Detection of Phishing Domains Using Hybridised Deep Learning Techniques

T.J. Ayo, O.D. Alowolodu
The Federal University of Technology
Nigeria

Abstract

Over the years series of attacks have been launched by attackers in order to steal from users, one of the most prevalent attacks is the phishing attacks, this phishing attacks is has become alarming, various approaches has been used to curb this menace but non has actually provide the best solution to this threatening cybercrime such as phishing and therefore the need for better system tools to counter such approaches is paramount. This research proposes the use of deep learning techniques using Convolutional Neural Networks (CNN), Gated Recurrent Units (GRU), and hybrid models of CNN-GRU in detecting phishing domains. The efficient feature extraction abilities of CNNs help in the usage of URL structures where spatial relations exist, whilst the sequential structural formulation of domain names is modeled using GRUs. CNN-GRU model Their integration allows the local pattern detection of the CNN to be meshed with the long sequence utilization of GRU for better accuracy. All models are subjected to standard datasets for phishing with conclusive performance metrics such as Detection Rate (DR), Precision, Recall, F1-Score, and Accuracy. The system utilized two datasets small and large datasets. In this regard, a hybrid CNN-GRU model did show better resilience against phishing attempts than its singular counterparts on a small dataset with an accuracy of 0.9440 while CNN 0.9337 and GRU 0.9402. For the larger datasets CNN outperformed the other two models with an accuracy of 0.9382 while GRU had 0.9026 and CNN-GRU settled for 0.9261. The primary aim of this paper is to evaluate standalone models and hybridized model on two datasets (small and large) to know which of datasets they can perform better in detecting phishing attacks.

Keywords: Deep Learning; Hybrid Approach; Phishing Domains Detection

1. Introduction

In the 21st century, the growing demand for technology and the desire to connect globally have significantly increased user reliance on digital media [1], [2]. While digitization offers convenience, this dependence can become hazardous over time [3]. While digital media enables us to connect with others and gain knowledge across various interests, the increasing reliance on technology and the widespread use of devices connected to the internet have also heightened our vulnerability to cyberattacks. The growth of the Internet of Things (IoT) and other emerging technologies has introduced new vulnerabilities, emphasizing the need for individuals and organizations to remain vigilant against cyber threats [4] which pose significant dangers to regular users [5]. Phishers primarily exploit familiarity through social engineering, capitalizing on users' psychological and behavioral tendencies [6].

Typically, the initiation of a phishing attack starts with an email that seems to be sent to the victim by an authentic source and then requests an update of user information accessing through the link sent. Phishers use many different techniques to initiate phishing attacks; the main methods used are email, SMS, social media, instant messaging, search engines, and malicious websites [7].

The spear phishing is a highly targeted form of phishing attack [7]. Rather than sending more phishing emails to anyone, the phisher sends spoofed emails to consumers that appear to originate from somebody they know [8]. Attackers have proven that relying solely on legal frameworks to curb phishing crimes is insufficient. This highlights the need to detect malicious domains through their URLs and contents to prevent attacks by using robust techniques to detect and prevent such attacks hence the proposed Deep Learning techniques.

2. Literature Review

The significant growth of internet usage has led to its proliferation by attackers. Phishing has been a significant form of attack due to the level of awareness of the victims about cyber ethics. Various approaches had been employed to curb this menace of which we have the traditional approach and the machine learning approach.

 | DOI: 10.20533/jitst.2046.3723.2024.0108

2.1. Traditional Approach

Blacklisting methods, content-based methods, heuristic approaches, and conventional machine learning techniques have been proposed to prevent phishing attacks, as highlighted by [9]. Heuristic-based approaches detect abnormalities on a website by applying a collection of rules derived from long-term observation. The challenge with this method is its tendency to produce high false positives.

Blacklisting and Heuristic approaches were the known methods for phishing domains detection. Phishing solutions like the Phish Tank till today make use of this techniques. However, considering the time to detect, confirm and publish malicious URLs in the database. To address the shortcomings of the traditional methods, researchers have turn to machine learning approaches for phishing detection.

2.2. Machine Learning Approach

Due to the fact that machine learning can learn from large data and accurately predict outcome, it has become a better choice over the traditional method that were in use. However, machine learning methods usually use feature selection to extract sensitive information. When the URL-based features are converted to a feature vector, those features can be directly applied to a machine learning algorithm.

Research by [10] tackled the challenge of detecting phishing websites using machine learning techniques. The higher accuracy of the Random Forest model demonstrates its robustness and ability to handle the complexity of phishing detection. However, the work could not handle large dataset as there is a tendency for the model's performance to degrade. This suggests that while Random Forests are highly accurate in controlled scenarios, they may struggle with scalability.

A work by [11] introduced a "Next Generation Phishing Detection and Prevention System" using machine learning techniques. To achieve this, they developed models using Random Forests (RF), Decision Trees (DT), and Logistic Regression (LR). Additionally, they created an automatic Chrome plugin that acts as a one-stop solution by identifying and classifying URLs as either malicious or safe. Their findings revealed that the Random Forest classifier outperformed the other models, delivering the highest accuracy with fewer false negative results. To achieve the high accuracy results, they had to prune the Random Forest model, a process that was both complex and time-consuming.

