Enhancing Cybersecurity through LSTM-Based Phishing URL Detection

T.J. Ayo
The Federal University of Technology

O.D. Alowolodu
The Federal University of Technology

D.P. Omotayo
Bowie State University, 14,000 Jericho Park Road Bowie Maryland 20715

ABSTRACT

Attackers over the years has frequently launch attacks on users of the internet in other to steal their personal vital information so as to achieve their fraudulent acts. To curb this attacks there is a need to develop a deep learning model that can conveniently detect URL that are phishing. Long short time memory (LSTM) model is used in this research, various approaches haven used but best result is yet to be preferred the approaches. LSTM is a variation of Recurrent Neural Network (RNN) Architecture designed to handle sequence related prediction problems and its does well when working on sequential data, such as speech recognition, natural language processing and Phishing detection. The two datasets from kaggle.com were used and the result shows that on Phishing_1 dataset (large dataset) has an accuracy of 0.8672 while on the second dataset Phishing_Legitimate_full (small dataset) has an accuracy of 0.9043 this therefore mean that LSTM can perform better on small datasets and there is tendency of result degradation on larger datasets. However, there are other metrics that makes LSTM a considerable model on larger datasets like the F1-score.

Keywords

Deep Learning; LSTM; Phishing URL; URLs Detection.

1. INTRODUCTION

Cyber-crimes, including financial scams, identity thefts, and the installation of malware, are often executed via malicious URLs. These URLs systematically gather personal information from users, leading to severe financial and psychological consequences for the victims. Given the increasing sophistication of these attacks, there is a critical need for robust techniques to detect and block malicious URLs, thereby reducing the incidence of such scams. These techniques are essential to protecting users from being blackmailed and suffering significant losses [1].

Cyber-attacks such as financial scams, identity thefts, and the installation of malware, are often executed via malicious URLs. These URLs systematically gather personal information from users, leading to severe financial and psychological consequences for the victims. Given the increasing sophistication of these attacks, there is a critical need for robust techniques to detect and block malicious URLs, thereby reducing the incidence of such scams. These techniques are essential to protecting users from being blackmailed and suffering significant losses [2].

Phishing attacks seek to trick recipients into believing that an email is legitimate, in order to solicit sensitive information (e.g., usernames, passwords, and credit card numbers) or install malware, as a result [3], phishing is a fundamental component of many cyber-attacks and is often used as a first step inadvanced persistent threats [4]. Due to the cyber-attacks that are daily encountered by users and organizations globally there is a need to develop reliable model that can curb this attacks.

2. LITERATURE REVIEW

The rate at which surfing of the internet has increase had made a lot of attacks possible. The phishing URLs is one of the most cyber-attacks to steal vital information from users. Researchers has employed various methods to curb this attacks of which we have the traditional methods and conventional machine learning approaches.

2.1 Traditional Approach

2.1.1 Blacklist-Based Methods

Blacklist-based methods block a URL by referencing pre-compiled lists of known phishing URLs. The methods have simple implementations, with very low false positives and being computationally efficient. The disadvantages of blacklists are that they tend to quickly get outdated, with the new phishing URLs coming out each day. They fail miserably against newly created phishing URLs or those utilizing dynamic URL generation techniques. Some of the notable examples include Google Safe Browsing and Microsoft SmartScreen, which are blacklist-based URL filtering.

2.1.2 Heuristic-Based Methods

Heuristic-based methods analyze the URL’s features (e.g., URL length, presence of special characters, subdomain count) and other webpage characteristics to identify potential phishing sites. The advantage of Heuristic approaches is that it can detect previously unseen phishing URLs by leveraging patterns common to phishing attempts. The noticeable limitation was that these methods may have a higher false positive rate because legitimate URLs may also match some of the heuristic patterns. An example of such is Checking for unusual port numbers, detecting URL obfuscation (e.g., using "0" instead of "O" or "1" instead of "I"), and identifying strange domain names [5].

2.2 Machine Learning Approach

Despite the ability of machine learning to learn patterns on data and accurately predicting outcome, this machine learning approach has become better choice over the traditional approaches, machine learning uses features converted to vectors by features selection method to extract the features with most information gain.

