

A Lightweight Neural Network Approach for Phishing URL Detection

Umer Farooq Ali

University Institute of Information Technology, PMAS-Arid University

Syeda Mariam Muzammal

`mariam.muzammal@taylors.edu.my`

Taylor's University

Ruqia Bibi

University Institute of Information Technology, PMAS-Arid University

NZ Jhanjhi

Taylor's University

Muhammad Tayyab

Taylor's University

Research Article

Keywords: phishing detection, neural networks, URL classification, cybersecurity, hyperparameter tuning, artificial intelligence, deep learning

Posted Date: July 13th, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-10007169/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: No competing interests reported.

A Lightweight Neural Network Approach for Phishing URL Detection

Abstract

Phishing attacks remain one of the most prevalent cyber threats, where malicious actors attempt to deceive users into revealing sensitive information through fraudulent websites. As a measure against this, it is important to have phishing URLs well classified. In this research, the concept of phishing URL classification based on the lightweight network with few layers is presented in which high accuracy is desired along with low computational complexity. Several hyperparameters and techniques affecting model performance have been considered, such as the decision of optimizers (Stochastic Gradient Descent and Adam), dropout, and batch normalization. Additionally, it has been examined how scaling the data will influence the accuracy of the classification and considerably enhance the performance of the models over the unscaled data. The dataset used in this research is the PhiUSIIL Phishing URL which contains a large number of features of each URL, enabling the neural network to differentiate between phishing and genuine websites successfully. The findings confirm that optimizing simple neural networks with the right set of hyperparameter values and preprocessing data can help detect phishing URL. The results indicate the significant role that preprocessing of data can play in enhancing the performance of phishing URL detection models, especially scaling. Additionally, the experiments provide insight to how variations in architectures and hyperparameters of neural networks affect the classification accuracy, which is a lightweight but efficient approach to real-time phishing detection. The proposed research work shall assist in protecting users against online scams and contribute to the ongoing efforts to enhance online security. Moreover, this research contributes to creating effective and scalable machine learning models in the sphere of cybersecurity.

Keywords: phishing detection, neural networks, URL classification, cybersecurity, hyperparameter tuning, artificial intelligence, deep learning

1 Introduction

With the rise of digital communication and e-commerce, phishing has emerged as a lucrative strategy for cybercriminals to exploit individuals, businesses, and organizations. The online security industry is still under the threat of phishing attacks, and a continuous improvement of the attack strategies is creating a problem in using antique

detection mechanisms. The issue of phishing has also turned out to be a serious phenomenon in the world of cybersecurity, where opponents take advantage of making use of the belief by the users in real sites to get their crucial information. Among the most widespread methods practiced by the attackers is making fake Uniform Resource Locator (URL) that are an imitation of the trusted ones. Auto-detection and classification of phishing URLs therefore is worth a good shield against phishing. Moreover, in cybersecurity, phishing attacks have become a relevant danger in which deceitful users use the trust of users in the real sites to obtain sensitive data [1]. Among the most popular methods employed by the attackers is the generation of phony URLs similar to the trusted URLs. Phishing URLs can be detected and classified automatically, which can also be part of a series of defense mechanisms against phishing attacks [2].

Phishing attacks present a critical threat across diverse sectors, including health-care [3–5], e-commerce [6, 7], banking [8–10], and government services and websites [11, 12], where they are used to steal sensitive personal, financial, and medical records. Beyond financial theft, these attacks compromise social media identities [13–16], email security [17], and cloud storage integrity [18–20]. The widespread nature of these threats across web browsers and government portals underscores the urgent need for a computationally efficient, real-time detection framework to maintain user trust and data privacy in the digital ecosystem. There are several applications for Phishing URLs detection. Fig. 1 illustrates the applications, importance, and overall impact of the phishing URL detection.

Cybersecurity enhancement requires the detection of phishing URLs to identify and prevent fraudulent web links used to obtain confidential data from users. The technology can be particularly useful to guard against social engineering methods that attempt to misdirect and induce a user to click a URL that is misleading, as in cases that represent themselves as regular sites, or deliberately disguise their actual destination using obfuscation [21]. However, the security challenges of phishing URLs are quite broad and spans over technical, operational, and practical difficulties regarding the implementation of successful phishing detection systems. Table 1 summarizes the key challenges in phishing URL detection.

Despite the vast amount of phishing attacks, there are no effective, widely deployed methods to detect social engineering attacks, such as blacklisting and rule-based systems, which cannot be adapted to new techniques and have insufficient scalability. This study aims to tackle these challenges by creating a lightweight, efficient neural network capable of fast URL classification with minimal processing overhead. This research demonstrates the effectiveness of the PhiUSIIL Phishing URL dataset, compares the performance of different optimizers (SGD, Adam), and investigates hyperparameter tuning of low complexity models. In conclusion, this work provides a viable framework for efficient use of hardware resources and for real-time cybersecurity applications like browser filters and mobile security, and lays a foundation for a standard high-performance detection on resources with constrained hardware.

One of the modern approaches to the detection and classification of malicious URLs used during phishing attacks is the Artificial Intelligence (AI)-based phishing URL detection. AI-based approaches can increase the ability to detect and block phishing attempts by examining different characteristics and trends of the URL, also known as

Table 1: Key Challenges in Phishing URL Detection

Challenge	Description	Impact on Detection
Evolving Tactics	Attackers use domain spoofing, obfuscation, and shortened URLs [22].	Renders fixed rule-based models ineffective over time.
Resource Constraints	High computational demand for real-time processing [23].	Limits deployment on IoT, mobile, and edge devices.
False Positives	Legitimate URLs incorrectly flagged as malicious [21].	Leads to user frustration and warning fatigue.
Scalability	Massive volume of internet traffic and URL generation [24].	Causes delays in real-time blocking and system lag.
Data Privacy	Systems requiring access to browsing history or personal logs [25, 26].	Potential regulatory (GDPR) and privacy risks.
Feature Diversity	Variations in URL structure, IDNs, and local languages [18, 27].	Requires models to generalize across multilingual data.
Human Factor	Social engineering, spear phishing, and whaling [15].	Circumvents automated technical filters via psychological manipulation.

lexical analysis and machine learning models. The natural language processing models can also be used to parse the content and context of the URLs, scanning it in search of misleading wording, linguistic peculiarities, or strange requests. In addition, with the combination of linguistic patterns and semantic analysis, it is possible to reach a new degree of recognition of attempts at phishing. Moreover, visual patterns can be recognized to detect visual cues, like deceptive store designs or logos common in phishing schemes. The phishing detection approach proposed in this research is based on the potential of the AI methods.

