Original Research Paper

Phishing Website Detection Using Improved Multilayered Convolutional Neural Networks

[1]Hadia Bibi, [2]Syed Rehan Shah, [3]Mirza Murad Baig, [4]Muhammad Imran Sharif, [5]Mehwish Mehmood, [6]Zahid Akhtar and [7]Kamran Siddique

[1]*Department of Computer Science, Bahauddin Zakariya University, Multan, Pakistan*
[2]*Department of Computer Science, Muhammad Nawaz Shareef University of Agriculture, Multan, Pakistan*
[3]*Department of Computer Science, National University of Modern Languages, Islamabad, Pakistan*
[4]*Department of Computer Science, Kansas State University, Manhattan, KS, USA*
[5]*Department of Electrical and Computer Engineering, Comsats University Islamabad, Islamabad Campus, Pakistan*
[6]*Department of Network and Computer Security, University of New York Polytechnic Institute, Utica, USA*
[7]*Department of Computer Science and Engineering, University of Alaska Anchorage, Anchorage, USA*

Article history
Received: 26-02-2024
Revised: 16-03-2024
Accepted: 30-03-2024

Corresponding Author:
Muhammad Imran Sharif
Department of Computer Science, Kansas State University, Manhattan, KS, USA
Email: imransharif@ksu.edu

Abstract: The internet has become an essential part of many fields: Communication, entertainment, commerce, industrial production, agriculture, etc. Unfortunately, online users are vulnerable to various attacks; this could lead to financial damages and loss of personal information. Phishing is seen as an internet threat and a cybercrime where anyone can capture personal information and data by posing as a reliable source. Data may include passwords to access confidential private or industrial repositories, emails, banks, financial information, etc. The prediction task is one of the crucial aspects of modern security systems, including anti-virus, firewall, and anti-spyware software. Currently, there is no availability of a single technique that can effectively detect every phishing attack. This study proposes a novel intelligent approach, phishing Prediction, using machine learning and deep learning to accurately predict phishing websites. We apply a pre-processing pipeline and develop the model using four machine learning models namely decision tree, Naive Bayes, support vector machine random forest, and Convolutional Neural Network (CNN) as a deep learning model. The UCI machine learning repository dataset comprised 11,055 websites, including lists of 4898 phishing and 6157 legitimate websites. The multilayered CNN has achieved the highest accuracy of 99.1% among all the listed algorithms, showcasing a precision of 97, a recall of 96%, and an F1-score of 96%.

Keywords: Machine Learning, Deep Learning, Phishing Website Prediction, Classification, CNN

Introduction

Phishing is a serious threat that numerous companies face in today's digital age. It involves cunningly deceiving unsuspecting individuals into revealing their personal information and data by masquerading as trustworthy (Alkhalil *et al*., 2021). These deceptive techniques can be directed toward various targets, including email accounts, financial details, debit card information, and the identification associated with Internet of Things (IoT) devices. The techniques for detecting phishing attacks encounter various accuracy and highly alarming issues (MacGregor John-Otumu *et al*., 2021).

In this context, this is crucial to detect and prevent website phishing attempts for security software systems, such as anti-virus programs, firewalls, anti-spyware tools, and intrusion detection systems (Selvan and Vanitha, 2016; Babagoli *et al*., 2019). An expectation marks the anti-phishing environment for more precise and efficient approaches. At the same time, binary detection techniques have been widely utilized to detect phishing attempts based on historical data and forecasts (Dewis and Viana, 2022). Artificial Intelligence (AI) solutions have shown promising results. Nevertheless, these techniques have their difficulties, specifically regarding time-consuming processing, especially when working with comparatively minor datasets. Scalability is also an issue when employing ML approaches in smaller contexts (Jameel and George, 2013). Despite their value, heuristics-based phishing detection algorithms have a non-negligible incidence of false positives. In response, previous research efforts have focused on tactics incorporating feature reduction and ensemble models to enhance the efficiency of phishing detection models (Alkhalil *et al*., 2021). Users frequently underestimate the significance of a website's URL, leaving them more vulnerable to phishing assaults. It is essential to be attentive and thoroughly check the legitimacy of URLs to reduce the potential risks (Lazar *et al*., 2021). The catastrophic effects of stealing victim-sensitive information with phishing attempts can be successfully averted by predicting early attempts. Regrettably, the efficiency of traditional approaches for detecting such assaults remains restricted since, on average, they detect only 20% of total attempts (Catal *et al*., 2022; Shah *et al*., 2023). The novelty of the paper highlights the deep learning approach to fill the literature gap and the selection of Multilayered Convolutional Neural Networks (ML-CNN) with four distinct feature classes. Each class contains multiple values or features. The proposed approach will address the scalability and handling of large dataset issues with binary or linear techniques. The CNN comprises simple processing units called neurons, facilitating pattern learning (Abunadi *et al*., 2013). Similar to the neurons found in the human brain, these units exhibit exceptional proficiency in parallel processing, input-output mapping of non-linear systems, and drawing generalized conclusions from hitherto unknown data (Abiodun *et al*., 2019). Mapping nonlinear systems allows CNNs to respond accurately even when confronted with unconventional training patterns (Efendy *et al*., 2022). Furthermore, for prediction predicaments like identifying phishing attacks, which adhere to fixed data patterns, neural network models enable the assimilation of past data by fostering improved accuracy (Mohammad *et al*., 2014). Nonetheless, it is worth noting that the training process of CNNs can be comparatively slow when juxtaposed with pre-trained alternative machine learning models (Catal *et al*., 2022). This manuscript contributes to developing an innovative approach by selecting four unique features and using Multilayered CNN architecture. CNN is a deep learning method that provides better accuracy and reliable prediction of phishing websites because of its exceptional ability to discover patterns, extract key information, and accurately categorize URLs.

