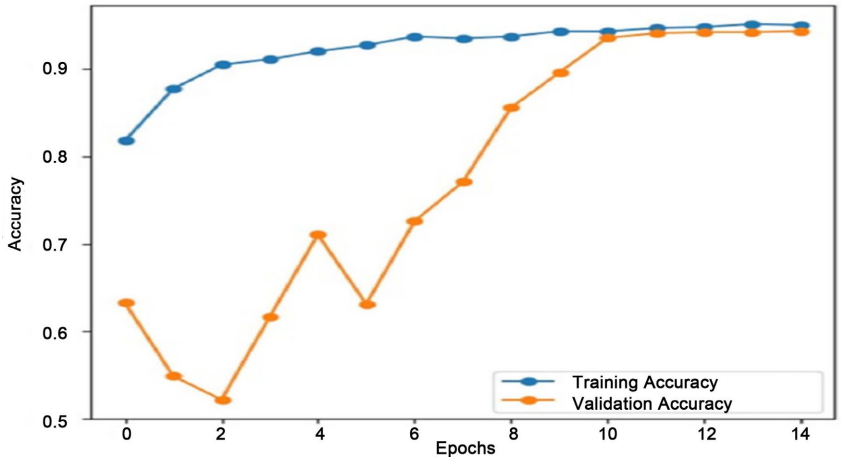

Figure 4. Accuracy curves for CNN model for Phishing dataset.

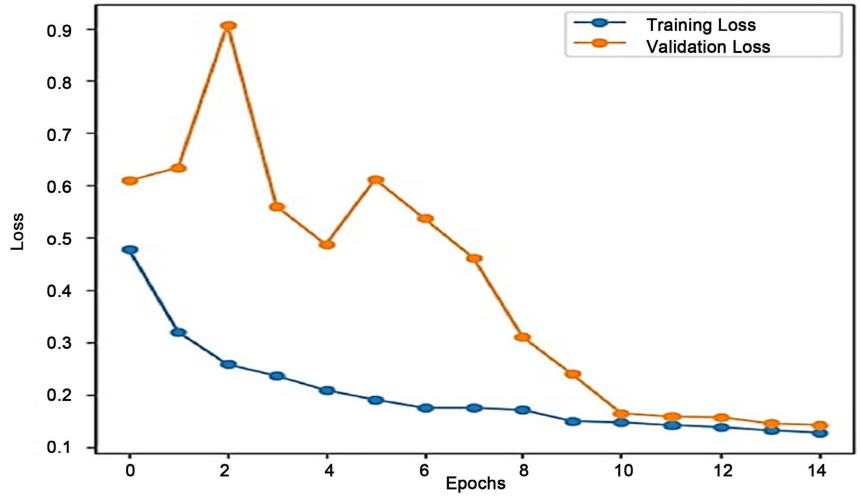

Figure 5. Loss curves for CNN model for Phishing dataset.

and stability in later epochs suggest reduced overfitting and strong generalization performance, highlighting the model's effectiveness after sufficient training.

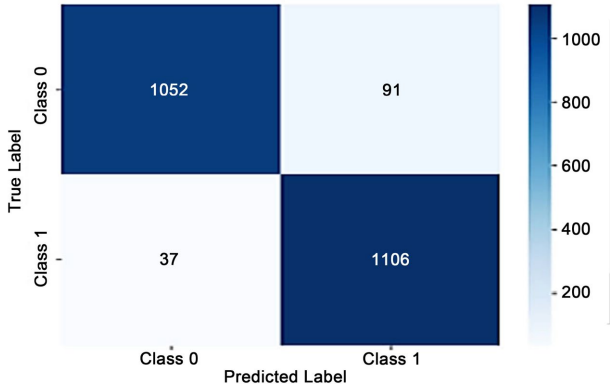

Figure 6. Confusion matrix for Phishing dataset.

A confusion matrix used to assess a binary classification model's performance is displayed in **Figure 6**. The matrix displays the number of false positives (1106), false negatives (37), true negatives (1052), and false positives (91). This demonstrates that 1106 Class 1 instances and 1052 Class 0 cases were accurately predicted by the model. It was incorrect to classify 37 Class 1 instances as Class 0 and Class 0 cases as Class 1. Although the model performs well overall, making many accurate predictions for both groups, there are still occasional misclassifications.