A research by [6] in a paper titled “An Emerging Solution for Detection of Phishing Attacks,” explores the application of machine learning algorithms to differentiate between phishing emails and genuine emails. The researcher specifically employed the J48 classification algorithm, a type of decision tree, to perform this classification task. The results were promising, with the model achieving a classification accuracy of 98%, indicating a high level of effectiveness in distinguishing between fake and legitimate emails.

All these necessitated the proposed Deep learning approach.

2.3 Deep Learning Approach

Deep learning (DL) is one of the fastest-growing topics in materials data science, with rapidly emerging applications spanning atomistic, image-based, spectral, and textual data modalities. DL allows analysis of unstructured data and automated identification of features. The recent development of large materials databases has fueled the application of DL methods in atomistic prediction in particular. In contrast, advances in image and spectral data have largely leveraged synthetic data enabled by high-quality forward models as well as by generative unsupervised DL methods [7].

A research by [8] tittled "From Phishing Behavior Analysis and Feature Selection to Enhance Prediction Rate in Phishing Detection" The objective of this research is to develop an URL phishing detection model that can demonstrate its robustness against constantly changing attacks. Eleven machine learning classification techniques are utilized for classification tasks and comparative objectives. Moreover, three datasets with different instance distributions were constructed at different times for the model's initial construction and evaluation. Several experiments were carried out to investigate and evaluate the proposed model's performance, effectiveness, and robustness. The analysis shows that LGMB has the highest accuracy. Algorithm Accuracy RF 99.72, GBoost 99.72, LGBM 99.73, SVM 96.98, LR 96.62, KNN 99.59, GaussianNB 98.38, CatBoost 96.85, DT 96.98, LDA 96.21, QDA 84.7

[9] 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. These URLs were categorized as legitimate, suspicious, or phishing. A simple CNN was developed in MATLAB to classify the image representations of the feature vectors. The CNN model achieved a classification accuracy of 86.5%.

3. METHODOLOGY

Literatures were review to know their weakness and strengths of the existing phishing detection or prevention approaches. To achieved a better phishing detection model there are procedures that has to be followed: Datasets are collected from kaggle.com labelled Phishing_1 (small) containing 10001 domain instances with 111 features and the Phishing_Legitimate_full (large) containing 88648 with 48 features extracted. The data collected is preprocessed using min-max approach for data normalization to scale values of phishing data with a vector range of zero (0) and one (1) as shown in mathematical expression in equation (1)

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

where v' is the normalized value, v represents the value being observed, max_a and min_a are maximum and minimum values of attribute a respectively. Feature selection is then applied to already normalized data to select the relevant phishing URLs by the use of Information gain (IG). The compilation of the information is done by determining the entropy of the whole training data (T). This method involves computing the probability of the data with respect to the classes in the data as shown in equation (2) and (3)

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

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

The 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 feature sets are then passed into the Deep Learning model concurrently for training and testing.

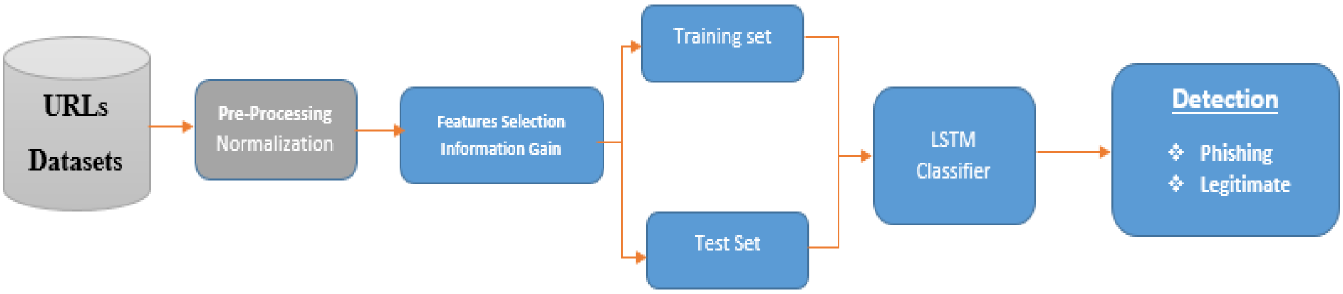


Figure 1: Proposed System Architecture

3.1 Data Split

The dataset was splinted in a ratio of 70:30. The 70% was used for training while the 30% was used for testing. For Phishing_1 had 62,5053 for training and 26,595 for testing. Phishing_Legitimate_full had 7,000 for training the model while 3,000 was used for testing.