This study employed Custom Multilayer (ML) CNN, where we initialized five Hiiden layers, an input layer, and an output layer with 164 neurons. Second, the proposed method commences with diligent data preprocessing to ensure optimal utilization of information during the neural network's learning and subsequent prediction phases. The method can analyze changing URL patterns and fill the existing literature gap.

Literature Review

Detecting a phishing website is crucial and hidden patterns could be disguised as legitimate. Various research studies attempted to detect or predict website suspicious activities using ML and Deep Learning (DL) techniques shown in Table 1.

Table 1: Comparison of different approaches from the literature

Ref	Problem	Methodology	Feature extraction method	Prediction method	Dataset	Accuracy %
Alnemari and Alshammari (2023)	Identify phishing Websites	Ensemble classifier techniques	Extracted features from legitimate and Phishing Websites	DT, RF, SVM, kNN, AdaBoost, Bagging	Phishing websites dataset	97.30
Awasthi and Goel (2022)	Predict phishing websites	Stacked ensemble and hybrid feature selection methods	Features extracted from legitimate and Phishing websites	Extra Tree classifier RF, NB, J48, KNN	PhishTank dataset	99.18
Alshingiti *et al*. (2023)	Identify phishing Websites	Deep learning algorithms (LSTM, CNN and LSTM-CNN)	Features extracted from website URLs	LSTM CNN and LSTM CNN	Phishing websites dataset	99.20
Nagunwa *et al*. (2022)	Detect phishing emails	A hybrid approach using deep learning and natural language	Features extracted from the text and numerical data	LSTM for text data, MLP for numerical data	Phishing emails dataset	94.00
Ariyadasa *et al*. (2022)	Detect phishing Websites hosted on FFSNs-hosted websites	Machine learning approach	Features extracted from hostnames of	Traditional machine learning algorithms	PhishTank dataset	98.42 and 97.81
Aljofey *et al*. (2022)	Detect phishing websites	A practical approach using HTML and web-page content	HTML and webpage content	Traditional machine learning algorithms	PhishTank dataset	96.76 on their dataset, 98.48 benchmark dataset
Alswailem *et al*. (2019)	Detect phishing Websites	Random forest-based technique	26 features extracted from URL and HTML code	Random forest	PhishTank dataset	98.80%
Rao *et al*. (2020)	Detect phishing websites	Feature extraction using 48 compelling features	48 features extracted from the URL	Eight machine learning algorithms LSTM	CatchPhish datasets	94.59 (random forest classifier
Wei *et al*. (2020)	Detect phishing attacks	LSTM classification model	4 Four features and 30 sub-features extracted from phishing data	LSTM	Phishing data	96.55
Kalabarige *et al*. (2022)	Detect phishing websites approach	The multilayered stacked ensemble learning 2020 (D3, D4) datasets	URL and HTML features	Estimators at various levels	UCI (D1), Mendeley 2018 (D2), and Mendeley	96.79-98.90
Saha *et al*. (2020)	Identify suspicious websites	Multilayer perceptron based approach using deep learning	10 Ten attributes of the dataset obtained from Kaggle	Multilayer perceptron	Kaggle dataset	93.00
Kalabarige *et al*. (2022)	Identify phishing attacks	Random Forest algorithm	URL and HTML features	Random forest	PhishTank dataset	94.79

The study described in Alnemari and Alshammari (2023) focuses on advancing predictive models in identifying phishing websites utilizing two distinct ML modules. The researchers employ cross-validation, a widely embraced method in ML, to thoroughly assess the performance of their models. The article aims to improve the accuracy of predicting phishing attempts by analyzing the efficacy of various classifier algorithms. The base classifiers utilized a decision tree, random forest, Support Vector Machine (SVM), and k-nearest Neighbors (k-NN). The ensemble classifier techniques employed include AdaBoost, Bagging, and Random Subspace. The authors use a dataset comprising extracted features from legitimate and Phishing websites to conduct their research. The study results demonstrate that ensemble classifier techniques consistently outperform the base classifiers regarding accuracy, precision, recall, and F1 score. Among all the models tested, the Random Subspace ensemble technique showcases the highest level of performance with 97.3% accuracy, 96.9% precision, 98.2% recall, and 97.6% F1-score.

The research study described in Awasthi and Goel (2022) primarily focused on predicting phishing websites by implementing a stacked ensemble and hybrid feature selection methods. The researchers obtained promising results by conducting experiments and meticulously studying results from techniques using diverse datasets. They have used the RF, NB, J48, and KNN machine learning models. The significant improvement in the accuracy of the Extra Tree classifier by 99.18% highlights the effectiveness of their strategies in accurately identifying and preventing fraudulent online activities, ultimately ensuring the security of users' sensitive information.

This study's authors Alshingiti *et al*. (2023) propose utilizing LSTM, CNN, and LSTM-CNN algorithms for identifying and categorizing website URLs as either real or phishing. The suggested approach demonstrates outstanding performance in detecting phishing websites. However, the deep learning methods exhibited different performance levels on the comparing dataset. The CNN algorithm achieved a prominent accuracy of 99.2%, while the LSTM-CNN and LSTM algorithms achieved 97.6% and 96.8%, respectively. The study described by Nagunwa *et al.* (2022) suggests a hybrid approach for detecting phishing emails using DL and NLP. Phishing and spam emails are unwanted if not handled properly and these attacks could result in disaster for any organization. The proposed method utilized the LSTM module for text-based and multilayer perceptron (MLP) numerical-based datasets and achieved 94% accuracy.

Another ML technique for addressing phishing websites hosted on Fast Flux Service Networks (FFSNs) was introduced by the authors Ariyadasa *et al*. (2022). The proposed method reached an overall accuracy of 98.42% for binary and 97.81% for multi-class prediction tasks, respectively, while demonstrating the effectiveness of features for traditional machine learning algorithms.