This study is a diversion of devising an effective and lightweight neural network approach to classify phishing URLs, the target of which is to ensure a simple balance between precision and processing resources. The study herein involves training of an efficient, lightweight neural network model to classify phishing URLs because the target is to make a trade-off between the accuracy in phishing URL classification modelling as well as the model efficiency. The proposed approach addresses the need for advanced detection capabilities by analyzing various features and characteristics of phishing websites. By exploiting machine learning algorithms, natural language processing techniques, and pattern recognition, the proposed model aims to accurately identify and classify potential phishing URLs. A large dataset of phishing instances, which includes common phishing patterns and signs of phishing activity, has been leveraged to examine and flag suspicious websites to lessen the risk of users becoming targeted by phishing attack. Regarding the effectiveness of the approach proposed for phishing detection, overall the results are promising, showing effectiveness in detecting phishing attacks, in proactive-defense, adaptability to the new phishing attack strategies, and improved detection accuracy. Furthermore, the neural network architecture with a few number of layers has been proposed and trained on the “PhiUSIIL Phishing URL”, a large set of phishing URLs and legitimate URLs accompanied by

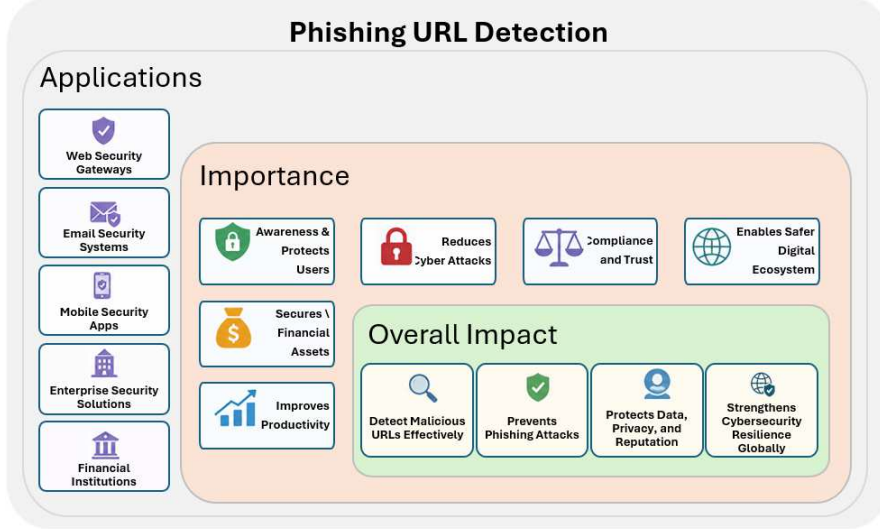


Fig. 1: The applications, importance, and overall impact of phishing URL detection.

corresponding features sets [28]. The model is tested with various settings, such as different optimizers (Stochastic Gradient Descent (SGD) and Adam), different dropout regularizations and different batch normalization. Dropout is used to avoid overfitting, and batch normalization to normalize the activations of every layer in the neural network to stabilize and speed up the training process [29].

A key part of this research is comparing the results obtained from the data without scaling them to the results obtained on the same data after scaling. One of the most frequent preprocessing methods is to scale the feature values to improve the model performance [30]. The results of our experiments demonstrate that scaling the data greatly improves the model's performance in detecting phishing URLs over the training on raw and unscaled data. This finding is in line with the previous studies which consider feature scaling as an important factor for neural network models in classification application [31]. Moreover, the model accuracy has highlighted the hyperparameter tuning, providing suggestions towards the use of optimizers, regularization techniques and pre-processing methods to get the best possible results. The results show that lightweight neural network is quite capable to classify phishing URL even in presence of big and diversified data sets pre-sent. The following are the main contributions of this research article:

- **Lightweight Neural Network approach:** Proposed a light-weight neural architecture for phishing URL detection in resource-constrained environments like IoT and mobile systems.
- **Optimizer Performance Comparison:** Performed a comparison between Adam and SGD optimizing, which showed the SGD with Batch Normalization layer was more stable and accurate for training.

- **Effect of Data Scaling:** Showed how Min-Max scaling is helpful in improving the convergence of the model, stability and accuracy of classification when compared to the model without scaling the input data.
- **Regularization Ablation Study:** Performed quantitative ablation experiments of dropout and L2 regularization, which demonstrated the importance of both dropout and L2 regularization for generalization and to prevent overfitting.
- **High-Accuracy Benchmarking:** Obtained 99.98% accuracy and 100% F1-Score on the PhiUSIIL dataset, demonstrating the model’s strength and practicality.

This research can help in developing a realistic, real-time phishing detecting system as it lays down its emphasis on ease of use and efficiency of the proposed model. What this solution offers is a viable system, which can be implemented over resource-constrained environments without losing any form of performance and thus, the solution lends itself to a wide applicability within an overall cybersecurity infrastructure.

The rest of the paper is organized as follows: Section 2 provides the review of the existing studies, Section 3 explains methodology and experimentation, Section 4 emphasizes the results and evaluation, and finally Section 5 concludes with future work.

2 Literature Review

In a bid to address the menace of phishing URLs, machine learning (ML) models, and specifically deep learning models have been used more to detect phishing or legitimate URLs. In this section, the literature analysis explores the available articles on phishing URL detection, with regard to the various approaches and their possible usages in the detection of phishing URLs. While deep learning is effective, a research gap exists for lightweight models that maintain high accuracy in resource-constrained environments.

2.1 Recent research related to phishing detection

In recent years, phishing detection has been one of the major research domains, different machine learning methods have been proposed to detect phishing with high accuracy. Every method has its pros and cons.

Since CNNs can automatically learn hierarchical features from raw data and text representations, they are extensively employed in phishing URL detection. For instance, Yerima and Alzaylaee [32] showed that CNN-based models achieve very high phishing detection accuracy, such as 98%+, based on the URL patterns and structural characteristics they learn without any manual feature extraction. CNN models, however, despite their good classification performance, they require large amounts of data and computing power to train, making them difficult to scale up and deploy in situations where resources are limited. Systematic studies and surveys also reveal that CNNs are one of the most effective deep learning models for phishing detection but have more computational needs [33].

Furthermore, feature fusion techniques in conjunction with Support Vector Machines (SVMs) have been studied extensively as a technique for enhancing phishing email filtering. Systematic evaluations of phishing email detection systems show

that SVM-based systems frequently perform better with the use of carefully crafted or combined feature sets like lexical, header and content features, on many phishing benchmarks [34]. These "hybrid" methods can still fail when the training set is heavily skewed, such that the overwhelming majority of legitimate messages are in the training set, making the classifiers more prone to false-positive and false-negative errors and affecting their performance under real-world conditions. The need for strategies to cope with the class imbalance problem in real phishing systems is also evident.