[12], presented a paper, “Effective Phishing Emails Detection Method,” presents a robust approach to detecting phishing emails by leveraging advanced feature extraction and selection techniques. The core of their method lies in the use of the Term Frequency-Inverse Document Frequency (TF-IDF) scheme, which is a well-established technique for weighting features in text data. This approach effectively highlights the importance of specific terms in an email relative to their occurrence across a larger corpus, thus aiding in distinguishing phishing emails from legitimate ones. The Random Forest (RF) algorithm employed in their study achieved an impressive accuracy of 99.46%, which is particularly noteworthy given that it was applied to a validated dataset.

[13], “Phishing Attacks: A Recent Comprehensive Study and a New Anatomy” The research aim to evaluate phishing attacks by identifying the current state and reviewing existing phishing techniques. This anatomy provides a wider outlook for phishing attacks and provides an accurate definition covering end-to-end exclusion and realization of the attack. The study classified phishing attacks according to fundamental phishing mechanisms and countermeasures discarding the importance of the end-to-end lifecycle of phishing.

[14], “Using a Machine Learning Model for Malicious URL Type Detection” 2022- The research aim at developing an automatic detection of malicious URLs, using machine learning approaches. The method employs random forest and decision tree as core mechanisms and is evaluated on a combined benign and malicious URL dataset. The k-fold cross-validation technique was applied to the dataset and generated used a 30/70 ratio for the testing/training subsets. Selected feature sets applied on supervised classification on a ground truth dataset yields a classification accuracy of 97 % with a low false positive rate.

In [15], in his paper titled “Phishing Email Detection Based on Binary Search Feature Selection,” explores an innovative approach to detecting phishing emails using Binary Search Feature Selection (BSFS) in combination with a Pearson correlation coefficient algorithm for ranking features. The detection process itself relied on the BSFS algorithm, which has proven to be effective in many phishing detection studies. While more complex feature selection methods may enhance detection accuracy, they may also introduce practical challenges related to computational resources and processing time. All these necessitated the proposed Deep learning approach.

2.3. Deep Learning Approach

Deep learning is a subset of machine learning that uses multilayered neural networks, called deep neural networks, to simulate the complex decision-making power of the human brain, and some form of deep learning powers most of the artificial intelligence (AI) applications in our lives today [16]. Some examples of the deep learning system are image classification (CNN is used for such a task), Natural

 | DOI: 10.20533/jitst.2046.3723.2024.0108

Language Processing (GRU, LSTM and Transformers models are best for such a task), Autonomous Driving (CNN can be best used for such a task), Time-series Prediction (RNN, GRU and LSTM algorithms to predict future values on past temporal data), Robotics and Control Systems (Dee Q-networks, Policy Gradient Methods are used for such tasks), for Phishing detection systems, algorithms such as 1D-CNN, GRU, LSTM has over the time proving to be amongst the best deep learning models for prediction.

A work by [17] explores the use of Convolutional Neural Networks (CNN) for phishing detection. The study involved converting feature vectors into images, using a dataset of 1,353 real-world URLs with 10 features, sourced from the UCI Machine Learning Repository. The CNN model achieved a classification accuracy of 86.5%. Despite the small dataset, the 2D CNN showed suboptimal performance with an accuracy of 86.5%.

In [18] focused on developing three deep learning algorithms—CNN, Logistic Regression, and Linear Regression—for detecting phishing attacks. The researcher utilized 10,000 legitimate and 10,000 phishing websites from the PhishTank platform and developed the models using Amazon EMR. In terms of performance, Logistic Regression achieved the highest accuracy at 99.97%, followed by CNN at 98.00%, and Linear Regression at 96.62%.

2.4. Hybrid Approach

In [19] a hybrid deep learning approach was employed, CNN and LSTM models was use to detect spoofing website URLs in real-time applications. The study leveraged the combined strengths of CNN and LSTM for improved detection accuracy. The implementation was done using Keras, Tensorflow, Pandas, Numpy, and Scikit-learn. The hybrid CNN-LSTM model achieved superior performance, with 98.9% accuracy on the UCL dataset and 96.8% accuracy on the Phishing dataset. In comparison, standalone CNN and LSTM models yielded lower accuracies: CNN achieved 90.4% (UCL) and 89.3% (Phishing), while LSTM reached 94.6% (UCL) and 92.6% (Phishing). The datasets used were small, with the UCL dataset consisting of 20 rows and 31 columns, and the PhishTank dataset consisting of 10 rows and 112 columns.

[20], the study introduces a novel hybrid approach for phishing URL detection using Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU). The research explores three non-hybrid deep learning models (CNN 1D, LSTM, and GRU) and four hybrid models (GRU-LSTM, LSTM-LSTM, BI(GRU)-LSTM, and BI(LSTM)-LSTM). A dataset comprising 58,645 samples with 111 features was used. The BI(GRU)-LSTM hybrid model emerged as the best performer, achieving an accuracy of 93.91%, precision of 93.94%, recall of 93.38%, and an F1-Score of 93.66%. However, the model's accuracy declined with the validated datasets, indicating that it did not generalize well during training.

3. Methodology

A comprehensive review of the existing phishing detection system was carried out. The phishing detection methodology involves a series of processes, including collection of data sets, Pre-Processing, feature extraction methods for selecting the required set of features and deep learning DL classifiers for classifying the input data into defined data categories.