The study described by Aljofey *et al*. (2022) suggests PhishDet, a universal technique for predicting phishing attempts with a recurrent long-term convolutional network and convolutional graph network while utilizing features like the URL and HTML of a website. PhishDet achieves an accuracy of 96.42% for detection, with 0.036 false negative rates. Similarly, the authors in this study Aljofey *et al*. (2022) suggest a practical approach to detect phishing websites using HTML and webpage content. Their approach achieves 96.76% accuracy having a 1.39% percentage of false positives with their collected samples and a precision of 98.48% on the comparable dataset resulting in a 2.09% false positive ratio.

A random forest-based technique for detecting phishing websites is suggested by Alswailem *et al*. (2019). Their proposed method achieves 98.8% accuracy while using 26 features. Another strategy for predicting phishing websites' suspicious attempts using ML-based URL static analysis is proposed by Korkmaz *et al*. (2020). Their approach performs feature extraction to obtain 48 compelling features. The researchers use eight ML modules to justify the URLs with three distinct datasets. The experimental results of the proposed method achieve up to 94.59% accuracy using the RF module while incorporating the CatchPhish datasets (Rao *et al*., 2020).

A comparison study on various phishing attacks on multiple websites is performed by Wei *et al*. (2020). They have used four main and thirty sub-features to predict phishing attacks in the LSTM classification model. The initialized LSTM technique produces 96.55% accuracy. Similarly, the article by Kalabarige *et al.* (2022) provides a multilayered stacked ensemble learning approach that employs estimators at various levels. Estimator predictions in each layer have been utilized as input for the next layer. Through experimental assessment, it was highlighted how the suggested model performs admirably across various datasets, with accuracies ranging from 96.8-98.9%. UCI (D1), Mendeley 2018 (D2), and Mendley 2020 (D3, D4) datasets were utilized for assessment.

The research article by Saha *et al*. (2020) proposes a multilayer perceptron-based approach using deep learning to identify suspicious websites. Using ten attributes of the dataset obtained from Kaggle, the proposed model can achieve 93% accuracy. In contrast, the article by Kalabarige *et al*. (2022) suggests using the Random Forest algorithm to identify phishing attacks. They compare multiple ML modules and the RF module shows the highest accuracy of 94.79%.

Furthermore, a Multilayer CNN model is proposed to allow the detection of phishing websites effectively. This method has demonstrated its efficiency in significantly boosting phishing detection ability. This study aims to increase the accuracy and efficiency of detecting and combating phishing attempts by concentrating on selecting and extracting unique and informative aspects of the dataset. The emphasis on obtaining unique features from the dataset distinguishes the proposed technique from the previously stated deep learning-based methods. Therefore, a Multilayer CNN model is proposed to identify the prominent 30 features from the dataset and effectively identify the correlation between the pattern for new data.

Materials and Methods

Dataset and Environment Setup

This section details the steps taken in dataset preprocessing and setting up experiment setup.

Preprocessing Pipeline

In the preprocessing phase, our focus was on ensuring the dataset was well-prepared for the subsequent modeling steps. This involved several key steps as outlined below.

Data Collection

The initial step involved obtaining the dataset from the UCI machine learning Repository, which comprised 11,055 websites meticulously identified and examined by Karabatak and Mustafa (2018). The dataset includes lists of 4898 phishing and 6157 legitimate websites and converting the file format from ARFF to CSV shown in Table 2. However, in real-world scenarios, there is a high proportion of legit websites compared to phishing websites. Phishing websites are only created for malicious acts and represent a low proportion of the internet. Therefore, the dataset consists of 4898 phishing and 6157 legitimate website data. Initially, the Synthetic Minority Oversampling Technique (SMOT) was evolved to equalize the phishing and legitimate website data imbalance.

Feature Selection and Encoding

After the conversion, data were preprocessed by selecting 30 unique features from the dataset. The selected feature was important to distinguish legitimate and phishing websites effectively. Next, we have applied one-hot encoding to accurately convert any categorical data into numerical labels the data is accurately shown in Table 3. This encoding paragram reduces the risk of misinterpretation in data by representing each category independently. Furthermore, the dataset contains 1, -1, and 0. The cleaning process cleaning involves converting the -1 into 0 to cover the NaN and missing values. The value of 1 indicates a TRUE result, indicating that the website under consideration is not a phishing website shown in Table 3. Conversely, a value of 0 indicates a false result, indicating that the website being evaluated is a phishing site. It is critical to evaluate categorical variable encoding as numerical values to avoid misinterpretation. Assigning numerical labels (such as 0, 1, 2) to categories without using a one-hot encoding approach may accidentally suggest an order or magnitude that does not exist. One-hot encoding solves this problem by expressing each category independently, avoiding potential misinterpretations Guptta *et al*. (2024).

Environmental Setup

The environmental setup is powered by an Intel Core i7 7th generation CPU in a Dell Latitude laptop, well-known for its dependability and adaptability. The system has a 1TB storage space for data and files, with an extra 520GB set aside for specialized reasons. The installation also contains Weka 3.0, Pycharm IDE, an important data mining and analysis software for experiments, and Python version 3.6. We ensured accurate results by creating the environment for the experiment and adopting the preprocessing steps carefully to detect phishing websites using Multilayer CNN.

Training and Testing Split

The technique of separating datasets is critical in the classification of the training and prediction task. The dataset was split into 70% for training and 30% for testing in our scenario. The training set comprises organized and distinct data with labels indicating whether a website is authentic or a phishing site. The remaining 30% of the data is set aside for testing, which allows us to evaluate the module’s efficiency and discover any potentially perplexing projections.