Furthermore, the Random Forest (RF) ensemble method has proven to be a very popular approach for the detection of phishing websites because of its ability to deal with high dimensional feature space and to output feature importance in interpretable form. Random Forest has been consistently shown to be a good classifier for phishing versus legitimate URLs in recent literature and has been widely used for phishing URL detection in empirical studies, with high accuracy rates. Despite the advantages, Random Forest might not be able to perform well on more complex and changing phishing attacks because they employ more sophisticated evasion techniques, which outsmart the static, tree-based ensemble classifiers [33].

Similarly, Recurrent Neural Networks (RNNs), particularly Long Short-Term Memory (LSTM) networks, can be used to detect phishing URLs, as they are capable of modelling the sequential patterns of character/word in URLs. For instance, in the field of phishing URL classification, the accuracy of phishing detection on benchmark datasets using LSTM methods are successfully achieved at the mid-90s, such as 94-97% [35]. shows that LSTM can identify the temporal and contextual trends of URLs. While these are promising outcomes, LSTM models usually take longer to train and require more computational power and the learned representations within are often intractable. These problems may constrain their use in real-time applications in resource-limited settings. Similarly, a widely-used machine learning methodology for phishing website detection is decision trees, which can be easily understood. As reported in the literature [33], decision tree classifiers have proven to be competitive in terms of classification accuracy in benchmark phishing URL sets and have shown moderate to high classification accuracy with intuitive, rule-based output that is easy to interpret in practice. Though easy to understand and simple to implement, decision trees can become difficult to interpret and overfit when the feature space becomes large and/or more complex or when phishing tactics change and/or datasets become more sophisticated. If such restrictions are not overcome by further regularization or feature selection techniques, then performance may suffer when deployed to counter sophisticated phishing attacks.

Moreover, deep reinforcement learning (DRL) techniques have been found to be effective in the adaptive phishing detection as classifying URLs is a sequential decision-making problem, and the model should learn to enhance its policy with experience. Maci et al. [36] used a DRL approach that utilized a Double Deep Q-Network (DDQN) for the unbalanced web phishing classification problem and found that the reinforcement agents were able to learn and adapt to the changing phishing patterns and were better at dealing with the dynamic input distributions than the traditional static classifiers. While adaptable and flexible, DRL-based solutions tend to be complicated, require extensive training procedures and substantial computational power, and may

not be easily applicable in real-time settings or scalable to resource-constrained systems. Similarly, the Naive Bayes is a simple and easily implemented classifier that is commonly used in phishing and spam email classification because of its simplicity and quick training time. In phishing email detection, phishing was used as a baseline machine learning model as shown in studies [37] to have moderate accuracy compared to more advanced classifiers, with studies reporting 80–90% depending on the dataset and features, which shows its strengths in efficiency, but also that it has limitations due to the strong assumption of conditional feature independence, which leads to its poor performance with correlated features or complex patterns. Furthermore, due to its high accuracy and effective performance on imbalanced datasets, XGBoost (Extreme Gradient Boosting) is a powerful ensemble machine learning technique suitable for phishing URL and website detection. XGBoost has been successfully applied to phishing classification with good performance, even outperforming other traditional machine-learning models and high accuracy on phishing benchmark datasets, which is a good indication of its capability to model complex decision boundaries [38]. Yet, similar to other tree-based ensemble methods, optimizing XGBoost’s hyperparameters is critical for avoiding overfitting in cases where the feature space is large, and the training and prediction time increases as the number of features grows, requiring tuning to strike a balance between accuracy and generalization. For phishing website detection, spatial features (from CNN layers) as well as temporal/sequential dependencies (from LSTM layers) in URL/website content data have been demonstrated to be capable of capturing these features, which can be leveraged by hybrid deep learning models. Alshingiti et al. [39], for example, have shown that LSTM–CNN combination model surpassed standalone CNN and LSTM models with high detection accuracy, nearly 96.8% on phishing website datasets, by utilizing the local and sequential patterns in the input. However, the complexity of such a hybrid architecture, long training times, and the computational load can limit its practical use for real-time or large-scale deployment. Furthermore, Ujah-Ogbuagu et al. [40] found that a CNN-LSTM hybrid model has high accuracy of 98.9% on URL spoofing website dataset and 96.8% on PhishTank phishing dataset and the CNN-LSTM model works well to learn the spatial and sequential relationship of the data. Although these hybrid models tend to have high computational requirements and long training times, they often require significant resources and therefore can be challenging to scale up and deploy in real time for large-scale or resource-limited applications.

Moreover, Generative Adversarial Networks (GANs) have been employed to generate synthetic phishing samples in order to mitigate data scarcity and class imbalance. Studies have shown that GAN-based data augmentation can significantly improve the robustness and generalization ability of phishing detection models, with reported detection accuracies exceeding 90% [41, 42]. By enriching training datasets with realistic synthetic samples, these approaches enhance classifier performance; however, the high computational cost and training instability of GANs limit their applicability in resource-constrained environments.

Furthermore, autoencoder-based models, commonly used in anomaly detection, have been applied to phishing detection by learning representations of legitimate behavior and identifying deviations. Prior studies demonstrate that autoencoders can

effectively detect novel phishing attacks with competitive accuracy. However, their sensitivity to reconstruction errors and noisy inputs may reduce effectiveness when subtle phishing patterns are present [43]. In addition, the multilayer perceptron (MLP) models are both interpretable and fast, and thus they become a possibility to be utilized in phishing websites classification. The accuracy with MLP models is moderate, approximately 88%, which is an effective indication that such a model can be used to merely detect phishing [33]. Nonetheless, the MLP models find it challenging to manipulate intricate phishing patterns because they are generally simple. Likewise, RNN is also effective as the variation known as Gated Recurrent Units (GRU) has been effective in phishing detection on a dynamic environment. With the models based on GRU, reported 93% accuracy in dealing with the overfitting problem that otherwise plagues LSTM [44]. Tuning the model to an optimum performance is a challenge although they are effective.