The data for the research was obtained from kaggle.com, Phishing_1 (small) and Phishing_legitimate_full (large) datasets was used to train and test the model. The large dataset contains 111 features from 88648 webpage instances while the small datasets contains 48 features extracted from 10001 webpage instances. Thereafter, the collected data was pre-processed using min-max method for data normalization to scale down values of phishing data within a specified range of zero (0) and one (1) as captured in equation (1):

$$v' = \frac{v - min_a}{max_a - min_a} \quad (1)$$

Where:

v' represents the normalized value, v denotes the observed value (that is, the value to be normalized), max_a and min_a are maximum and minimum values of attribute a respectively. After normalization, feature selection was applied on the normalized data to select relevant phishing data using information gain. The information is compiled by determining the entropy of the entire training data (T). This process involves computing the probability of the data with respect to the classes in the data as captured in equation (2) and (3):

$$P_i = \frac{|c_i, T|}{|T|} \quad (2)$$

$$E(T) = -\sum_{i=1}^{m} p_i log_2 p_i \quad (3)$$

Where:

T is the training set, p_i represents the probability that a sample in T belong to a distinct class c_i, $E(T)$ represents the entropy of T, and m represents the total number of distinct classes in T. In this study dropouts were used inside the deep learning models to reduce the overfitting by removing certain features randomly by making them zero. The reduced data feature set will be passed into the three (3) Deep Learning model concurrently for training and Testing.

 | DOI: 10.20533/jitst.2046.3723.2024.0108

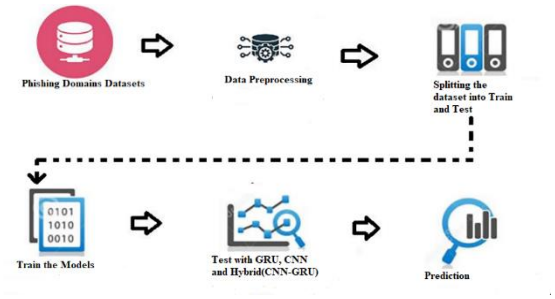

Figure 1. Proposed System Architecture [21]

3.1. Data Split

The process involves splitting the dataset into training and test set using splitting ratio 70:30. The training set was apportioned 70% of the dataset while the test set took the remaining 30%. Hence, for Phishing_1 instance count for training is 62,053, while that for testing is 26,595. For Phishing_Legitimate_full instance count for training is 7,000 was for training and 3,000 was used for testing.

a. Convolutional Neural Network (CNN)

CNN is a specialized type of deep learning neural network that is particularly well-suited for tasks involving grid-like data structures, such as images and video. 1D Convolutional Neural Network (CNN) was used for phishing domain classification, the first input was 1D CNN which was applied to effectively capture patterns and features from the sequential input data as represented in equation (4):

$$c_i = \text{ReLU}(\text{Conv1D}(e, w_i, b_i)) \quad (4)$$

Where:

e is the input, c_i is the output of the i-th convolutional layer, w_i and b_i are the convolutional filter weights and biases, respectively, and Conv1D performs the 1-dimensional convolution operation across the input sequence e.

b. Gated Recurrent Unit (GRU)

GRU is a type of recurrent neural network (RNN) architecture designed to model sequential data, much like the Long Short-Term Memory (LSTM) network. However, GRU offers a simpler and more computationally efficient architecture. While LSTM maintains a separate memory cell state that is updated through three gates (input, output, and forget gates), GRU streamlines this process by using only two gates: the reset gate and the update gate. The reset gate controls the amount of the previous hidden state to forget, and the update gate determines how much of the new information should be incorporated into the hidden state.

$$\hat{y} = \sigma(W_o \cdot h_T + b_o) \quad (5)$$

Where:

W_o and b_o are weights and biases of the output layer, after processing the entire sequence of features, the final hidden state h_T (where T is the sequence length) will be used as input to a fully connected layer followed by a sigmoid activation function (σ) to output the probability of the domain being phishing.

c. Hybrid CNN-GRU

Detecting phishing domains using a CNN-GRU (Convolutional Neural Network - Gated Recurrent Unit) involves combining the capabilities of both architectures to process sequential data effectively. The entire model is trained end-to-end using backpropagation and optimization techniques such as gradient descent (Adam). This approach involves using the shared layer of the two models. To achieve this using CNN for features extraction and the GRU models is used for classification.

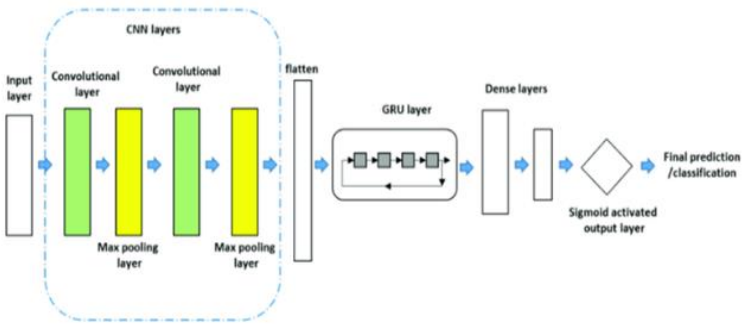

Figure 2. Hybridized CNN-GRU Architecture [22]

 | DOI: 10.20533/jitst.2046.3723.2024.0108

To achieve the above Model in Figure 2, the following steps are to be applied.