Table 3. Results of CNN model for PhishTank dataset

Performance Matrix	Convolutional Neural Networks (CNNs)
Accuracy	99.3
Precision	99.5
Recall	99.2
F1-score	99.34

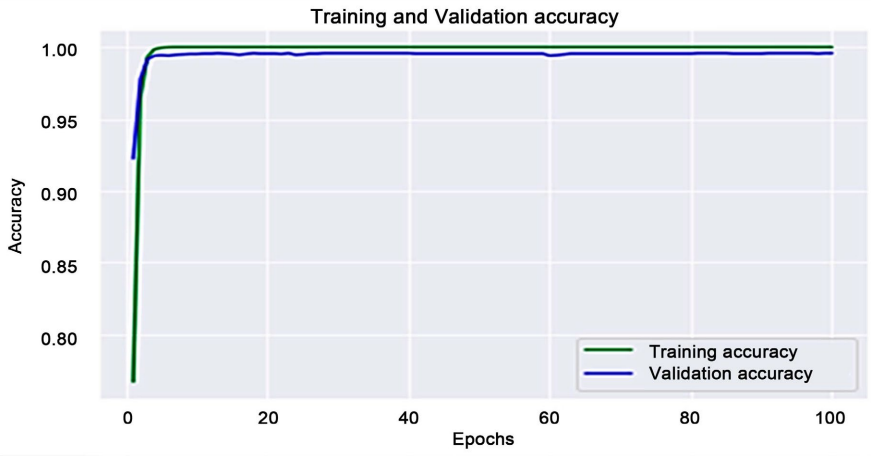

Figure 7. Training and validation accuracy for PhishTank dataset.

The model's accuracy throughout 100 epochs of training and validation is displayed in Figure 7. Both accuracy curves rapidly converge to values above 99% within the first few epochs, indicating swift and stable learning. Excellent generalization performance is indicated by the small difference between the training and validation curves, which shows no discernible overfitting during the training phase.

In Figure 8, the training and validation loss progression is displayed across 100 epochs. Within the first few epochs, the loss for both training and validation datasets drops dramatically before levelling off near zero. Strong learning behavior is ensured by the model's strong performance and little underfitting or overfitting, as further evidenced by the two curves' near alignment.

Comparison with Discussion

This comparative study presents an evaluation of classification models on two distinct datasets. The resulting performance metrics and key observations from these experiments are summarized below.

Figure 8. Training and validation loss for PhishTank dataset.

1) For Phishing Dataset

Here, using a Phishing dataset, the effectiveness of several models for stopping phishing assaults is assessed, as shown in Table 4, a comparison among Logistic Regression (LR) [20], XGBoost (XGB) [21], and a CNN model that has been presented shows that the CNN model is the most effective. With comparatively straightforward probabilistic assumptions, the XGBoost model demonstrated a strong baseline performance with an accuracy of 91.11%. With an accuracy of 93.97%, LR somewhat beat NB thanks to its superior capacity to represent linear connections. However, the CNN model surpassed both traditional ML approaches, achieving an accuracy of 94.92%. This improvement highlights the CNN's capability to automatically extract and learn complex patterns from the dataset, making it more adept at distinguishing between phishing and legitimate data. The results suggest that DL models like CNN offer a more powerful solution for phishing

attack prevention, especially when working with rich and complex feature representations.

Table 4. Comparative performance of proposed and baseline models based on Phishing attack using Phishing dataset.

Models	Accuracy
XGBOOST [21]	91.11
LR [20]	93.97
CNN	94.92

The CNN model can offer several important benefits in detecting phishing attacks. Complex and hierarchical features can be taken from the data automatically without anyone having to manually create them. CNNs do well at finding patterns in the layouts of phishing data, making them the appropriate choice for analyzing structured Phishing datasets. Besides, the model has the ability to generalize well, so the risk of overfitting stays low with noisy or high-dimensional data. Neural network technology allows it to learn complex relationships between features, which makes phishing classifiers more able to adapt as attacks change.

2) For PhishTank Dataset

In this evaluation, the performance of three classification models—Recurrent Neural Network (RNN) [22], Random Forest (RF) [23], and Convolutional Neural Network (CNN) was assessed for phishing detection, with results summarized in **Table 5**. The RNN model, known for its ability to model sequential dependencies, achieved a strong accuracy of 95.61%, effectively capturing the temporal structure in URL patterns. The RF model performed better than the RNN, with an accuracy of 97.14 percent, leveraging its ensemble learning mechanism to handle feature variability and improve generalization. However, with an astounding accuracy of 99.3%, the CNN model showed the most efficacy. This notable improvement highlights CNN's strength in automatically learning and extracting complex patterns from structured inputs such as URLs. The results demonstrate that, in contrast to conventional and sequence-based ML models, DL models, in particular, CNNs offer a very reliable and accurate method of phishing detection.

Table 5. Comparative performance of proposed and baseline models based on Phishing attack using PhishTank dataset.