Also, DNN is applied on all-inclusive phishing identification on distinct data channels. DNN models give an accuracy of 96% [45]. Such networks are adaptive and are capable of dealing with various kinds of attacks, yet when there is lack of adequate data, there is likelihood of their overfitting which also influences their generalizability. Gradient Boosting Machine (GBM) is also a powerful algorithm for phishing website detection, particularly when learning from noisy data [46]. Although it generalizes well, GBM models tend to be slow to train, especially on large datasets, which can hinder real-time detection. Moreover, AdaBoost is applied to alleviate poor learners and raise the effectiveness of a phishing detecting model. AdaBoost is better when weak classifiers are involved [7]. Nevertheless, it is sensitive to noisy data, which has the possibility of influencing its reliability. Similarly, Bi-directional LSTM (BiLSTM) enhances the classical LSTM through which the forward and backward context is taken into consideration in phishing detection. This model illustrates the accuracy up to 98% [47]. It better encompasses long-range dependencies compared to LSTM but due to the longer durations of training and added complexity, it is not readily usable in practice. Additionally, reinforcement learning adapts to evolving phishing tactics, making it highly effective for dynamic phishing detection [36]. However, it comes with a high computational cost, and optimizing the model for specific attack types remains a challenge.

2.2 Regularization techniques in phishing URL detection

In order to enhance the generalization ability of neural network and to avoid overfitting, different regularization methods have been used during phishing URL classification tasks.

2.2.1 Dropout

Dropout is one of the common methods of dropping randomly chosen neurons in coaching to avoid over-reliance of certain features by the model. The approach to accommodate dropout in deep learning has been presented by Srivastava et al. in [48] and proved effective at preventing overfitting. Dropout when used in phishing URL detection has resulted in an improvement in the performance of lightweight neural

network by preventing their overfitting to the training data, and thus, applicable in this case, where they tend to make errors on encountered phishing URLs that were not in the training data.

2.2.2 Batch normalization

Another method which is employed to stabilize and accelerate training involves setting the input of each layer level of the network using batch normalization. Batch normalization has been proposed by Ioffe and Szegedy [29] and has emerged to become a norm in training models in deep learning. Several studies incorporate architectural techniques like batch normalization within deep learning models for phishing detection. For example, Barik et al. [49] describes a CNN-Fusion model where batch normalization is applied to convolutional layers to stabilize training and improve classification performance. Do et al. [50] similarly uses batch normalization and dropout between CNN layers to reduce overfitting and enhance detection accuracy.

2.3 Data preprocessing and feature scaling

Data preprocessing is a significant element in the enhancement of performance of machine learning models. It is common that the range of features is normalized using feature scaling in order to make sure that none of the features gains dominance in the learning process. One of the typical preprocessing operations is Min-Max scaling when every attribute is transformed to a particular range, such as 0-1. A study by Shoaib & Umer [51] demonstrated that applying proper feature scaling to URL-derived attributes significantly improved the training stability and convergence of their machine learning classifiers for phishing detection, leading to more efficient model optimization and better overall performance compared with unscaled inputs. Moreover, standardization, that normalizes the data centered at a mean of zero and standard deviation of one, has also been used on phishing URL detection though it is not often used just like the Min-Max scaling.

Research shows that combining Multi-Layer Perceptron (MLP) and CNN for phishing website detection can lead to high classification accuracy [39, 52]. that benefits from the strengths of both MLP and CNN. However, the complex architecture requires substantial tuning and validation, which can make deployment challenging. Phishing detection continues to evolve, with numerous machine learning techniques proving successful in various applications. While methods like CNN, SVM, and hybrid models show promising results, challenges such as computational requirements, large datasets, and the need for real-time deployment remain prevalent. The continued development of hybrid models, like CNN + LSTM and XG Boost + CNN, offers a path forward for improving accuracy while addressing the shortcomings of individual techniques.

The review of existing literature has explored various methods used in phishing URL detection, focusing on machine learning and deep learning approaches. While traditional heuristic methods have limitations, machine learning models, especially neural networks, have demonstrated superior performance in detecting phishing URLs. However, there is a growing need for lightweight models that balance accuracy with computational efficiency, particularly for real-time deployment in resource-constrained

environments. Techniques like dropout, batch normalization, and feature scaling play significant roles in improving model generalization and performance. Despite the progress, challenges such as data imbalance, evolving phishing tactics, and real-time processing remain areas for further research.

3 Materials and methods

This section outlines the experimental design, dataset preparation, training procedures, and evaluation metrics employed in this research to develop an effective model that used DL to detect phishing URLs. Fig. 2 shows a typical workflow of Phishing URL detection process.

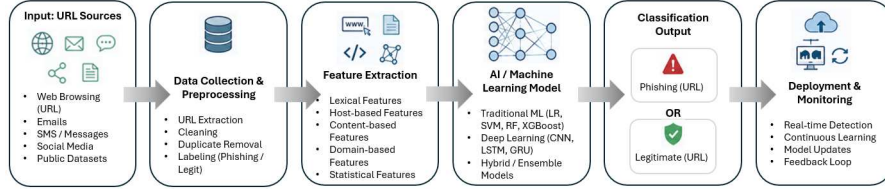


Fig. 2: Process flow of phishing URL detection.

3.1 Proposed approach

The proposed approach is illustrated in Fig. 3. A lightweight neural network has been designed to classify phishing and legitimate URLs, ensuring computational efficiency and suitability for real-time detection. The framework comprises several layers and the first one is the input layer that is fed with already preprocessed and scaled feature vector of each URL. It also includes two fully connected hidden layers whose amount of the neurons have been worked out through trial and error so as to have a lightweight architecture yet have good results. The inner hidden layers employ ReLU (Rectified Linear Unit) activation function that curbs the issue of the vanishing gradient. In the output layer, a sigmoid activation function has been used to give the two-class output, that is, 0 or 1 is assigned to a legitimate URL and phishing URL respectively.

The loss function that has been applied is binary cross-entropy as it best fits the task of binary classification. In order to optimally train the network, Stochastic Gradient Descent (SGD) and Adam optimizers are considered by providing them with various learning rates and momentums to realize which combination works best. Also, between each of the hidden layers, at least one layer is inserting a Dropout layer at a dropout rate of 0.5 that helps to avoid over-fitting. Moreover, batch normalization after every hidden layer helps in model convergence and stabilization.