i. Input Layer

Let the input domain data be represented as a sequence of characters or words:

$$X = \{x1, x2, x3, \dots, xT\} \quad (6)$$

Where:

T is the length of the input sequence (e.g., the domain name).

ii. Embedding Layer

The input is passed through an embedding layer to transform it into a continuous vector space:

$$E(X) = \{e1, e2, e3, \dots, eT\} \quad (7)$$

Where:

$ei \in Rd$ is the embedded vector representation of the i-th input element, and d is the dimensionality of the embedding space.

iii. CNN Layers:

Convolutional Layer: The input to the convolutional layer is the embedded sequence E(X). The 1D convolution operation applies a filter W of size k (kernel size) to extract local features:

$$h_t = f(W \cdot E(xt: t + k) + b) \quad (8)$$

Where:

W is the convolution filter, b is the bias, f is a non-linear activation function (ReLU), and t:t+k represents a window of k elements.

Max Pooling Layer is applied to reduce the dimensionality and retain the most important features:

$$p_t = max(h_{t:t+m}) \quad (9)$$

Where:

m is the pooling size.

iv. Flatten

After the convolution and max-pooling layers, the resulting feature map is flattened into a 1D vector:

$$F = flatten(\{p1, p2, \dots, pn\}) \quad (10)$$

Where:

n is the number of features after pooling.

v. GRU Layer

The flattened output is passed through a GRU layer to capture sequential dependencies. In the GRU, the following operations occur at each time step t:

$$z_t = \sigma(W_z \cdot [ht - 1, xt]) \quad (11)$$

$$r_t = \sigma(W_r \cdot [ht\text{-}1, xt]) \quad (12)$$

$$\tilde{h}_t = tanh(W_h \cdot [rt \odot ht - 1, xt]) \quad (13)$$

$$ht = (1 - zt) \odot ht - 1 + zt \odot \tilde{h}_t \quad (14)$$

Here, z_t is the update gate, r_t is the reset gate, $\tilde{h}t$ is the candidate activation, and ht is the final hidden state.

vi. Dense Layers:

The GRU's final hidden state h_T is passed through one or more dense layers:

$$d = f(W_d \cdot h_T + b_d) \quad (15)$$

Where:

W_d is the weight matrix, b_d is the bias, and f is a non-linear activation function.

vii. Output Layer

The final dense layer output is passed through a sigmoid function to obtain a probability for binary classification (phishing or legitimate):

$$\tilde{y} = \sigma(W_o \cdot d + b_o) \quad (16)$$

where Wo is the weight matrix of the output layer, and σ is the sigmoid function:

3.2. Evaluation Metrics

The performance evaluation of the model was carried out based on the following metrics which are presented as follows:

$$ACC = \frac{TP+TN}{TP+TN+FP+FN} \quad (17)$$

$$Precision = \frac{TP}{TP+FP} \quad (18)$$

$$Recall = \frac{TP}{TP+FN} \quad (19)$$

$$F1 = \frac{2*(Recall * Precision)}{(Recall+Precision)} \quad (20)$$

Where:

- True positive (TP) represents an attack data detected as attack,

 | DOI: 10.20533/jitst.2046.3723.2024.0108

- True negative (TN) represents normal data detected as normal
- False positive (FP) represents normal data detected as an attack, and False Negative (FN) is denoted as an attack data detected as normal.

4. Results and Discussion

The training results determine the accuracy and loss during training of the models on 70% of the datasets apportioned for training the models to ascertain if the models are learning as shown in Figures 3, 4, and 5.

The loss started higher in the initial epochs and decreased over time, indicating that the model is learning and its predictions are becoming more accurate. The accuracy started lower and increased with each epoch, reflecting improvement in the model's performance.

```
Training with Phishing_Legitimate_full Dataset with GRU algorithm on 10 selected features
Epoch 1/10
219/219 [==============================] - 4s 6ms/step - loss: 0.6587 - accuracy: 0.5780
Epoch 2/10
219/219 [==============================] - 1s 7ms/step - loss: 0.3657 - accuracy: 0.8449
Epoch 3/10
219/219 [==============================] - 2s 8ms/step - loss: 0.3125 - accuracy: 0.8737
Epoch 4/10
219/219 [==============================] - 2s 8ms/step - loss: 0.2896 - accuracy: 0.8886
Epoch 5/10
219/219 [==============================] - 2s 8ms/step - loss: 0.2675 - accuracy: 0.8943
Epoch 6/10
219/219 [==============================] - 1s 6ms/step - loss: 0.2518 - accuracy: 0.9003
Epoch 7/10
219/219 [==============================] - 1s 6ms/step - loss: 0.2342 - accuracy: 0.9116
Epoch 8/10
219/219 [==============================] - 1s 6ms/step - loss: 0.2279 - accuracy: 0.9169
Epoch 9/10
219/219 [==============================] - 1s 6ms/step - loss: 0.2195 - accuracy: 0.9180
Epoch 10/10
219/219 [==============================] - 1s 6ms/step - loss: 0.2136 - accuracy: 0.9200
94/94 [==============================] - 1s 2ms/step
```