3.2 Long Short-Term Memory (LSTM) Model

LSTM networks are specifically develop to address the issue of long-term dependencies, where a standard RNN may struggle to retain information over many time steps due to problems like the vanishing gradient the help overcome overfitting problem. LSTM networks achieve this with a structure that allows them to remember information for longer periods.

There are three gates that make up the LSTM architecture they are; Input gate, Forget gate and the Output gate.

3.2.1 Forget Gate

The forget gate decides which information from the previous cell state $Ct - 1$ should be retained or discarded based on the current input xt and previous hidden state $ht - 1$.

$$ft = \sigma(Wf \cdot [ht - 1, xt] + bf) \qquad (4)$$

where:

Wf is the weight matrix for the forget gate, bf is the bias for the forget gate, σ is the sigmoid activation function, outputting values between 0 and 1 to indicate how much information to keep (close to 1) or forget (close to 0).

3.2.2 Input Gate

The input gate determines which new information from the current input should be added to the cell state.

$$it = \sigma(Wi \cdot [ht - 1, xt] + bi) \qquad (5)$$

where:
Wi and $bi$are the weights and biases for the input gate.

it values close to 1 allow more information to be added to the cell state.

3.2.3 Output Gate

The output gate determines the next hidden state

ht (which is also the LSTM output for the time step) based on the updated cell state $\boldsymbol{Ct}$.

$$ot = \sigma(Wo \cdot [ht - 1, xt] + bo)) \qquad (6)$$

where:

wo and $bo$are the weights and biases for the output gate. ht represents the hidden state passed on to the next time step.

3.3 Loss Function and Optimization

The LSTM model is typically trained using a classification loss function, such as *binary cross-entropy*, to distinguish between phishing (label 1) and legitimate URLs (label 0):

$$loss = -\frac{1}{N}\sum_{t-1}^{N}\left[y_{i \cdot Log(\hat{y}) + Log}(1 - y_i) \cdot \log(1 - \hat{y}_i)\right] \qquad (7)$$

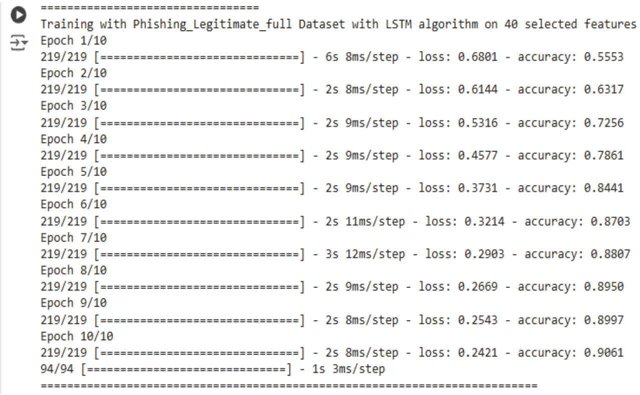
where:

yi is the true label for the i-th URL. $\hat{y}$ is the predicted probability for the phishing class.