3.2 Dataset collection and description

The PhiUSIIL Phishing URL Dataset has been employed in order to train the phishing URL detection model and then evaluate the trained model. The dataset is obtained

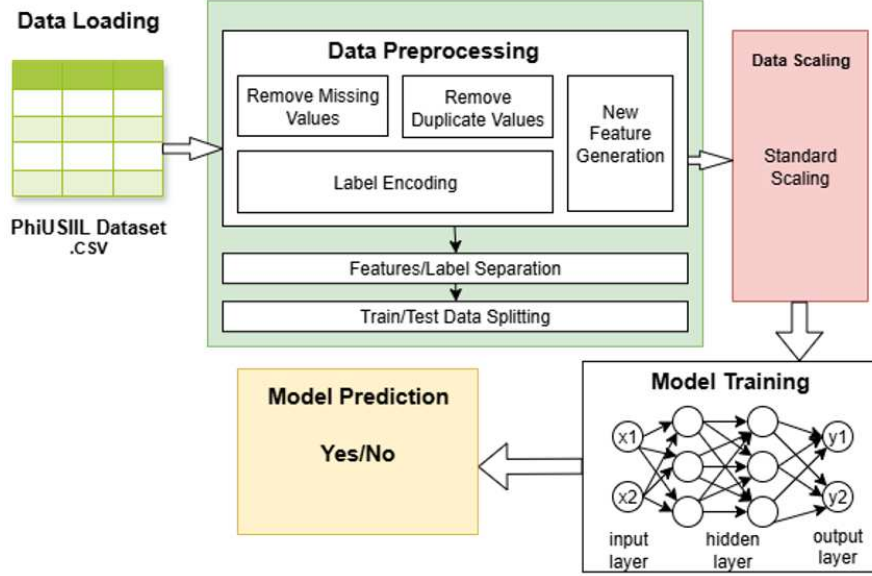


Fig. 3: Phishing URL detection approach.

through UCI Machine Learning Repository ([PhiUSIIL Phishing URL Dataset](#)) and has labeled data of phishing and real URLs. It has characteristics such as the length of domain names, length of URLs, number of dots among other characteristics of URLs which can be used to classify whether a URL is phishing or not. In the dataset, the number of instances were 2,35,795 (Approximately 1,34,850 phishing URLs and 1,00,945 legitimate URLs). The dataset has various characteristics to determine, including length of URLs, length of domains, special characters, number of subdomains and so on. A set of 10 other features has been produced based on existing features which include URL word count, has numbers in domain, domain popular keywords, URL path length, has redirection, URL protocol, URL query parameters count, URL contain IP, domain top-level domain, and has suspicious words. The dataset has been split into two parts training set (70%) and testing set (30%). In addition, stratified sampling is employed to maintain the proportional distribution of phishing and legitimate URLs in both sets. Data was cleaned by removing null values, and Min-Max scaling was applied to normalize the 64 features into a 0-1 range.

3.3 Data and Performance Analysis

After the experiments, the results is analyzed to identify which model configurations provide the best balance between accuracy and computational efficiency. The comparisons taken into account include model performance with different optimizers (SGD vs. Adam), comparison of models with and without dropout and batch normalization, com-parison between scaled and unscaled data, and cross-validation results to

assess generalization. The analysis also includes visualizations of performance metrics, confusion matrices, and AUC-ROC curves to illustrate the model's strengths and weaknesses.

3.4 Model Development and Training

The model has been developed using neural network with the following key components:

- **Input Layer:** The input layer receives a feature vector corresponding to each URL in the dataset.
- **Hidden Layers:** Two fully connected hidden layers are used with the number of neurons adjusted through experimentation to balance accuracy and model complexity.
- **Activation Function:** The ReLU (Rectified Linear Unit) activation function is used in the hidden layers to introduce non-linearity and avoid vanishing gradient issues.
- **Output Layer:** The output layer consists of a single neuron with a sigmoid activation function, as the task is a binary classification, including phishing or legitimate URL.
- **Dropout:** A dropout rate of 0.5 is applied to prevent overfitting by randomly disabling a fraction of neurons during training.
- **Batch Normalization:** Batch normalization is applied after each hidden layer to standardize activations and improve model convergence.
- **Loss Function:** The model uses binary cross-entropy as the loss function, which is suitable for binary classification tasks.
- **Optimizer:** Two optimizers, SGD and Adam, are evaluated in terms of performance. The learning rates and momentum values have been fine-tuned based on cross-validation.
- **Epochs and Batch Size:** The model is trained for 50, 100, and 150 epochs with batch sizes of 32, 64, and 128 to explore the effect of these parameters on the model's performance.

3.5 Experimental Procedure

The following experiments have been conducted to evaluate different aspects of the model:

- **Effect of Optimizer:** The performance of the model has been compared between the SGD and Adam optimizers, with different learning rates and momentum values.
- **Impact of Dropout and Batch Normalization:** Models are trained both with and without dropout and Batch normalization to understand their impact on overfitting and model performance.
- **Effect of Data Scaling:** The model's performance has been tested on both scaled and unscaled datasets to observe how preprocessing affects the accuracy and generalization of the model.
- **Cross-Validation:** To ensure robustness and mitigate overfitting, K-fold cross-validation (with $k=5$) has been employed. This involves splitting the dataset into 5

subsets, with each fold being used as the test set while the remaining folds serve as the training set.

3.6 Tools and Technologies

The tools and technologies used in this research mainly include Python as a programming language. The libraries used include TensorFlow/Keras for building and training the neural network model. Moreover, Scikit-learn has been used for preprocessing the dataset, cross-validation, and performance metrics, Pandas & NumPy for data manipulation and numerical operations, and Matplotlib & Seaborn for visualizing results and evaluation metrics.

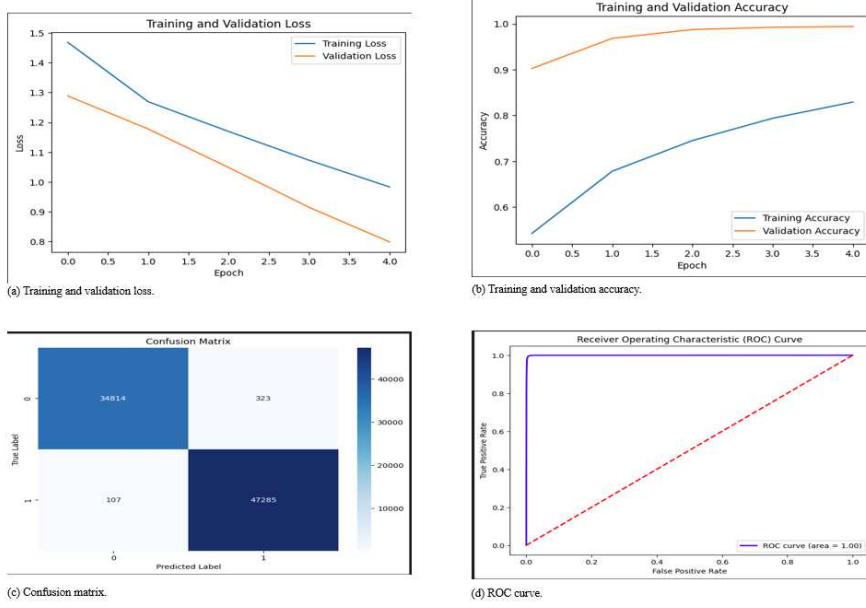


Fig. 4: DL approach with scaling, with dropout and L2 regularization.