Table 2: Overall dataset representation

Dataset representation	Count
Extracted unique features	30
Dataset of websites	11055
Total phishing	4898
Total legitimate	6157

Table 3: Encoded dataset

Encoded	Result	Description
1	True	Legitimate website
0	False	Phishing website

DOI: 10.3844/jcssp.2024.1069.1079

Table 4: Selected innovative domain-based features

Features	Description
Domain Age	A domain can last for a long or short-term period. It is essential to determine the domain age
DNS Record	This unique feature is essential as it doesn't appear in WHO searches
Website Traffic	This feature is responsible for providing any website traffic, whether high or low
Page Rank	This unique domain feature provides link quality
Google Index	This feature provides analysis of videos and images on the website
A link pointing to a page	This domain feature indicates link building from one page to another, indicating authenticity
Statistical Report	This feature indicates the list of all subdomains

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 11055 entries, 0 to 11054
Data columns (total 32 columns):
 #   Column                       Non-Null Count  Dtype
---  ------                       --------------  -----
 0   id                           11055 non-null  int64
 1   having_IP_Address            11055 non-null  int64
 2   URL_Length                   11055 non-null  int64
 3   Shortining_Service           11055 non-null  int64
 4   having_At_Symbol             11055 non-null  int64
 5   double_slash_redirecting     11055 non-null  int64
 6   Prefix_Suffix                11055 non-null  int64
 7   having_Sub_Domain            11055 non-null  int64
 8   SSLfinal_State               11055 non-null  int64
 9   Domain_registeration_length  11055 non-null  int64
 10  Favicon                      11055 non-null  int64
 11  port                         11055 non-null  int64
 12  HTTPS_token                  11055 non-null  int64
 13  Request_URL                  11055 non-null  int64
 14  URL_of_Anchor                11055 non-null  int64
 15  Links_in_tags                11055 non-null  int64
 16  SFH                          11055 non-null  int64
 17  Submitting_to_email          11055 non-null  int64
 18  Abnormal_URL                 11055 non-null  int64
 19  Redirect                     11055 non-null  int64
 20  on_mouseover                 11055 non-null  int64
 21  RightClick                   11055 non-null  int64
 22  popUpWidnow                  11055 non-null  int64
 23  Iframe                       11055 non-null  int64
 24  age_of_domain                11055 non-null  int64
 25  DNSRecord                    11055 non-null  int64
 26  web_traffic                  11055 non-null  int64
 27  Page_Rank                    11055 non-null  int64
 28  Google_Index                 11055 non-null  int64
 29  Links_pointing_to_page       11055 non-null  int64
 30  Statistical_report           11055 non-null  int64
 31  Result                       11055 non-null  int64
dtypes: int64(32)
memory usage: 2.7 MB
```

Fig. 1: Illustrations of selected unique features

Selecting Four Distinct Features from the Dataset

After the preprocessing, features of the Address bar were first selected, which comprised 12 sub-features. Secondly, features of Anomalous actions were selected, comprising six sub-sets; thirdly, the HTML and JavaScript features, having five sub-sets, were selected. Lastly, the features of the Domain have been extracted with seven unique sub-features. Figure 1 illustrates all selected features.

The selection of the above 30 distinct features reflects the effectiveness of the phishing detection approach. Furthermore, seven distinct domain features are included among these sub-features, significantly improving the model's accuracy and resilience, as shown in Table 4.

Furthermore, the study aims to create a well-rounded and comprehensive dataset of phishing websites with all the relevant features. Thus, the ML-CNN model can be trained on a comprehensive set of features to find learning patterns and correlations between the features. Ultimately, the selected features provide a robust architecture for the ML-CNN model to detect malicious websites. This study has potential limitations such as class imbalance within the dataset which can lead our model to biased predictions which were handled by the SMOTE. Secondly, while we selected 30 unique features for distinguishing between legitimate and phishing websites some of the unique features were not considered.

Machine Learning Models

DTC

A decision tree classifier is a machine learning algorithm used for both classification and regression tasks (Hasan *et al*., 2022). It models decisions based on a tree-like structure, where each internal node represents a feature and each branch represents a decision rule leading to a leaf node with a class label. When classifying data, it traverses the tree from the root to a leaf, following the decision rules, and assigns the most frequent class to that leaf. Decision trees are interpretable, handle both categorical and numerical data, and are prone to overfitting, which can be mitigated through techniques like pruning. They are widely used for their simplicity and effectiveness in various applications.

NB