Models	Accuracy
RNN [22]	95.61
Random Forest [23]	97.14
CNN	99.3

There are several advantages to using DL and sophisticated ML models for phishing detection. The use of DL and advanced ML models for phishing detection has

a number of benefits. In addition to their excellent accuracy, these models are flexible enough to adjust to the ever-changing landscape of phishing attempts. For instance, CNNs can automatically extract intricate spatial patterns from URL structures, removing the requirement for human feature extraction and greatly enhancing detection capabilities. RNNs are effective in capturing sequential dependencies within URLs, making them suitable for modeling temporal patterns in phishing behavior. Random Forests offer robustness through ensemble learning, handling diverse features and minimizing overfitting, while also maintaining interpretability. Collectively, these models enhance detection precision, reduce false positives, and enable scalable and real-time deployment, making them highly effective tools for proactive phishing threat prevention.

According to a comprehensive analysis of many DL and ML models for phishing detection, CNN outperforms the others. Because it can predict linear connections, LR marginally outperformed NB, although traditional models like NB and LR obtained acceptable accuracies of 93.48% and 93.97%, respectively. More advanced models like RNN and RF further improved detection capabilities, achieving accuracies of 95.61% and 97.14% respectively, by capturing sequential patterns and leveraging ensemble learning. However, across both evaluations, the CNN model consistently demonstrated the highest performance, achieving 94.47% in one study and an exceptional 99.3% in another, highlighting its ability to automatically extract deep, complex features from URL structures. These results clearly establish CNN as the most effective model among those evaluated for phishing URL detection.

5. Conclusion and Future Study

Attacks that use phishing are known to frequently expose data and cause monetary losses. As a crucial first step in defending against online dangers, one study suggested using a CNN model to identify phishing websites. This paper uses two benchmark datasets, Phishing and PhishTank, to assess a robust phishing detection framework based on CNN. The CNN model excelled on the PhishTank dataset with an exceptional accuracy of 99.3%, regularly outperforming more conventional ML techniques, including Logistic Regression, XGBoost, Random Forest, and Recurrent Neural Networks. The model's high F1-scores, stable training curves, and generalization capability, validated through 5-fold cross-validation, demonstrate its reliability in identifying phishing threats with minimal false positives and overfitting. Future work could explore integrating attention mechanisms and CNN-LSTM and Transformer-based architectures, which are examples of hybrid DL models that improve interpretability and performance. Additionally, applying transfer learning across multilingual Phishing datasets, incorporating real-time URL streaming, and implementing continual learning strategies will further extend the model's applicability to evolving phishing techniques in dynamic cybersecurity environments.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Tounsi, W. and Rais, H. (2018) A Survey on Technical Threat Intelligence in the Age of Sophisticated Cyber Attacks. *Computers & Security*, **72**, 212-233. https://doi.org/10.1016/j.cose.2017.09.001

[2] Alswailem, A., Alabdullah, B., Alrumayh, N. and Alsedrani, A. (2019) Detecting Phishing Websites Using Machine Learning. 2019 2*nd International Conference on Computer Applications & Information Security* (*ICCAIS*), Riyadh, 1-3 May 2019, 1-6. https://doi.org/10.1109/cais.2019.8769571

[3] Kolluri, V. (2016) A Pioneering Approach to Forensic Insights: Utilization AI for Cybersecurity Incident Investigations. *International Journal of Research and Analytical Reviews*, **3**, 919-922.

[4] Varshney, G., Misra, M. and Atrey, P.K. (2016) A Phish Detector Using Lightweight Search Features. *Computers & Security*, **62**, 213-228. https://doi.org/10.1016/j.cose.2016.08.003

[5] Ben-Asher, N. and Gonzalez, C. (2015) Effects of Cyber Security Knowledge on Attack Detection. *Computers in Human Behavior*, **48**, 51-61. https://doi.org/10.1016/j.chb.2015.01.039

[6] Parmar, J.D. and Patel, J.T. (2017) Anomaly Detection in Data Mining: A Review. *International Journal of Advanced Research in Computer Science and Software Engineering*, **7**, 32-40. https://doi.org/10.23956/ijarcsse/v7i4/0142

[7] Abu-Nimeh, S., Nappa, D., Wang, X. and Nair, S. (2007) A Comparison of Machine Learning Techniques for Phishing Detection. Proceedings of the Anti-Phishing Working Groups 2nd Annual eCrime Researchers Summit, Pittsburgh, 4-5 October 2007, 60-69. https://doi.org/10.1145/1299015.1299021

[8] Kolluri, V. (2015) A Comprehensive Analysis on Explainable and Ethical Machine: Demystifying Advances in Artificial Intelligence. *SSRN Electronic Journal*, **2**, a1-a5.