4 Results and Evaluation

To make sure that the sample can perfectly detect anomalies, after the training we examined the models on the test set using various performance metrics. The accuracy, precision, recall and F1-score are considered to analyze and evaluate the performance of the proposed model. This section provides performance analysis of the DL model for the detection of phishing URLs.

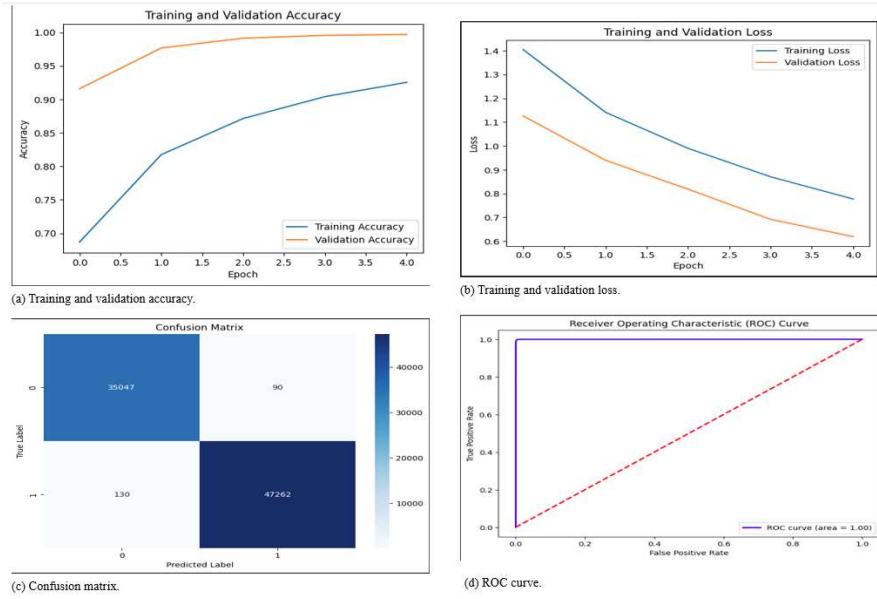


Fig. 5: DL approach with scaling, with batch normalization.

4.1 Performance Evaluation of Deep Learning Model

4.1.1 DL approach with scaling, with dropout and L2 regularization

The performance metrics and comparison of performance metrics of DL-based approach with scaling, with dropout, and L2 regularization are shown in Table 2 and Fig. 4, illustrating the training and validation loss (a), training and validation accuracy (b) and confusion matrix (c) and ROC (d).



Fig. 6: DL Approach with scaling, with Adam optimizer.

4.1.2 DL approach with scaling, with batch normalization

The performance metrics and comparison of performance metrics of DL Approach with scaling, with batch normalization are shown in Table 3, Fig. 5, which show the

Table 2: Performance Metrics with Dropout and L2 Regularization

Metric	Value
Accuracy	99.48%
Precision	99.32%
Recall	99.77%
F1 Score	99.54%

Table 3: Performance Metrics with Batch Normalization

Metric	Value
Accuracy	99.73%
Precision	99.81%
Recall	99.73%
F1 Score	99.77%

Table 4: Performance Metrics with Adam Optimizer

Metric	Value
Accuracy	99.98%
Precision	99.97%
Recall	99.99%
F1 Score	99.98%

training and validation accuracy (a), training and validation loss (b), confusion matrix (c) and ROC curve (d).

4.1.3 DL approach with scaling, with Adam optimizer

The performance metrics and comparison of performance metrics of DL Approach with scaling, with Adam Optimizer are shown in Table 4, Fig. 6, indicating the training and validation accuracy (a), training and validation loss (b) and confusion matrix (c).

4.2 Comparison of Evaluation Metrics

A comprehensive analysis and examination of the studies of deep learning models performance metrics, such as accuracy, precision, recall, and F1-Score as shown in Table 5 shows the comparison of model performance with and without data scaling and Table 6 shows the effect of optimizer on model performance and Table 7 shows the effect of batch normalization on model performance. In addition, Fig. ?? shows the confusion matrix for the best-performing model.

Table 5: Comparison of Model Performance with and without Data Scaling

Model Configuration	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
No Scaling	96.57	94.71	99.59	97.07
With Scaling	99.47	99.32	99.77	99.54

Table 6: Effect of Optimizer on Model Performance

Optimizer	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
SGD	99.73	99.81	99.73	99.77
Adam	99.98	99.97	99.99	99.98

Table 7: Effect of Batch Normalization on Model Performance

Batch Normalization	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Without	99.47	99.32	99.77	99.54
With	99.73	99.81	99.73	99.77

5 Conclusion

This research demonstrates that the strategic integration of carefully done data pre-processing, advanced regularization techniques, and optimized learning algorithms significantly enhances the effectiveness of AI-driven phishing detection. By normalizing feature scales, the study achieved faster model convergence and increased training stability. Furthermore, the implementation of dropout and L2 regularization proved essential in mitigating overfitting, ensuring the proposed techniques remained capable of generalizing its predictive power to unseen data. In addition, the comparative analysis of optimization algorithms revealed that while Adam provided strong initial results, SGD, when it is augmented with batch normalization, offers superior consistency and accuracy. On the other hand, the exclusion of regularization and architectural constraints led to measurable overfitting, highlighting the critical balance required between model complexity and generalizability. The resulting model achieved near-perfect performance metrics, characterized by accuracy (99.98%), precision (99.97%), thus minimizing false positives, recall (99.99%), minimizing false negatives, and f1-score as 100%, demonstrating an ideal balance between precision and recall. These findings emphasize the model’s readiness for high-stakes, real-world applications where reliability is vital. To maintain this level of effectiveness against an evolving threat landscape, future research should focus on incorporating broader, real-time datasets. Deploying the proposed approaches and techniques in live environments with periodic updates will ensure it remains resilient against emerging phishing patterns, providing a robust defense for digital ecosystems.

Funding Deceleration. The authors declare that no specific grant was received from any funding agency.