Figure 3. Training Result for GRU

```
Training with Phishing_Legitimate_full Dataset with CNN algorithm on 10 selected features
Epoch 1/10
219/219 [==============================] - 3s 7ms/step - loss: 0.4454 - accuracy: 0.7971
Epoch 2/10
219/219 [==============================] - 1s 6ms/step - loss: 0.3209 - accuracy: 0.8633
Epoch 3/10
219/219 [==============================] - 1s 6ms/step - loss: 0.2978 - accuracy: 0.8766
Epoch 4/10
219/219 [==============================] - 1s 5ms/step - loss: 0.2783 - accuracy: 0.8914
Epoch 5/10
219/219 [==============================] - 1s 5ms/step - loss: 0.2600 - accuracy: 0.8956
Epoch 6/10
219/219 [==============================] - 1s 4ms/step - loss: 0.2347 - accuracy: 0.9111
Epoch 7/10
219/219 [==============================] - 1s 4ms/step - loss: 0.2222 - accuracy: 0.9161
Epoch 8/10
219/219 [==============================] - 1s 5ms/step - loss: 0.2147 - accuracy: 0.9164
Epoch 9/10
219/219 [==============================] - 1s 5ms/step - loss: 0.2078 - accuracy: 0.9206
Epoch 10/10
219/219 [==============================] - 1s 4ms/step - loss: 0.2077 - accuracy: 0.9213
94/94 [==============================] - 0s 2ms/step
```

Figure 4. Training Result for CNN

```
Training with Phishing_Legitimate_full Dataset with CNN-GRU algorithm on 10 selected features
Epoch 1/10
219/219 [==============================] - 6s 7ms/step - loss: 0.4358 - accuracy: 0.7997
Epoch 2/10
219/219 [==============================] - 2s 7ms/step - loss: 0.3218 - accuracy: 0.8627
Epoch 3/10
219/219 [==============================] - 2s 7ms/step - loss: 0.3016 - accuracy: 0.8761
Epoch 4/10
219/219 [==============================] - 2s 7ms/step - loss: 0.2843 - accuracy: 0.8826
Epoch 5/10
219/219 [==============================] - 2s 7ms/step - loss: 0.2707 - accuracy: 0.8900
Epoch 6/10
219/219 [==============================] - 2s 7ms/step - loss: 0.2637 - accuracy: 0.8970
Epoch 7/10
219/219 [==============================] - 2s 8ms/step - loss: 0.2344 - accuracy: 0.9094
Epoch 8/10
219/219 [==============================] - 2s 9ms/step - loss: 0.2286 - accuracy: 0.9130
Epoch 9/10
219/219 [==============================] - 2s 8ms/step - loss: 0.2119 - accuracy: 0.9200
Epoch 10/10
219/219 [==============================] - 2s 7ms/step - loss: 0.2008 - accuracy: 0.9256
```

Figure 5. Training Result for CNN-GRU

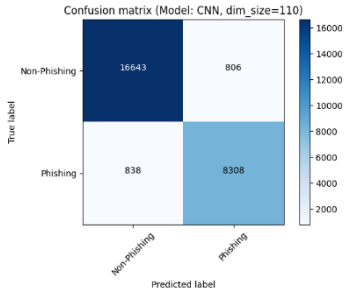

Figure 6. Confusion matrix for CNN on Phishing_Legitimate_full

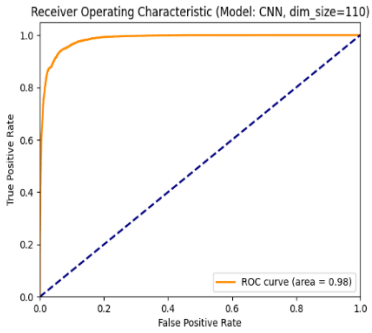

Figure 7. ROC Curve for CNN on Phishing_Legitimate_full

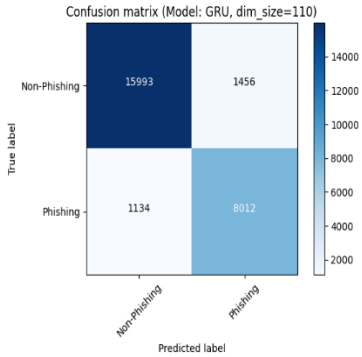

Figure 8. Confusion matrix for GRU on Phishing_Legitimate_full

4.1. Test Result

This section comprises analysis of the test result for the two datasets. Confusion matrix is a summary of predicted results on classification. Its shows the number of phishing and non-phishing, categorized

 | DOI: 10.20533/jitst.2046.3723.2024.0108

each class, making it useful for understanding performance of a classification. The ROC curve gives an understanding of the trade-offs between true positive and false positive.

a. Confusion matrix and ROC Curves for Phishing_legitimate_full

The Confusion Matrix and ROC curves for Phishing_legitimate_full Dataset is shown in Figure 6 to Figure 11, it is a graphical representation of the confusion matrixes and ROC curve whose result is summarized in Table 1.