[9] Rege, M. and Mbah, R. (2018) Machine Learning for Cyber Defense and Attack. 7*th International Conference on Data Analytics*, Athens, 7–14, 18-22 November 2018, 73-78.

[10] Wei, B., Hamad, R.A., Yang, L., He, X., Wang, H., Gao, B., *et al.* (2019) A Deep-Learning-Driven Light-Weight Phishing Detection Sensor. *Sensors*, **19**, Article 4258. https://doi.org/10.3390/s19194258

[11] Noor, U., Anwar, Z., Amjad, T. and Choo, K.R. (2019) A Machine Learning-Based Fintech Cyber Threat Attribution Framework Using High-Level Indicators of Compromise. *Future Generation Computer Systems*, **96**, 227-242. https://doi.org/10.1016/j.future.2019.02.013

[12] Nathezhtha, T., Sangeetha, D. and Vaidehi, V. (2019) WC-PAD: Web Crawling Based Phishing Attack Detection. 2019 *International Carnahan Conference on Security Technology*, Chennai, 1-3 October 2019, 1-6. https://doi.org/10.1109/ccst.2019.8888416

[13] Arivudainambi, D., Varun Kumar, K.A., Sibi Chakkaravarthy, S. and Visu, P. (2019) *Computer Communications*, **147**, 50-57. https://doi.org/10.1016/j.comcom.2019.08.003

[14] Niu, W., Zhang, X., Yang, G., Ma, Z. and Zhuo, Z. (2017) Phishing Emails Detection Using CS-SVM. 2017 *IEEE International Symposium on Parallel and Distributed Processing with Applications and* 2017 *IEEE International Conference on Ubiquitous Computing and Communications* (*ISPA/IUCC*), Guangzhou, 12-15 December 2017, 1054-1059. https://doi.org/10.1109/ispa/iucc.2017.00160

[15] Li, J.H. and Wang, S.D. (2017) PhishBox: An Approach for Phishing Validation and

Detection. 2017 *IEEE* 15*th International Conference on Dependable, Autonomic and Secure Computing*, 2017 *IEEE* 15*th International Conference on Pervasive Intelligence and Computing*, 2017 *IEEE* 3*rd International Conference on Big Data Intelligence and Compu*, Orlando, 6-10 November 2017, 557-564.

[16] Salloum, S.A., Al-Emran, M., Monem, A.A. and Shaalan, K. (2018) Using Text Mining Techniques for Extracting Information from Research Articles. In: *Studies in Computational Intelligence*, Springer, 373-397. https://doi.org/10.1007/978-3-319-67056-0_18

[17] Rodríguez, P., Bautista, M.A., Gonzàlez, J. and Escalera, S. (2018) Beyond One-Hot Encoding: Lower Dimensional Target Embedding. *Image and Vision Computing*, **75**, 21-31. https://doi.org/10.1016/j.imavis.2018.04.004

[18] He, H., Zhang, W. and Zhang, S. (2018) A Novel Ensemble Method for Credit Scoring: Adaption of Different Imbalance Ratios. *Expert Systems with Applications*, **98**, 105-117. https://doi.org/10.1016/j.eswa.2018.01.012

[19] Kanter, J.M. and Veeramachaneni, K. (2015) Deep Feature Synthesis: Towards Automating Data Science Endeavors. 2015 *IEEE International Conference on Data Science and Advanced Analytics* (*DSAA*), Paris, 19-21 October 2015, 1-10. https://doi.org/10.1109/dsaa.2015.7344858

[20] Yazhmozhi, V.M. and Janet, B. (2019) Natural Language Processing and Machine Learning Based Phishing Website Detection System. 2019 *Third International Conference on I-SMAC* (*IoT in Social*, *Mobile*, *Analytics and Cloud*) (*I-SMAC*), Palladam, 12-14 December 2019, 336-340. https://doi.org/10.1109/i-smac47947.2019.9032492

[21] Yi, P., Guan, Y., Zou, F., Yao, Y., Wang, W. and Zhu, T. (2018) Web Phishing Detection Using a Deep Learning Framework. *Wireless Communications and Mobile Computing*, **2018**, Article ID: 4678746. https://onlinelibrary.wiley.com/doi/full/10.1155/2018/4678746

[22] Wang, W., Zhang, F., Luo, X. and Zhang, S. (2019) PDRCNN: Precise Phishing Detection with Recurrent Convolutional Neural Networks. *Security and Communication Networks*, **2019**, 1-15. https://doi.org/10.1155/2019/2595794

[23] Mahajan, R. and Siddavatam, I. (2018) Phishing Website Detection Using Machine Learning Algorithms. *International Journal of Computer Applications*, **181**, 45-47. https://doi.org/10.5120/ijca2018918026