4. RESULTS AND DISCUSSION

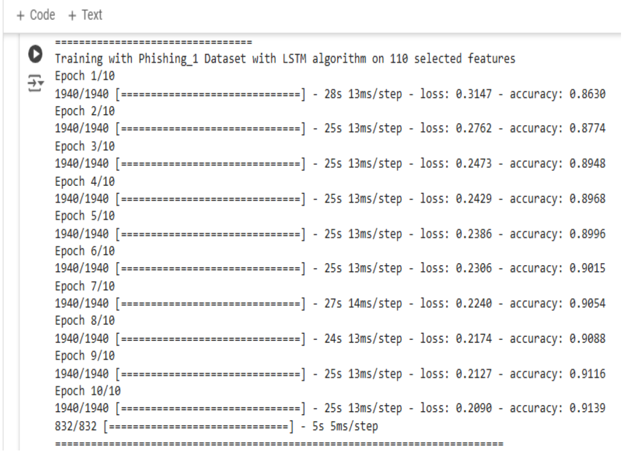
4.1 Training Results

The performance of model during training is determined by the loss and accuracy in each epoch. If the loss is high at the initial stage and decreases during the training it means the model is learning. Also if the accuracy on initial epoch start low and increases in the subsequent epochs its indicate that the model is learning.



```
=================================
Training with Phishing_Legitimate_full Dataset with LSTM algorithm on 40 selected features
Epoch 1/10
219/219 [==============================] - 6s 8ms/step - loss: 0.6801 - accuracy: 0.5553
Epoch 2/10
219/219 [==============================] - 2s 8ms/step - loss: 0.6144 - accuracy: 0.6317
Epoch 3/10
219/219 [==============================] - 2s 9ms/step - loss: 0.5316 - accuracy: 0.7256
Epoch 4/10
219/219 [==============================] - 2s 9ms/step - loss: 0.4577 - accuracy: 0.7861
Epoch 5/10
219/219 [==============================] - 2s 9ms/step - loss: 0.3731 - accuracy: 0.8441
Epoch 6/10
219/219 [==============================] - 2s 11ms/step - loss: 0.3214 - accuracy: 0.8703
Epoch 7/10
219/219 [==============================] - 3s 12ms/step - loss: 0.2903 - accuracy: 0.8807
Epoch 8/10
219/219 [==============================] - 2s 9ms/step - loss: 0.2669 - accuracy: 0.8950
Epoch 9/10
219/219 [==============================] - 2s 8ms/step - loss: 0.2543 - accuracy: 0.8997
Epoch 10/10
219/219 [==============================] - 2s 8ms/step - loss: 0.2421 - accuracy: 0.9061
94/94 [==============================] - 1s 3ms/step
==========================================================================
```

Figure 2: Training Result For Phishing_1



```
=================================
Training with Phishing_1 Dataset with LSTM algorithm on 110 selected features
Epoch 1/10
1940/1940 [==============================] - 28s 13ms/step - loss: 0.3147 - accuracy: 0.8630
Epoch 2/10
1940/1940 [==============================] - 25s 13ms/step - loss: 0.2762 - accuracy: 0.8774
Epoch 3/10
1940/1940 [==============================] - 25s 13ms/step - loss: 0.2473 - accuracy: 0.8948
Epoch 4/10
1940/1940 [==============================] - 25s 13ms/step - loss: 0.2429 - accuracy: 0.8968
Epoch 5/10
1940/1940 [==============================] - 25s 13ms/step - loss: 0.2386 - accuracy: 0.8996
Epoch 6/10
1940/1940 [==============================] - 25s 13ms/step - loss: 0.2306 - accuracy: 0.9015
Epoch 7/10
1940/1940 [==============================] - 27s 14ms/step - loss: 0.2240 - accuracy: 0.9054
Epoch 8/10
1940/1940 [==============================] - 24s 13ms/step - loss: 0.2174 - accuracy: 0.9088
Epoch 9/10
1940/1940 [==============================] - 25s 13ms/step - loss: 0.2127 - accuracy: 0.9116
Epoch 10/10
1940/1940 [==============================] - 25s 13ms/step - loss: 0.2090 - accuracy: 0.9139
832/832 [==============================] - 5s 5ms/step
==========================================================================
```

Figure 3: Training Result for Phishing_Legitimate full

4.2 Test Results

This section comprises analysis of the test result for the two datasets.

Confusion matrix summarizes the results predicted on classification. It makes the performance more understandable by showing the number of phishing and legitimate categorized in each class. Figure 4 and figure 6 were used to compute the evaluation result in table 1 below.

The ROC curve shows an understanding of the trade-offs between true positive and false positive.

Confusion matrixes and ROC Curves for Phishing for the two datasets are shown in figure 5 and figure 7 below. The ROC curve area was 0.96 respectively which was a metric that the model trained very well.