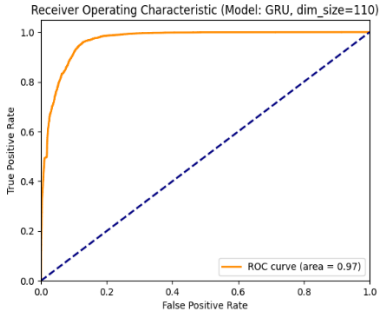

Figure 9. ROC curve for GRU on Phishing_Legitimate_full

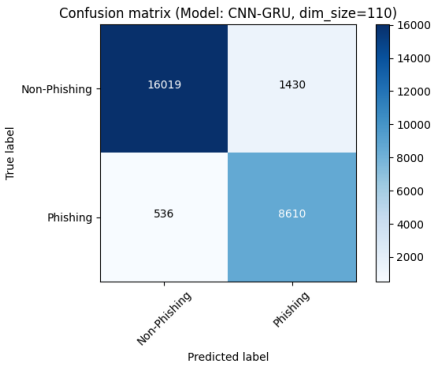

Figure 10. Confusion matrix for CNN-GRU on Phishing_Legitimate_full

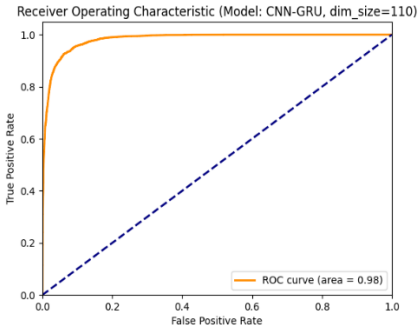

Figure 11. ROC curve for CNN-GRU on Phishing_Legitimate_full

b. Confusion matrix and ROC Curves for Phishing_1

The Confusion Matrix and ROC curves for Phishing_1 Dataset are shown in Figures 12 to 17. It shows how the confusion matrixes and ROC were used for evaluation of the models and the ROC curve shows the graph of true positive against true negative. The result is summarized in Table 1.

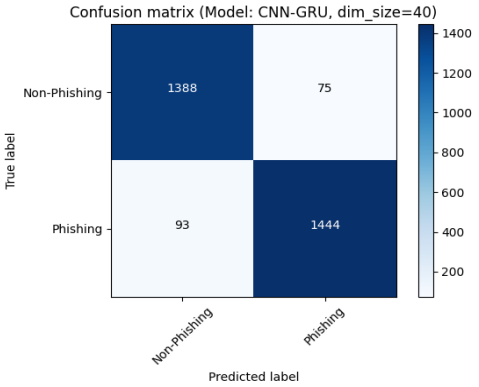

Figure 12. Confusion Matrix on CNN for Phishing_1

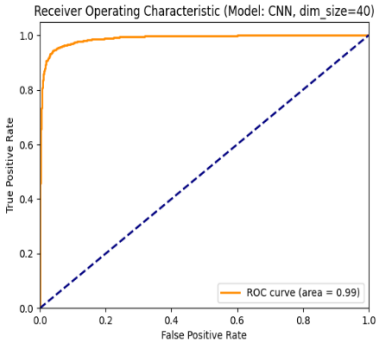

Figure 13. ROC for CNN on Phishing_1

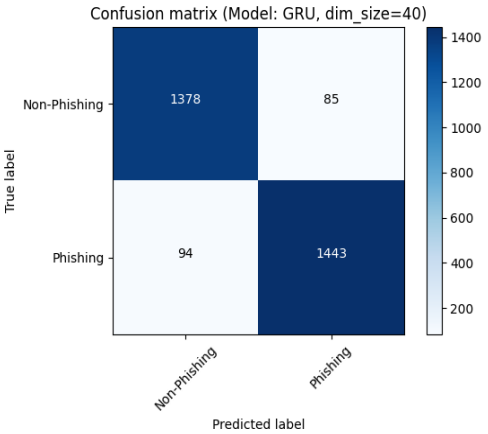

Figure 14. Confusion Matrix on GRU for Phishing_1

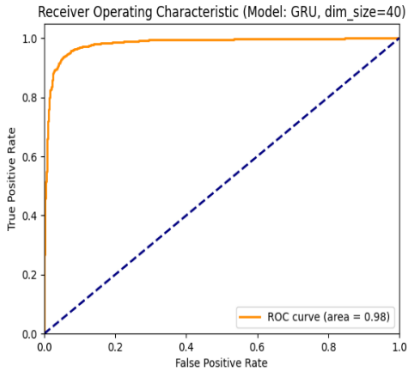

Figure 15. Confusion Matrix on GRU for Phishing_1

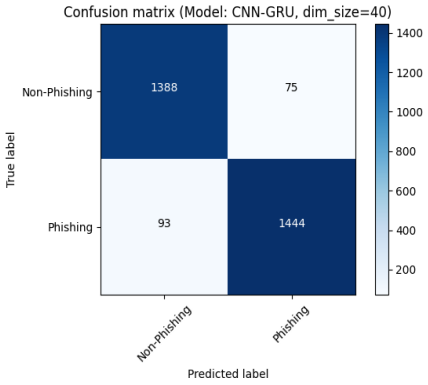

Figure 16. Confusion Matrix for CNN-GRU on Phishing_1

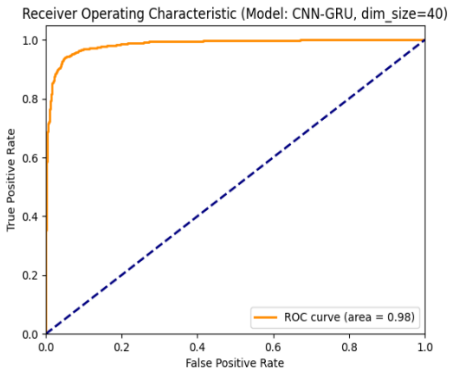

Figure 17. ROC for CNN-GRU on Phishing_1

5. Discussion and Result Comparison

All the results were summarized in Table 1. Figure 18 is a tabular representation of the results that were gotten from both datasets using the appropriate evaluation techniques.

