IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 6, June Month 2021

DOI 10.17148/IJARCCE.2021.106133

PHISHING ATTACK DETECTION USING DEEP NEURAL NETWORK

Sandhya G.V[1], Dr. Harish Kumar B T[2]

M.Tech Student, Department of Computer Science and Engineering, Bangalore Institute of Technology, Bangalore, Karnataka, India[1]

Assistant Professor, Department of Computer Science and Engineering, Bangalore Institute of Technology, Bangalore, Karnataka, India[2]

Abstract: The term phishing reminds us of the malpractice that targets the end-user, making him or her a victim unknowingly. The term phishing came into the limelight in the year 1987. It is a fraudulent practice where in the attacker traps the victim to navigate to an illegal website that resembles the legitimate website. The victim's most sensitive information related to their login data and the card details is tampered with. Hence, phishing is the best illustration for social engineering attack to trap the end-users. Hackers use the internet as a medium to deceive people.

Keywords: DNN, Phishing attack and LSTM

1. INTRODUCTION

The act of phishing requires enormous technical knowledge and smart ways to gain the details from the users trickily. The different sources for phishing attacks may be through mails, telephone calls, google documents, messages, etc. Comparison with real life as to how fishermen catch hold of the fishes in a similar fashion the important details regarding the end-users are fished through the internet from sources like emails, calls etc. The term came into being through a newsletter in the year 1996 by hackers to exploit the login credentials of American online accounts.
To lure the users the hackers generally make use of genuine images or sentences like adding a disclaimer to avoid being a victim of phishing attacks or genuine company images attached with a fraudulent link in the background and on clicking the image the user believes that they are accessing the genuine websites.
Assuming the link as genuine, the user clicks the link and enters all the information requested in the website like DOB, residential address, card details, phone number, etc. As days pass by phishing is becoming more dangerous and the effect on the end-users or organizations who fall trap is increasing exponentially.

2. LITERATURE SURVEY

G.Jaspher Willsie Kathrine et al. [1] describes about the different phishing attacks that that impact medium to large organizations to extract private and confidential data. There are many different ways a phishing attack works, and he also discusses the different ways to identify a real phishing attack.
Merlin .V.Kunju *et al.* [2] this review provides a better understanding of how phishing is detected and how to troubleshoot many of its problems. It also reports that some methods have limitations, such as accuracy and inability to detect phishing attacks.

Mohammad Mehdi *et al.* [3] developed a detection system for phishing using a training a classification system called XCS. It is an online machine learning method that develops a set of mandates called a classifier.
Yasin Sonmez *et al.* [4] describes a classification model to identify phishing attack characteristics and classify them. It involves extracting features from the websites and distribution sections. Feature extraction clearly defines the rules for extracting phishing features. The "SVM", "NB", and "ELM" were used to classify these features.
Mahdieh Zabihimayvan *et al.* [5] considers a consensus on the key features to use when spotting phishing sites. Based on three sets of phishing data, the detection performance is measured with the approximate fuzzy set (FRS) function.
Ankesh Anand *et al.* [6] proposed a novel method for generating an improved group of URLs (synthetic) inspired by generative model called "Generative Adversarial Networks". They collected suspicious URLs they discovered recently. As attackers rethink their strategy, it was ensured that data was updated on time.
V.V. Ramalingam et al. [7] explained that machine learning is an ideal candidate for identifying phishing websites as it can learn how to automatically find out the sites that are phished. Comparison between KNN and logistic regression is

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 6, June Month 2021

DOI 10.17148/IJARCCE.2021.106133

done by identifying malicious web addresses.

Yongjie Huang *et al.* [8] provides an effective deep learning method for detecting phishing URL, and extensive experimentation with several modern solutions for large sets of URLs demonstrates the superior performance of this method. As the results show, this approach outperforms modern solutions with new features.

Mohammed Alqahtani [9] proposed a new connection classification algorithm called PWCAC (Phishing Site Classification Using Connection Classification). A new way to create rules are suggested. Apply this for the classification of phishing sites, which is a major problem in web security, to evaluate the relevance of the method to predict phishing.

Athulya A *et al.* [10] emphasis phishing as one of the broadest threats that cannot be easily avoided. Few phishing attacks are launched with one click to steal the targeted victim's information. The best way to escape an attack due to phishing is to educate our users about the different various attacks caused by hackers. Choosing the correct software for security or applications as your browser extension against phishing to escape security issues due to data. Effective phishing is another way to significantly prevent phishing.