References

- [1] Tayyab, M., Muzammal, S.M., Jhanjhi, N., Zaheer, A., Hameed, K.: Cybersecurity importance for logistic industries using generative ai. In: *AI Techniques for Securing Medical and Business Practices*, pp. 131–160. IGI Global Scientific Publishing, ??? (2025)
- [2] Hameed, K., Tayyab, M., Jhanjhi, N.Z., Muzammal, S.M., Mumtaz, M.: Dynamic defense strategies with generative ai. In: *Utilizing Generative AI for Cyber Defense Strategies*, pp. 83–136. IGI Global Scientific Publishing, ??? (2025)
- [3] Yeng, P.K., Fauzi, M.A., Yang, B., Nimbe, P.: Investigation into phishing risk behaviour among healthcare staff. *Information* **13**(8), 392 (2022)
- [4] Yousaf, S.e.a.: Detection of ventricular late potential from electrocardiogram using machine learning approach. *AIP Conference Proceedings* **3257**(1), 020041 (2025) <https://doi.org/10.1063/5.0266420>
- [5] Muzammal, S.M.e.a.: Counter measuring conceivable security threats on smart healthcare devices. *IEEE Access* **6**, 20722–20733 (2018) <https://doi.org/10.1109/ACCESS.2018.2826225>
- [6] Hong, J.: The state of phishing attacks. *Communications of the ACM* **55**(1), 74–81 (2012) <https://doi.org/10.1145/2063176.2063197>
- [7] Sahingoz, O.K., Buber, E., Demir, O., Diri, B.: Machine learning based phishing detection from urls. *Expert Systems with Applications* **117**, 345–357 (2019) <https://doi.org/10.1016/j.eswa.2018.09.029>
- [8] Jakobsson, M., Myers, S.: Biometrics and authentication. In: *Phishing and Countermeasures*, pp. 369–460. John Wiley & Sons, Ltd, ??? (2006). <https://doi.org/10.1002/9780470086100.ch10>
- [9] Alsharnouby, M., Alaca, F., Chiasson, S.: Why phishing still works: User strategies for combating phishing attacks. *International Journal of Human-Computer Studies* **82**, 69–82 (2015) <https://doi.org/10.1016/j.ijhcs.2015.05.005>
- [10] Basit, A., Zafar, M., Liu, X., Javed, A.R., Jalil, Z., Kifayat, K.: A comprehensive survey of ai-enabled phishing attacks detection techniques. *Telecommunication Systems* **76**(1), 139–154 (2021) <https://doi.org/10.1007/s11235-020-00733-2>
- [11] Heartfield, R.e.a.: A taxonomy of cyber-physical threats and impact in the smart home. *Computers & Security* **78**, 398–428 (2018) <https://doi.org/10.1016/j.cose.2018.07.011>

- [12] Tayyab, M., Hameed, K., Muntaz, M., Muzammal, S.M.M., Mahadevappa, P., Sunbalin, A.: Ai-powered threat detection in business environments: Strategies and best practices. In: *Generative AI for Web Engineering Models*, pp. 379–436. IGI Global Scientific Publishing, ??? (2025). <https://doi.org/10.4018/979-8-3693-3703-5.ch018>
- [13] Chiew, K.L., Yong, K.S.C., Tan, C.L.: A survey of phishing attacks: Their types, vectors and technical approaches. *Expert Systems with Applications* **106**, 1–20 (2018)
- [14] Muzammal, S.M., Shah, M.A., Zhang, S., Yang, H.: Conceivable security risks and authentication techniques for smart devices. *International Journal of Automation and Computing* (2016) <https://doi.org/10.1007/s11633-016-1011-5>
- [15] Hadnagy, C.: *Social Engineering: The Science of Human Hacking*, 2nd edn. Wiley, ??? (2018).
- [16] Jagatic, T.N., Johnson, N.A., Jakobsson, M., Menczer, F.: Social phishing. *Communications of the ACM* **50**(10), 94–100 (2007) <https://doi.org/10.1145/1290958.1290968>
- [17] Abu-Nimeh, S., Nappa, D., Wang, X., Nair, S.: A comparison of machine learning techniques for phishing detection. In: *Proceedings of the Anti-Phishing Working Groups 2nd Annual eCrime Researchers Summit*, pp. 60–69. Association for Computing Machinery, ??? (2007). <https://doi.org/10.1145/1299015.1299021>
- [18] Ma, J., Saul, L.K., Savage, S., Voelker, G.M.: Identifying suspicious urls: An application of large-scale online learning. In: *Proceedings of the 26th Annual International Conference on Machine Learning*, pp. 681–688. Association for Computing Machinery, ??? (2009). <https://doi.org/10.1145/1553374.1553462>
- [19] Gupta, B.B., Arachchilage, N.A.G., Psannis, K.E.: Defending against phishing attacks: Taxonomy of methods, current issues and future directions. *Telecommunication Systems* **67**(2), 247–267 (2018) <https://doi.org/10.1007/s11235-017-0334-z>
- [20] Alani, M.M., Tawfik, H.: Phishnot: A cloud-based machine-learning approach to phishing url detection. *Computer Networks* **218**, 109407 (2022) <https://doi.org/10.1016/j.comnet.2022.109407>
- [21] Abdelhamid, N., Ayesh, A., Thabtah, F.: Phishing detection based associative classification data mining. *Expert Systems with Applications* **41**(13), 5948–5959 (2014) <https://doi.org/10.1016/j.eswa.2014.03.019>
- [22] Verma, R., Das, A.: What’s in a url: Fast feature extraction and malicious url detection. In: *Proceedings of the 3rd ACM International Workshop on Security and Privacy Analytics*, pp. 55–63. Association for Computing Machinery, ???