A Naive Bayes classifier is a probabilistic machine learning algorithm commonly used for classification tasks (Murphy, 2006). It's based on Bayes' theorem and the assumption of feature independence, hence "Naive." It calculates the probability of an instance belonging to a class by multiplying the probabilities of each feature occurring in that class and then normalizing. Despite its simplifying assumptions, Naive Bayes often performs surprisingly well in text classification and spam detection. It's computationally efficient and can handle high-dimensional data. However, it may struggle when the feature independence assumption is significantly violated and it doesn't capture complex relationships between features.

SVC

A Support Vector Classifier (SVC) is a powerful machine learning algorithm used for binary classification tasks Hasan *et al*. (2023a). It operates by finding the optimal hyperplane that best separates data points belonging to different classes while maximizing the margin between them. Support vectors are the data points closest to the decision boundary, which helps define the hyperplane. SVC aims to find the hyperplane that minimizes classification errors and generalizes well to unseen data. It can handle both linear and nonlinear classification problems through kernel functions. While effective and robust, SVC can be sensitive to outliers and its performance may degrade when dealing with large datasets.

RF

A random forest classifier is an ensemble machine-learning model used for classification and regression tasks Hasan *et al*. (2023b). It works by constructing multiple decision trees during training, each based on a random subset of the data and features. The final prediction is made by aggregating the results of these trees, often through a majority vote for classification. Random forests are highly effective because they reduce overfitting and increase accuracy compared to individual decision trees. They handle high-dimensional data, are robust to outliers, and can capture complex relationships. Random forests are widely used for various applications, making them a versatile and powerful tool in machine learning.

Deep Learning Method

Convolutional neural networks are particularly good at quickly learning significant features from raw input Marjan *et al*. (2022). These characteristics might include visual patterns, linguistic content, or structural components, making CNNs useful for detecting phishing websites. Support Vector Machine (SVM), Decision Tree (DT) and Na¨ıve Bayes (NB) frequently rely on manually built features, which can be time-consuming to develop and may not capture the entire complexity of the data. Key attributes can be easily extracted and used, directly from the input data by exploiting the intrinsic capabilities of CNNs, resulting in more effective and complete analysis. The Multilayered CNN (ML-CNN) model consists of five layers, each utilizing different activation functions such as Relu and Softmax (Kattenborn *et al*., 2021).

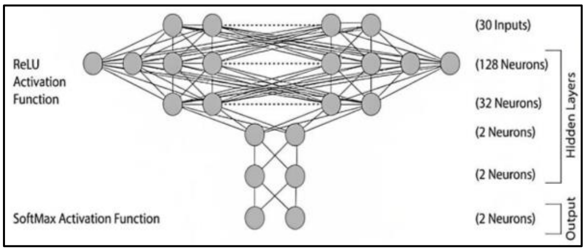

Fig. 2: Multilayer CNN input and output functions

Figure 2 describes the indictive structure of the steps and details of the input and output functions of the model. The first convolutional layer of the ML-CNN model was configured to accept 30 input features as enabled with the relu activation function. The aforementioned setup enables the machine learning algorithm to receive and analyze a dataset with 30 distinct characteristics, allowing it to capture different incoming data elements. The convolutional or hidden layers with 128, 30, 2, and 2 neurons worked as a simulated and abstract environment for complex data processing, respectively. This processing involves intricate changes in the raw data and entails the complex gathering of patterns, relationships, and important information from the input data.

The outputs of this processing are subsequently passed on to underlying layers. Each hidden layer polishes the data, eliminating and combining it to provide more in-depth findings. Lastly, the output layer with 2 set neuron servers is positioned where the model results have been generated. These results are based on provided phishing data and corresponding features.

The model performance has been ensured by applying the Softmax activation function to the output layer. The function converts raw scores into probability for the phishing and legitimate classes during prediction.

Figure 3 illustrates the overall prediction of phishing websites; the UCI phishing websites dataset was examined to learn more about the URLs. First, the data's attributes were analyzed to learn about its content, structure, and qualities. The proposed data types were examined for each attribute in the dataset to check that they were properly allocated. A comprehensive assessment procedure was implemented to identify the best features, considering their relevance and contribution to the prediction task. All missing values in the dataset are eliminated to maintain data integrity and reliability. The dataset is split into two groups (training and testing) to assess the overall learning achievement of the proposed model. This divide enables the model to efficiently train on a sample of the data and assess how well it performs on previously unknown examples.

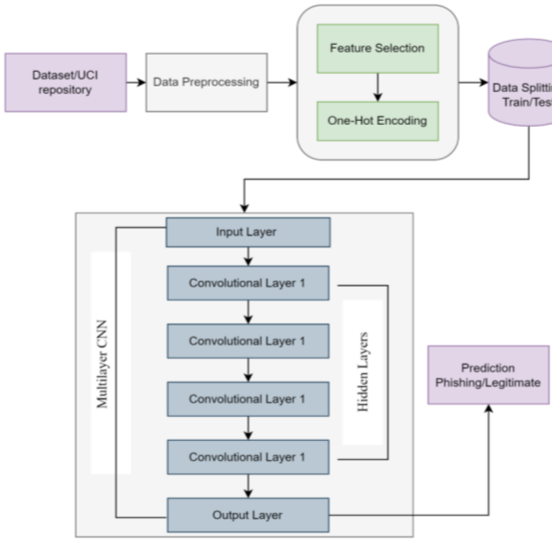

Fig. 3: Flow chart of multilayer CNN

Evaluation Metrics