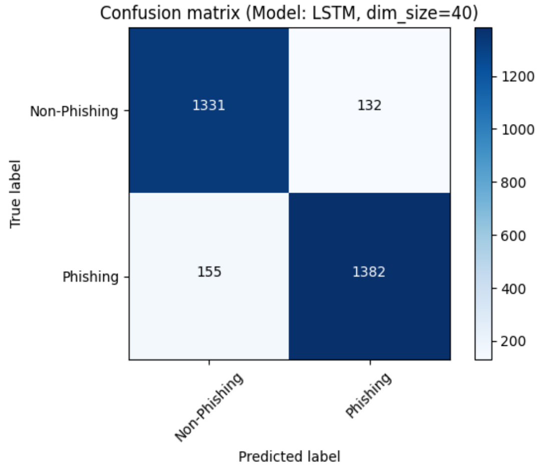


Figure 4: Confusion matrix on Phishing_1

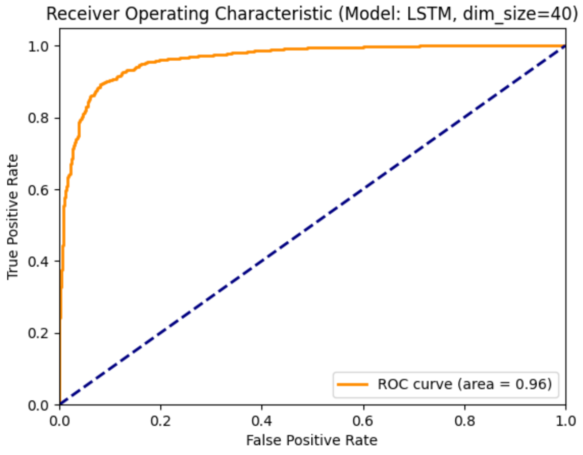


Figure 5: ROC Curve for CNN on Phishing_Legitimate_full

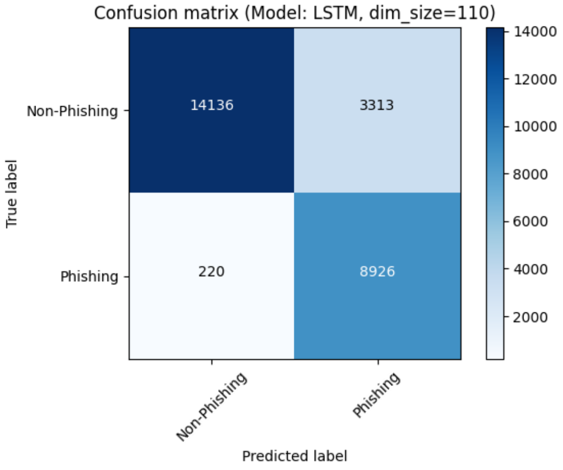


Figure 6: Confusion matrix for Phishing Legitimate full

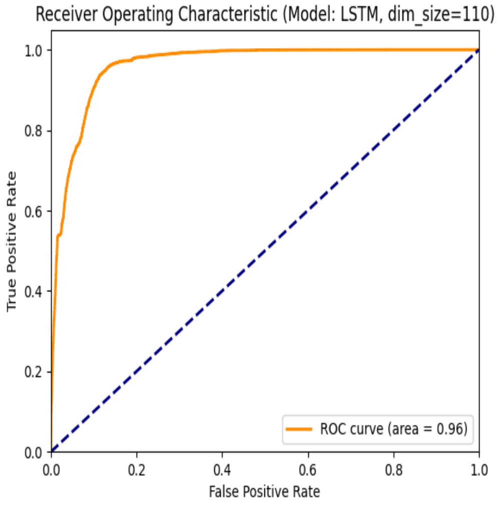


Figure 7: ROC curve for Phishing_Legitimate full

4.3 Test Results for Phishing_1

Table 1: Test Result for Phishing_1 Datasets

	DR	Recall	F1-Score	Accuracy
10 features	0.8390	0.9258	0.8803	0.8710
20 features	0.9054	0.9401	0.9224	0.9190
20 features	0.8464	0.9610	0.9001	0.8907
40 features	0.9128	0.8992	0.9059	0.9043