Table 1 shows a tabular representation of the results from both datasets using the appropriate evaluation techniques.

Table 1. Test Result from the two Datasets

Datasets	Model	DR	RECALL	F1-SCORE	ACCURACY
Phishing_1 from Kaggle.com	CNN	0.9035	0.9746	0.9377	0.9337
	GRU	0.9444	0.9388	0.9416	0.9402
	CNN-GRU	0.9506	0.9395	0.9450	0.9440
Phishing_Legitimate_full from Kaggle.com	CNN	0.9116	0.9084	0.9100	0.9382
	GRU	0.8462	0.8760	0.8609	0.9026
	CNN-GRU	0.8576	0.9414	0.8975	0.9261

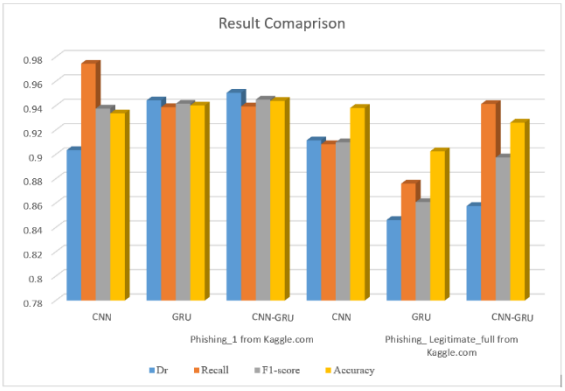

Figure 18. Graph for Test Result Comparison

5.1. Test Results

The results present the performance of three models (CNN, GRU, and CNN-GRU) across two phishing datasets from Kaggle: Phishing_1 and Phishing_Legitimate_full. The metrics used for evaluation are Detection Rate (DR), Recall, F1-Score, and Accuracy.

Phishing_1 Dataset

CNN Model: This model exhibits a high Recall (0.9746), which means it performs well in identifying true positive phishing URLs of phishing domains. The F1-Score of 0.9377 and an Accuracy of 0.9337 indicate that the CNN model balances precision and recall effectively, suggesting that the model is robust in distinguishing between phishing and legitimate URLs.

GRU Model: The GRU model shows a Detection Rate (DR) of 0.9444, which is slightly higher than CNN's 0.9035. However, the Recall (0.9388) is marginally lower than CNN. With an F1-Score of 0.9416 and an Accuracy of 0.9402, the GRU model performs slightly better overall than CNN in terms of Accuracy and F1-Score, showcasing its ability to effectively model sequential data.

CNN-GRU Hybrid Model: The CNN-GRU model combines the strengths of both CNN and GRU, yielding the best overall performance in this dataset. It achieves the highest Detection Rate (0.9506), F1-Score (0.9450), and Accuracy (0.9440).

 | DOI: 10.20533/jitst.2046.3723.2024.0108

This indicates that the hybrid model effectively captures both spatial features (through CNN) and sequential dependencies (through GRU), making it more effective for phishing URL detection in the Phishing_1 dataset.

Phishing_Legitimate_full Dataset

- CNN Model: The CNN model shows strong performance in this dataset as well, with an Accuracy of 0.9382 and an F1-Score of 0.9100. However, compared to the Phishing_1 dataset, its Recall is lower (0.9084), suggesting it is slightly less effective at identifying true positive phishing URLs in this dataset.
- GRU Model: The GRU model shows a decrease in performance here, with a DR of 0.8462 and a Recall of 0.8760. The Accuracy drops to 0.9026, which is significantly lower compared to its performance in the Phishing_1 dataset. This suggests that the GRU model struggles more with this dataset, likely due to its complexity or differences in data distribution.
- CNN-GRU Hybrid Model: In this dataset, the CNN-GRU hybrid model continues to outperform GRU but not CNN. The hybrid model achieves a Recall of 0.9414, which is the highest among all models for this dataset, indicating its strong ability to identify phishing URLs. The Accuracy (0.9261) and F1-Score (0.8975) are lower than CNN's, but it still strikes a balance between precision and recall.

The CNN-GRU hybrid model consistently outperforms the individual CNN and GRU models in the Phishing_1 dataset, indicating that combining convolutional and recurrent layers offers better performance for phishing URL detection. This is likely due to the hybrid model's ability to capture both spatial features and sequential dependencies.

In the Phishing_Legitimate_full dataset, while CNN shows strong performance, the hybrid model's higher Recall (0.9414) suggests that it excels in phishing detection, though at the cost of a slightly lower F1-Score and Accuracy compared to CNN.

The GRU model performs notably worse in the Phishing_Legitimate_full dataset compared to Phishing_1, which may indicate that it struggles with more complex or diverse datasets.