This part summarizes the parameters used to assess deep learning algorithms' performance. The effectiveness of ML prediction modules is evaluated by examining the classification predictive algorithm results. This manuscript showcases prediction outcomes that are investigated using a variety of measures, including precision, recall, F1-score, confusion matrix, and accuracy. The above metrics were used to estimate the ML-CNN module's efficacy in predicting phishing websites. Accuracy: The proportion of correctly predicted variables of a given class to its actual members in the dataset measures the prediction technique's accuracy.

We may use the following equation to determine the model's accuracy. Typically, a prediction model yields four distinct outcomes: True Positive (*TP*), True Negative (*TN*), False Positive (*FP*) and False Negative (*FN*). The precision is determined by calculating the total phishing websites correctly classified as an actual class. The recall is the proportion of phishing URL predictions that the prediction system predicted right out of all the URLs in the data. The precision and recall of a classifier are combined to produce a harmonic mean. It is an F1-score:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{4}$$

Results and Discussion

In this study, we employed Multilayer CNN with 30 unique phishing website features and performed a comparison with decision tree, Naïve Bayes, random forest and SVM. The correlation of the variables is in Fig. 4. The correlation indicates that the variables have a weak relationship among them that why we have selected the most important features from the huge number of features to make it more trainable for the ML and DL models.

Table 5 illustrates the results after calculating evaluation metrics; first, the DTC algorithm achieved an accuracy of 93.44%. Both precision and recall were at 93.00%, resulting in an F1-score of 93.00%. Second, the NB algorithm attained an accuracy of 92.00%. The model exhibited 93.00% precision, a recall of 90.4%, and an F1-score of 92.00%. Third, among the listed algorithms, SVM scores the maximum accuracy of 93.62%. It demonstrated a precision of 95.00%, a recall of 91.00%, and an F1-score of 93.00%. Fourth, The RF algorithm outperformed the others with an accuracy of 96.47%. It displayed a precision of 97.00%, a recall of 95.00%, and an F1-score of 96.00%. Lastly, the proposed multilayer CNN reached a prominent accuracy of 99.10% among all the listed algorithms. It showcased a precision of 97.00%, a recall of 96.00%, and an F1-score of 96.00%. The examination through the confusion matrix provides extra information about model-specific performance in predicting phishing websites. Figure 5 illustrates the ML-CNN model has accurately predicted 96% of phishing classes as true positive and misclassified 0.3% as false positive. The comparisons are visualized in Fig. 5.

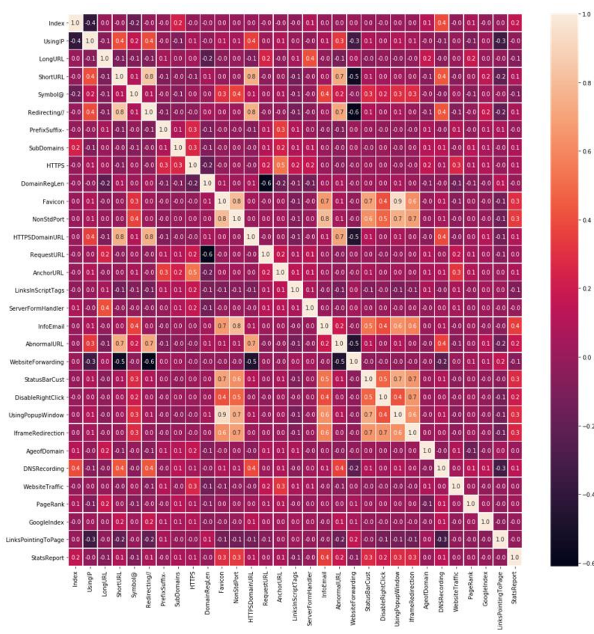

Fig. 4: Correlation of the features

DOI: 10.3844/jcssp.2024.1069.1079

Table 5: Evaluation results of the models

Algorithms	Accuracy %	Precision %	Recall %	F1-score %
DTC	93.44	93.00	93.00	93.00
NB	92.00	93.00	90.40	92.00
SVM	93.62	95.00	91.00	93.00
RF	96.47	97.00	95.00	96.00
Multilayer CNN	99.10	97.00	96.00	96.00

Table 6: Comparison with state-of-the-art model

Reference	Dataset	Methodology	Highest accuracy %
Alnemari and Alshammari (2023)	Phishing web-	Ensembleclassifier techniques/ Bagging/Boosting	97.30
Alshingiti *et al.* (2023)	Sites dataset Phishing web-	Deep learning algorithms (LSTM, CNN and LSTM-CNN)	99.20
Nagunwa *et al.* (2022)	sites dataset phishing	A hybrid approach using deep	94.00
Ariyadasa *et al.* (2022)	Emails dataset Phishtank	Learning and natural language processing Machine learning approach/FFSN	98.40
Aljofey *et al.* (2022)	Dataset Phishtank Dataset	A practical approach using HTML and webpage content	96.80
ML-CNN	UCI/phishing website dataset	Multilayered CNN/ensemble classifiers	99.10

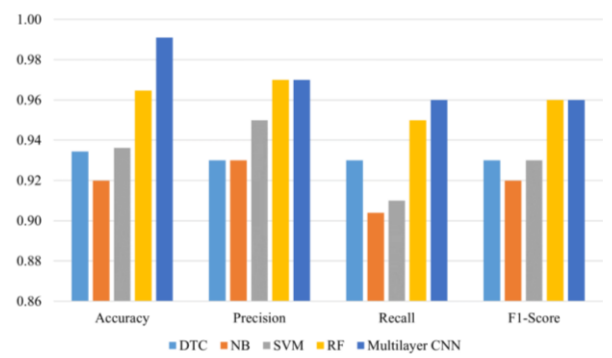

Fig. 5: Comparison of the performance of the different algorithms

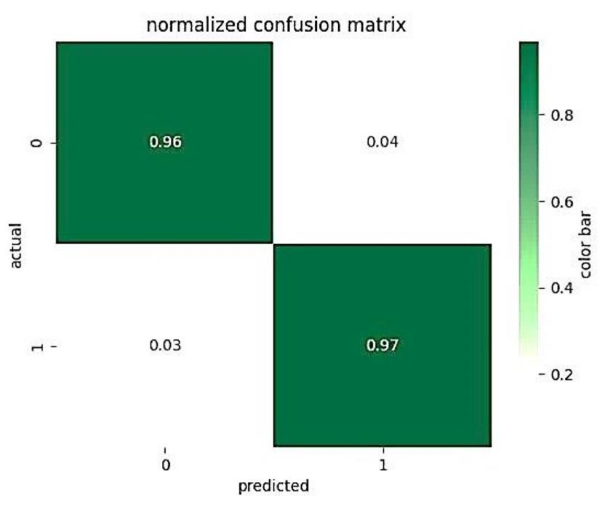

Fig. 6: ML-CNN confusion matrix

Discussion

In today's digital environment, Phishing is a serious problem as people are fooled into disclosing personal information. Security software systems must be able to recognize and stop such fraudulent efforts because phishing assaults largely rely on false emails. Although machine learning techniques have shown promising results in detecting Phishing, there are still issues with processing speed and scalability. Prior research has concentrated on feature reduction and ensemble models to increase effectiveness and accuracy. The convolutional neural networks method is suggested in this research to identify phishing websites identify phishing websites. The study utilized a feature selection process where four different feature classes were selected. These feature classes include address bar, abnormal, HTML and Javascript, and domain-based. The domain-based features were important and this study selected domain age, DNS record, website traffic, page rank, google index, a link pointing to a page, and Statistical Reports. The CNN architecture thrives when evaluating URL structures and identifying patterns to distinguish between legitimate and counterfeit websites. Moreover, enabling multilayered perception in CNN architecture showed an improved and accurate prediction of malicious attempts as shown in Fig. 6. However, different ML models have also performed well while trained on the same dataset. Similarly, we have worked on hyperparameter tunning where the model showed the same accuracy. However, various research studies have used a publically available dataset, which is available on the UCI repository.