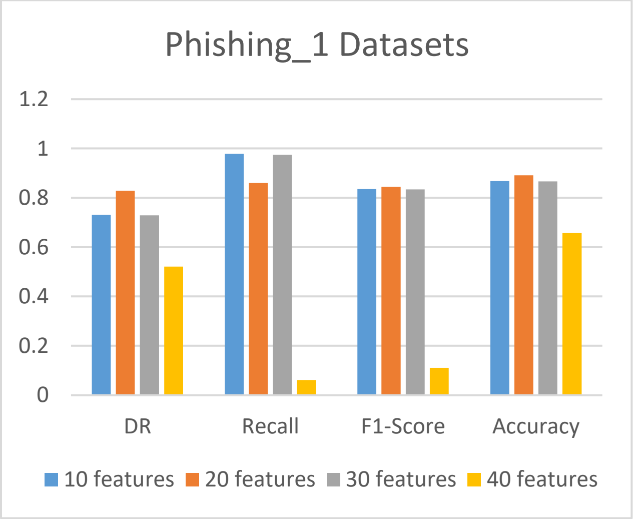


Figure 8: Graph for Test Result Phishing_1

4.4 Discussion of Test Results for Phishing_1

Feature Count and Performance: The performance of the model improves as the number of features increases, as evidenced by the rise in Detection Rate and Accuracy with 20 and 40 features. However, there is a noticeable trade-off between Recall and Detection Rate with the increase in features. Initially, with more features, the model becomes better at detecting phishing URLs (higher DR), but this comes at the expense of some Recall, particularly in the second instance with 20 features and the 40-feature set.

Recall vs. Detection Rate: The significant increase in Recall in the second 20-feature set (0.9610) suggests that the model is highly sensitive, but this leads to a decrease in Detection Rate (0.8464). A higher Recall implies fewer phishing URLs are missed (fewer false negatives), but the model might be classifying more legitimate URLs as phishing (increasing false positives). This trade-off between precision and recall is common in classification tasks, and here, it reflects the model's sensitivity to detecting phishing at the cost of being overly cautious.

F1-Score Balance: The F1-Score is highest with the 20-feature set (1st instance), reflecting a good balance between precision and recall. It indicates that this model is effective at classifying both phishing and legitimate URLs correctly without leaning too heavily towards one at the expense of the other.

The trend of increasing features leads to better Detection Rates and Accuracy, but the Recall is slightly impacted. The results suggest that the model is most effective with 20 features, striking a good balance between all metrics. However, with 40 features, while the Accuracy improves, the slight drop in Recall could indicate that the model is becoming more conservative in predicting phishing URLs, possibly sacrificing some sensitivity.

Table 2: Test Result for Phishing Legitimatefull Datasets

	DR	Recall	F1-Score	Accuracy
10 features	0.7320	0.9790	0.8351	0.8679
20 features	0.8297	0.8601	0.8446	0.8912
30 features	0.7292	0.9752	0.8345	0.8670
40 features	0.5211	0.0622	0.1111	0.6578
50 features	0.9003	0.7728	0.8317	0.8924
60 features	0.7940	0.9518	0.8697	0.8981
70 features	0.8006	0.9518	0.8697	0.9019
80 features	0.7421	0.9733	0.8421	0.8745
90 features	0.8523	0.8715	0.8618	0.9039
100 features	0.8671	0.8459	0.8563	0.9024
110 features	0.7293	0.9759	0.8348	0.8672

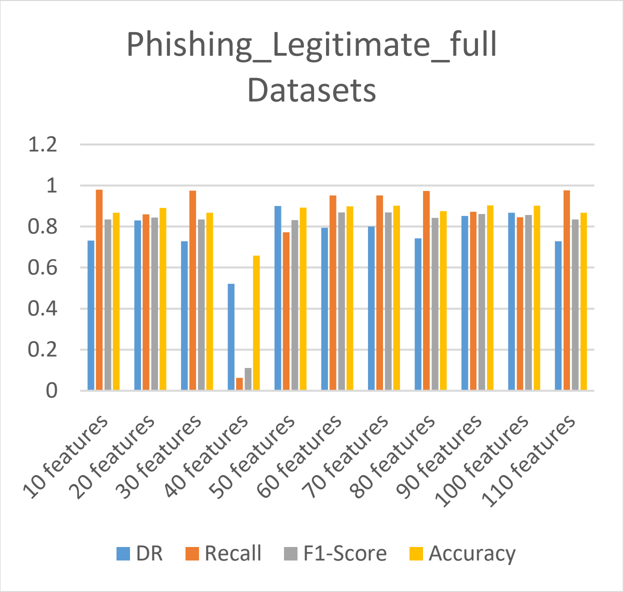


Figure 9: Graph for Test Result Phishing Legitimate full

4.5 Discussion of Test Results for Phishing Legitimate full Datasets

Result in table 2: is discussed below;

Feature Selection and Model Performance:

The performance of the model is highly dependent on the number of features selected. A small number of features (10-30) tends to result in high Recall but lower Detection Rate and Accuracy, indicating that the model is highly sensitive but lacks specificity.