The results indicate that while CNN and GRU models perform well individually, the CNN-GRU hybrid model provides superior performance in detecting phishing Domains, particularly in datasets like Phishing_1 where both spatial and temporal information is crucial. However, the performance difference in the Phishing_Legitimate_full dataset suggests that dataset characteristics can significantly influence the models' effectiveness, and further tuning may be required to optimize model performance across different datasets.

5. Conclusion

As phishing techniques become increasingly sophisticated, the combination of deep learning and cybersecurity will be crucial in staying ahead of cybercriminals and protecting critical digital assets from exploitation. The results form the two data sets show that standalone models is more efficient in large data sets while the Hybridized model had a noticeable advantage of the standalone models when small data is involved.

6. Future work

To enhance more detection systems essenbling approaches can be applied and also deep learning models, such as GRUs and CNNs, usually operates as black boxes, making it difficult to explain why a particular domain or email was flagged as phishing, explain ability tools such as LIME (Local Interpretable Model-agnostic Explanations) or SHAP (Shapley Additive explanations) can be integrated to provide insights into which features or parts of the domain name/URL influenced the model's decision.

7. References

[1] Mathura, K., and Arunkumar, V. (2023). Cyber security in the age of digitalization. International Journal of Computer Applications, 178(5), 12–17.

[2] Pinki, A. (2019). Digital media and its effects on society. Journal of Digital Media Studies, 9(2), 45–53.

[3] Shankhwar, P. (2023). Hazards of over-reliance on digital technologies. Journal of Cyber Psychology, 7(1), 22–29.

[4] Shweta, R., Chaudhary, R., and Bhattacharya, A. (2021). Understanding the dual nature of digital technologies. International Journal of Information Technology, 13(2), 100–110.

[5] Saqib, M., Ali, M., and Gupta, R. (2023). Vulnerabilities in the age of IoT: An analysis. Journal of Internet Security, 15(3), 150–158.

[6] Andy, T. (2023). Cybersecurity in emerging technologies. Tech Insights, 8(4), 45–52.

[7] Prenali, A., and Soni, R. (2023). Cyber threats to everyday users. Cybersecurity Journal, 6(1), 70–78.

[8] Alessandro, L. (2020). Social engineering in cyber attacks. Cybersecurity Review, 11(2), 102–107.

[9] Apandi, S. H., Sallim, J., and Sidek, R. (2020). Types of anti-phishing solutions for phishing attack. IOP

 | DOI: 10.20533/jitst.2046.3723.2024.0108

Conference Series: Materials Science and Engineering, 769(1), 012072. DOI: 10.1088/1757-899X/769/1/012072.

[10] Alnemari, S., and Alshammari, M. (2023). Detecting phishing domains using machine learning. Applied Sciences, 13(8), 4649. DOI: 10.3390/app13084649.

[11] Tyagi, S., Tyagi, R. K., Dutta, P. K., and Dubey, P. (2023). Next generation phishing detection and prevention system using machine learning. In 2023 1st International Conference on Advanced Innovations in Smart Cities (ICAISC).IEEE, DOI: 10.1109/ICAISC56366.2023.100 85529.

[12] Dalia, M., Rahman, H., and Iftikhar, M. (2021). Detecting malicious URLs using machine learning techniques: Review and research directions. ACM Computing Surveys, 54(6), 1-35. DOI: 10.1145/3447945.

[13] Zainab, A., Sadiq, M., and Babar, M. (2021). Phishing attacks: A recent comprehensive study and a new anatomy. Journal of Computer Networks and Communications, 2021, Article ID 8843635. DOI: 10.1155/2021/8843635.

[14] Suet, P., Lim, Y., and Chua, Y. (2022). Using a machine learning model for malicious URL type detection. Journal of Network and Computer Applications, 205, 103317. DOI: 10.1016/j.jnca.2022.103 317.

[15] Sonowal, G. (2020). Phishing email detection based on binary search feature selection. SN Computer Science, 1(4). DOI: 10.1007/s42979-020-00194-z.

[16] Kulkarni, A. D. (2022). Convolution neural networks for phishing detection. International Journal of Advanced Computer Science and Applications, 14(4). DOI: 10.14569/IJACSA.2023.0140403.

[17] Sabah, A. (2021). Phishing attack detection using deep learning. International Journal of Cyber Security, 13(1), 99–107.

[18] Dilhara, S. (2021). Phishing URL detection: A novel hybrid approach using long short-term memory and gated recurrent units. International Journal of Computer Applications, 183(44), 41–54. DOI: 10.5120/ijca202192 1859.

[19] Ujah-Ogbuagu, U., et al. (2024). A hybrid deep learning technique for spoofing website URL detection in real-time applications. IEEE Access, 12(4), 123-135.

[20] Dalia, M., and Bhandari, D. (2021). Effective phishing emails detection method. International Journal of Computer Applications, 182(2), 25-30. DOI: 10.5120/ ijca2021922153.

[21] Kalla, D., and Chandrasekaran, A., (2023). Heart disease prediction using chi-square test and linear regression. Department of Computer Science, Colorado Technical University, Colorado, USA. DOI: 10.5121/csit. 2023.130712.

[22] Yerima, S. Y., Alzaylaee, M. K., Shajan, A., and P, V. (2021). Deep Learning Techniques for Android Botnet Detection. Electronics, 10(4), 519. https://doi.org/10.3390/ electronics10040519.

 | DOI: 10.20533/jitst.2046.3723.2024.0108