The proposed ML-CNN model is compared with the existing modules with publically available datasets from the literature. The dataset was split into 70% for training and 30% for the test; the model was trained only on 50 epochs by considering the overfitting and underfitting problems. Table 6 describes the approach's accuracies compared with the proposed model. However, Multilayered CNN performed well against state-of-the-art approaches such as begging, boosting, ensemble classifiers, and LSTM.

Conclusion

Phishing is a serious issue in today's digital world, making it critical to identify and prevent phishing assaults to maintain the security of persons and businesses. While standard methods have limits in accuracy and efficiency, machine learning techniques, notably CNN, show potential in recognizing phishing websites properly. The model can ensure optimal use of resources and split the dataset in unique ways to assess the hidden patterns. Previous research has looked at various methodologies, such as ensemble classifier techniques, DL algorithms such as LSTM, CNN, and LSTM-CNN, and hybrid approaches that include natural language processing. However, those techniques evolve publically available datasets in identifying phishing websites and emails; these techniques have obtained excellent accuracies ranging from 93-99.2%. In this study, we constructed a multilayered unique detection technique called Phishing Prediction using ML-CNN that uses the CNN architecture's capacity to find patterns, extract critical information, and reliably categorize URLs. The goal is to enhance the overall efficiency of phishing attempt prediction by concentrating on unique and useful characteristics retrieved from the dataset. The ML-CNN model can produce 5.56, 5.38, 7.00, and 2.53% better accuracy than the DT classifier, SVM, NB, and RF modules. In future work, we shall target a generic approach incorporating a CNN-based model while using diverse datasets to detect phishing attacks. Furthermore, ongoing efforts will be directed at refining and improving the feature extraction method. Investigating new data from other sources, such as webpage content, user behavior, and network traffic, might give useful insights for enhanced phishing detection. Furthermore, the trained model will be implemented using API on cloud services, allowing the receiving of data from applications for prediction. Additionally, we will put our efforts into enhancing model performance and its robustness in future updates in phishing attacks on websites. Where model will train on the latest dataset and possess valuable information.

Declaration of Competing Interest. There are no potential competing interests among the authors. All authors have seen the manuscript and approved to submit to your journal. We confirm that the content of this manuscript has not been published or submitted for publication elsewhere.

Acknowledgment

Thank you to the authors and publishers for the support in this research article. We appreciate effort of publisher providing a vast audience and platform for area selection.

Funding Information

The authors have not received any financial support or funding to report.

Author's Contributions

Hadia Bibi: Conceptualization, methodology, data curation and results.

Syed Rehan Shah: Methodology, written original draft and results.

Mirza Murad Baig: Conceptualization, data curation, preparation, interim reviewed and edited.

Muhammad Imran Sharif: Conceptualization, data curation, preparation, interim reviewed and edited.

Mehwish Mehmood, Zahid Akhtar and Kamran Siddique: Validation, visualization and finalization**.**

Ethics

Ethical and informed consent for data used. Any information we give will be used for research only and will not be used for any other purpose and data will not be misused.

References

Abiodun, O. I., Jantan, A., Omolara, A. E., Dada, K. V., Umar, A. M., Linus, O. U., Arshad, H., Kazaure, A. A., Gana, U., & Kiru, M. U. (2019). Comprehensive Review of Artificial Neural Network Applications to Pattern Recognition. *IEEE Access*, *7*, 158820-158846. https://doi.org/10.1109/access.2019.2945545

Abunadi, A., Akanbi, O., & Zainal, A. (2013). Feature Extraction Process: A Phishing Detection Approach. *2013 13th International Conference on Intellient Systems Design and Applications*, 331-335. https://doi.org/10.1109/isda.2013.6920759

Aljofey, A., Jiang, Q., Rasool, A., Chen, H., Liu, W., Qu, Q., & Wang, Y. (2022). An Effective Detection Approach for Phishing Websites Using URL and HTML Features. *Scientific Reports*, *12*(1), 8842. https://doi.org/10.1038/s41598-022-10841-5

Alkhalil, Z., Hewage, C., Nawaf, L., & Khan, I. (2021). Phishing Attacks: A Recent Comprehensive Study and a New Anatomy. *Frontiers in Computer Science*, *3*, 563060. https://doi.org/10.3389/fcomp.2021.563060

Alnemari, S., & Alshammari, M. (2023). Detecting Phishing Domains Using Machine Learning. *Applied Sciences*, *13*(8), 4649. https://doi.org/10.3390/app13084649

Alshingiti, Z., Alaqel, R., Al-Muhtadi, J., Haq, Q. E. U., Saleem, K., & Faheem, M. H. (2023). A Deep Learning-Based Phishing Detection System Using CNN, LSTM, and LSTM-CNN. *Electronics*, *12*(1), 232. https://doi.org/10.3390/electronics12010232

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

Ariyadasa, S., Fernando, S., & Fernando, S. (2022). Combining Long-Term Recurrent Convolutional and Graph Convolutional Networks to Detect Phishing Sites Using URL and HTML. *IEEE Access*, *10*, 82355-82375. https://doi.org/10.1109/access.2022.3196018