As the number of features increases (50-70), there is an improvement in Detection Rate and Accuracy, with Recall remaining reasonably high. This suggests that increasing the feature set leads to better generalization and reduces false positives.

However, with more than 70 features (80-110), the model's performance begins to degrade. Recall drops significantly in some cases, and the Detection Rate either stagnates or decreases. This suggests that overfitting may be occurring with the larger feature sets, where the model becomes too complex and starts to misclassify or miss certain phishing URLs.

Optimal Feature Range:

Based on the results, the model performs best with 70 features, which provides a good balance between Recall, Detection Rate, and Accuracy. Beyond this point, the addition of more features does not result in significant improvements and might even harm performance due to overfitting.

Trade-off Between Sensitivity and Specificity:

The results show a classic trade-off between Recall (sensitivity) and Detection Rate (specificity). As Recall increases, Detection Rate tends to decrease, and vice versa. The goal should be to find an optimal balance where the model detects phishing URLs without excessively misclassifying legitimate ones.

5. CONCLUSION

In conclusion, there is a high chance of cyber-attacks, which includes phishing through malicious URLs. This is really very threatening in terms of online security. As it was revealed in this research, an LSTM model, a variant of RNN, can serve as an effective tool in detecting phishing URLs. The LSTM model works well with sequential data, and therefore, it is very appropriate for phishing detection tasks. This can be further assured from the performance of the model on two datasets from Kaggle. While the model achieved an accuracy of 0.8672 on the larger Phishing_1 dataset, it performed better on the smaller Phishing_Legitimate_full dataset with an accuracy of 0.9043. This may indicate that although LSTM tends to work great on small datasets, performance may drop on larger datasets due to overfitting or simply because the data is much more complex. Yet, an F1-score and other similar metrics reinforce the potential of the model for larger data, making it useful for real-world applications. In summary, this study highlights the need to enhance and optimize deep learning models such as LSTM in enhancing phishing detection to protect users against increasingly sophisticated cyber threats.

6. FUTURE WORK

Future direction can be geared towards other Deep Learning models and also considering ensemble approaches for detecting phishing URLs.

7. REFERENCES

[1] Alkhalil, Z., Nawaf, L., & Khan, I. 2021. Phishing attacks: A recent comprehensive study and a new anatomy.

Frontiers in Computer Science, 3, Article 563060. https://doi.org/10.3389/fcomp.2021.563060.

[2] Zanab, S., Jamil, K., and Khan, H. 2021. Identity theft and phishing attacks in the digital age: An empirical study. *Journal of Cybersecurity Studies*, 9(2), 48-60.

[3] Burita. (2021). The growing threat of phishing in a connected world. *Journal of Information Security and Cyber Crime, 22*(3), 85-100.

[4] Singer, P., and Friedman, A. 2014. *Cybersecurity and cyberwar: What everyone needs to know*. Oxford University Press.

[5] Reddy, Kalyani S., and D. Sasikala 2017. "Blacklist-Based Techniques for Detection of Phishing Attacks: A Survey." *International Journal of Engineering and Technology*.

[6] Prasanta, B. 2021. An emerging solution for detection of phishing attacks. *International Journal of Computer Applications,181*(6),11-16.doi.org/10.5120/ijca 2021921313.

[7] Kamal, H., Stewart, A., and Rehn, M. 2022. The applications of deep learning in material sciences: A comprehensive review. *Journal of Materials Science and Technology, 30*(2), 89-115.

[8] Asmaa, R., Ahmed, H., and Saleh, M. 2023. From phishing behavior analysis and feature selection to enhance prediction rate in phishing detection. *Journal of Information Security and Applications, 72*, 103031. https://doi.org/10.1016/j.jisa.2023.103031.

[9] Kulkarni, A. 2022. Convolution neural networks for phishing detection. *International Journal of Advanced Computer Science and Applications, 14*(4). https://doi.org/10.14569/IJACSA.2023.0140403