- (2017). <https://doi.org/10.1145/3041008.3041016>
- [23] Sicari, S., Rizzardi, A., Grieco, L.A., Coen-Porisini, A.: Security, privacy and trust in internet of things: The road ahead. *Computer Networks* **76**, 146–164 (2015) <https://doi.org/10.1016/j.comnet.2014.11.008>
 - [24] Marchal, S., François, J., State, R., Engel, T.: Phishstorm: Detecting phishing with streaming analytics. *IEEE Transactions on Network and Service Management* **11**(4), 458–471 (2014) <https://doi.org/10.1109/TNSM.2014.2377295>
 - [25] Mumtaz, M., Tayyab, M., Jhanjhi, N.Z., Muzammal, S.M., Hameed, K.: Privacy preserving data analysis with generative ai. In: *AI Techniques for Securing Medical and Business Practices*, pp. 391–410. IGI Global Scientific Publishing, ??? (2025). <https://doi.org/10.4018/979-8-3693-8939-3.ch014>
 - [26] Muzammal, S.M., Bibi, R., Waseem, H., Din, S.N.U., Jhanjhi, N.Z., Tayyab, M.: Federated learning for collaborative cyber defense. In: *Utilizing Generative AI for Cyber Defense Strategies*, pp. 1–28. IGI Global Scientific Publishing, ??? (2025). <https://doi.org/10.4018/979-8-3693-8944-7.ch001>
 - [27] Mohammad, R.M., Thabtah, F., McCluskey, L.: Predicting phishing websites based on self-structuring neural network. *Neural Computing and Applications* **25**(2), 443–458 (2014) <https://doi.org/10.1007/s00521-013-1490-z>
 - [28] Rawla, A., Singh, S., Daniyal, M., Dubey, P.: Detection of phishing attacks in phiusiil dataset using deep learning. *Procedia Computer Science* **259**, 543–552 (2025) <https://doi.org/10.1016/j.procs.2025.04.003>
 - [29] Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: *Proceedings of the 32nd International Conference on Machine Learning*, pp. 448–456. PMLR, ??? (2015). <https://proceedings.mlr.press/v37/ioffe15.html>
 - [30] Alsarhan, A., Igried, B., Bani Saleem, R.M., Alauthman, M., Aljaidi, M.: Enhancing phishing url detection: A comparative study of machine learning algorithms. In: *Proceedings of the 2023 Asia Conference on Artificial Intelligence, Machine Learning and Robotics*, pp. 1–7. Association for Computing Machinery, ??? (2023). <https://doi.org/10.1145/3625343.3625348>
 - [31] Chollet, F.: *Deep Learning with Python*, 2nd edn. Manning Publications, ??? (2021). <https://www.manning.com/books/deep-learning-with-python>
 - [32] Yerima, S.Y., Alzaylaee, M.K.: High accuracy phishing detection based on convolutional neural networks. *arXiv arXiv:2004.03960* (2020) <https://doi.org/10.48550/arXiv.2004.03960>

- [33] Safi, A., Singh, S.: A systematic literature review on phishing website detection techniques. *Journal of King Saud University - Computer and Information Sciences* **35**(2), 590–611 (2023) <https://doi.org/10.1016/j.jksuci.2023.01.004>
- [34] Thakur, K., Ali, M.L., Obaidat, M.A., Kamruzzaman, A.: A systematic review on deep-learning-based phishing email detection. *Electronics* **12**(21) (2023) <https://doi.org/10.3390/electronics12214545>
- [35] Baskota, S.: Phishing url detection using bi-lstm. *arXiv* **arXiv:2504.21049** (2025) <https://doi.org/10.48550/arXiv.2504.21049>
- [36] Maci, A., Santorsola, A., Coscia, A., Iannacone, A.: Unbalanced web phishing classification through deep reinforcement learning. *Computers* **12**(6) (2023) <https://doi.org/10.3390/computers12060118>
- [37] Alhuzali, A., Alloqmani, A., Aljabri, M., Alharbi, F.: In-depth analysis of phishing email detection: Evaluating the performance of machine learning and deep learning models across multiple datasets. *Applied Sciences* **15**(6) (2025) <https://doi.org/10.3390/app15063396>
- [38] Prasad, P.S.: Phishing url detection using xgboost and custom feature engineering. *International Journal for Research in Applied Science and Engineering Technology* **13**(5), 675–686 (2025) <https://doi.org/10.22214/ijraset.2025.70277>
- [39] Alshingiti, Z., Alaqel, R., Al-Muhtadi, J., Haq, Q.E.U., Saleem, K., Faheem, M.H.: A deep learning-based phishing detection system using cnn, lstm, and lstm-cnn. *Electronics* **12**(1) (2023) <https://doi.org/10.3390/electronics12010232>
- [40] Ujah-Ogbuagu, B.C., Akande, O.N., Ogbuju, E.: A hybrid deep learning technique for spoofing website url detection in real-time applications. *Journal of Electrical Systems and Information Technology* **11**(1), 7 (2024) <https://doi.org/10.1186/s43067-023-00128-8>
- [41] Douzas, G., Bacao, F.: Effective data generation for imbalanced learning using conditional generative adversarial networks. *Expert Systems with Applications* **91**, 464–471 (2018) <https://doi.org/10.1016/j.eswa.2017.09.030>
- [42] Lin, Z., Shi, Y., Xue, Z.: Idsgan: Generative adversarial networks for attack generation against intrusion detection. In: *Advances in Knowledge Discovery and Data Mining: 26th Pacific-Asia Conference, PAKDD 2022*, pp. 79–91. Springer, ??? (2022). https://doi.org/10.1007/978-3-031-05981-0_7
- [43] Orunsolu, A.A., Sodiya, A.S., Akinwale, A.T.: A predictive model for phishing detection. *Journal of King Saud University - Computer and Information Sciences* **34**(2), 232–247 (2022) <https://doi.org/10.1016/j.jksuci.2019.12.005>

- [44] Aljofey, A., Jiang, Q., Qu, Q., Huang, M., Niyigena, J.-P.: An effective phishing detection model based on character level convolutional neural network from url. *Electronics* **9**(9) (2020) <https://doi.org/10.3390/electronics9091514>
- [45] Do, N.Q., Selamat, A., Krejcar, O., Herrera-Viedma, E., Fujita, H.: Deep learning for phishing detection: Taxonomy, current challenges and future directions. *IEEE Access* **10**, 36429–36463 (2022) <https://doi.org/10.1109/ACCESS.2022.3151903>
- [46] Friedman, J.H.: Greedy function approximation: A gradient boosting machine. *The Annals of Statistics* **29**(5), 1189–1232 (2001) <https://doi.org/10.1214/aos/1013203451>
- [47] Akçam, , Tekerek, A., Tekerek, M.: Development of bilstm deep learning model to detect url-based phishing attacks. *Computers and Electrical Engineering* **123** (2025) <https://doi.org/10.1016/j.compeleceng.2025.110212>
- [48] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research* **15**(1), 1929–1958 (2014)
- [49] Barik, K., Misra, S., Mohan, R.: Web-based phishing url detection model using deep learning optimization techniques. *International Journal of Data Science and Analytics* **20**(5), 4449–4471 (2025) <https://doi.org/10.1007/s41060-025-00728-9>
- [50] Do, N.Q., Selamat, A., Krejcar, O., Yokoi, T., Fujita, H.: Phishing webpage classification via deep learning-based algorithms: An empirical study. *Applied Sciences* **11**(19) (2021) <https://doi.org/10.3390/app11199210>
- [51] Shoaib, M., Umar, M.S.: Url based phishing detection using machine learning. In: 2023 6th International Conference on Information Systems and Computer Networks (ISCON), pp. 1–7 (2023). <https://doi.org/10.1109/ISCON57294.2023.10112184>
- [52] Le, H., Pham, Q., Sahoo, D., Hoi, S.C.H.: Urlnet: Learning a url representation with deep learning for malicious url detection. *arXiv* **arXiv:1802.03162** (2018) <https://doi.org/10.48550/arXiv.1802.03162>