Awasthi, A., & Goel, N. (2022). Phishing Website Prediction Using Base and Ensemble Classifier Techniques with Cross-Validation. *Cybersecurity*, *5*(1), 1-23. https://doi.org/10.1186/s42400-022-00126-9

Babagoli, M., Aghababa, M. P., & Solouk, V. (2019). Heuristic Nonlinear Regression Strategy for Detecting Phishing Websites. *Soft Computing*, *23*(12), 4315-4327. https://doi.org/10.1007/s00500-018-3084-2

Catal, C., Giray, G., Tekinerdogan, B., Kumar, S., & Shukla, S. (2022). Applications of Deep Learning for Phishing Detection: A Systematic Literature Review. *Knowledge and Information Systems*, *64*(6), 1457-1500. https://doi.org/10.1007/s10115-022-01672-x

Dewis, M., & Viana, T. (2022). Phish Responder: A Hybrid Machine Learning Approach to Detect Phishing and Spam Emails. *Applied System Innovation*, *5*(4), 73. https://doi.org/10.3390/asi5040073

Efendy, M. A., Munirah, M., & Faizal, M. (2022). *Phishing Malware Detection Using Machine Learning (weka)*. 127-133.

Guptta, S. D., Shahriar, K. T., Alqahtani, H., Alsalman, D., & Sarker, I. H. (2024). Modeling Hybrid Feature-Based Phishing Websites Detection Using Machine Learning Techniques. *Annals of Data Science*, *11*(1), 217-242. https://doi.org/10.1007/s40745-022-00379-8

Hasan, M., Roy, P., & Nitu, A. M. (2022). Cervical Cancer Classification using Machine Learning with Feature Importance and Model Explainability. *IEEE Xplore*, 1-4. https://doi.org/10.1109/icecte57896.2022.10114548

Hasan, M., Zobair, M. J., Akter, S., Ashef, M., Akter, N., & Sadia, N. B. (2023a). Ensemble Based Machine Learning Model for Early Detection of Mother's Delivery Mode. *IEEE Xplore*, 1-6. https://doi.org/10.1109/ECCE57851.2023.10101558

Hasan, M., Marjan, M. A., Uddin, M. P., Afjal, M. I., Kardy, S., Ma, S., & Nam, Y. (2023b). Ensemble Machine Learning-Based Recommendation System for Effective Prediction of Suitable Agricultural Crop Cultivation. *Frontiers in Plant Science*, *14*, 1-18. https://doi.org/10.3389/fpls.2023.1234555

Jameel, N. G. M., & George, L. E. (2013). Detection of Phishing Emails using Feed Forward Neural Network. *International Journal of Computer Applications*, *77*(7), 10-15. https://doi.org/10.5120/13405-1057

Kalabarige, L. R., Rao, R. S., Abraham, A., & Gabralla, L. A. (2022). Multilayer Stacked Ensemble Learning Model to Detect Phishing Websites. *IEEE Access*, *10*, 79543-79552. https://doi.org/10.1109/access.2022.3194672

Karabatak, M., & Mustafa, T. (2018). Performance comparison of classifiers on reduced phishing website dataset. *IEEE Xplore*, 1-5. https://doi.org/10.1109/isdfs.2018.8355357

Kattenborn, T., Leitloff, J., Schiefer, F., & Hinz, S. (2021). Review on Convolutional Neural Networks (CNN) in vegetation remote sensing. *ISPRS Journal of Photogrammetry and Remote Sensing*, *173*, 24-49. https://doi.org/10.1016/j.isprsjprs.2020.12.010

Korkmaz, M., Sahingoz, O. K., & Diri, B. (2020). Detection of Phishing Websites by Using Machine Learning-Based URL Analysis. *IEEE Xplore*, 1-7. https://doi.org/10.1109/icccnt49239.2020.9225561

Lazar, D., Cohen, K., Freund, A., Bartik, A., & Ron, A. (2021). IMDoC: Identification of Malicious Domain Campaigns via DNS and Communicating Files. *IEEE Access*, *9*, 45242-45258. https://doi.org/10.1109/access.2021.3066957

MacGregor John-Otumu, A., Mahmudur Rahman, M. D., & Ugochinyere Oko, C. (2021). An Efficient Phishing Website Detection Plugin Service for Existing Web Browsers Using Random Forest Classifier. *American Journal of Artificial Intelligence*, *5*(2), 66-75. https://doi.org/10.11648/j.ajai.20210502.13

Marjan, M. A., Hasan, M., Islam, M. Z., Uddin, M. P., & Afjal, M. I. (2022). Masked Face Recognition System using Extended VGG-19. *IEEE Xplore*, 1-4. https://doi.org/10.1109/icecte57896.2022.10114484

Mohammad, R. M., Thabtah, F., & McCluskey, L. (2014). Predicting Phishing Websites Based on Self-Structuring Neural Network. *Neural Computing and Applications*, *25*(2), 443-458. https://doi.org/10.1007/s00521-013-1490-z

Murphy, K. P. (2006). Naive Bayes Classifiers. (1). *University of British Columbia*.

Nagunwa, T., Kearney, P., & Fouad, S. (2022). A Machine Learning Approach for Detecting Fast Flux Phishing Hostnames. *Journal of Information Security and Applications*, *65*, 103125. https://doi.org/10.1016/j.jisa.2022.103125

Rao, R. S., Vaishnavi, T., & Pais, A. R. (2020). CatchPhish: Detection of Phishing Websites by Inspecting URLs. *Journal of Ambient Intelligence and Humanized Computing*, *11*(2), 813-825. https://doi.org/10.1007/s12652-019-01311-4

Saha, I., Sarma, D., Chakma, R. J., Alam, M. N., Sultana, A., & Hossain, S. (2020). Phishing Attacks Detection Using Deep Learning Approach. *IEEE Xplore*, 1180-1185. https://doi.org/10.1109/ICSSIT48917.2020.9214132

Shah, S. R., Qadri, S., Bibi, H., Shah, S. M. W., Sharif, M. I., & Marinello, F. (2023). Comparing Inception V3, VGG 16, VGG 19, CNN, and ResNet 50: A Case Study on Early Detection of a Rice Disease. *Agronomy*, *13*(6), 1633. https://doi.org/10.3390/agronomy13061633

Selvan, K., & Vanitha, M. (2016). A Machine Learning Approach for Detection of Phished Websites Using neural networks. *International Journal of Recent Technology and Engineering* (*IJRTE*), *4*(6), 19-23.

Wei, W., Ke, Q., Nowak, J., Korytkowski, M., Scherer, R., & Woźniak, M. (2020). Accurate and Fast URL Phishing Detector: A Convolutional Neural Network Approach. *Computer Networks*, *178*, 107275. https://doi.org/10.1016/j.comnet.2020.107275