3. PROPOSED ARCHITECTURE

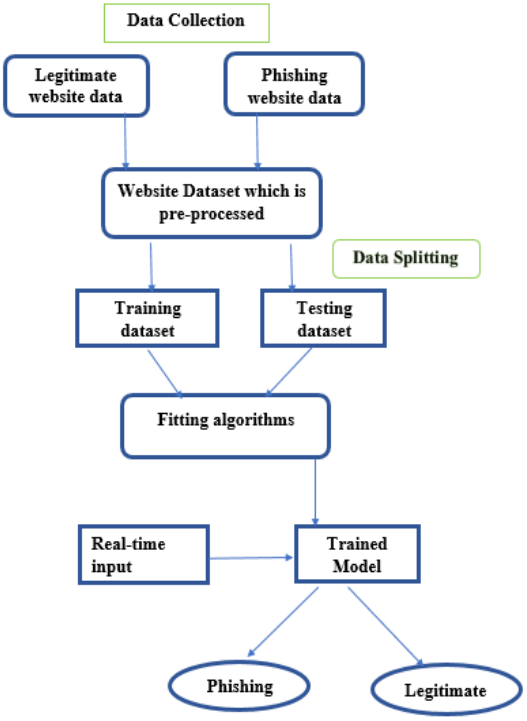

Fig 3.1 Proposed System Architecture

Figure 3.1, displays the phishing attack detection architecture related to the proposed model. The foremost step which is called dataset collection is crucial in finding the right dataset and extracting the suitable features. A total of 30 different features from the UCI database repository for verifying approximately 11,055 datasets are proposed for this project. The 30 different features are grouped under 4 main feature categories.

- The Address bar related features
- The Abnormal related features
- The HTML and JavaScript-related features
- Domain related features

The deep neural network (DNN)

The neural network which are deep in nature consist of many layers that are hidden at multi-tiers with non-linear operations. The different deep learning methods master the various hierarchies related to features by making use of the lower level feature hierarchy to shape up the higher-level hierarchies. This method proves to be a good approach that gives finer outcomes since each layer is pre-conditioned with an algorithm for unsupervised learning.

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 6, June Month 2021

DOI 10.17148/IJARCCE.2021.106133

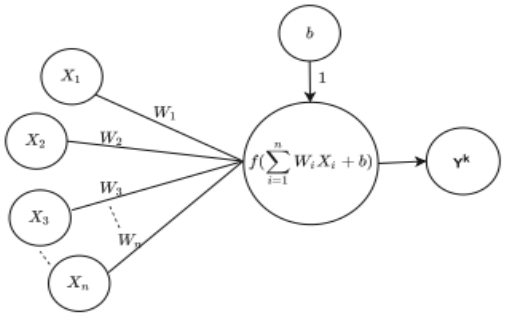

Fig 3.2 simple neuron architecture

Figure 3.2, tells that the deep neural network is a mechanism that is learnt by the machines. It contains large neural network layers in common. The structure basically consists of one layer called input, another layer called output, and many layers that are hidden. Layers contain the fundamental unit for computation called the neuron. This inspiration comes from biological neurons that perform mathematical functions to store the details. Further, the information traverses to the next neuron and therefore the information travels in the network.

A neuron's simple mathematical illustration is,

$$Y^k = \Phi\left(\sum_{k=0}^{k=n} W_{kj}x_j + b_k\right)$$

Below,

Φ Represents the "activation function".

$W_k \in R^{LB}$ Represents the "Kth neuron weight".

Y^k Represents the "Kth neuron output".

The neurons which belong to the input layer is related with the dimensions of the datasets or with the different feature attributes of the dataset. And the amount of neurons we expect would display for the output layer .Neurons showing up for the hidden layer seems to be a hyper-parameter that requires tuning to get the best results. Since every neuron is performing variety of computations, the network complexity is defined by the amount of neurons it contains.

Each deep neural network looks like a mathematical function that is very complex and that needs to adapt itself based on the nature of the data. Building a complex network would result in the problem of data "over-fitting" which works well with trained dataset but seems to fails to provide accurate results with unknown data.

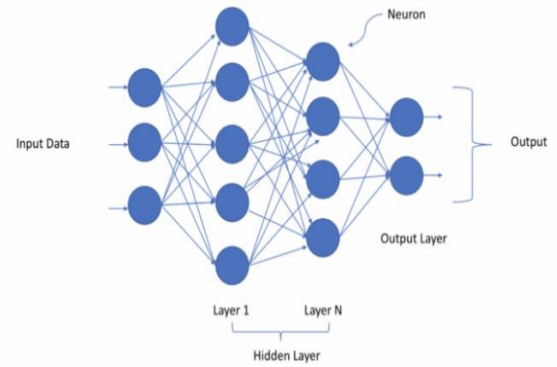

Fig 3.3 A Deep Neural Network with N hidden layers

LSTM Structure

The LSTM algorithm is an unusual type of recurrent neural network, fit for adapting "long short-term" situations. By default, their nature is to remember data for a long duration of time. The pioneers for LSTM "Hochreiter" and "Schmidhuber" in the year "1997". A mixed combination of "hidden units", sums and element-wise products among the units are used to execute the gates for controlling the cell memory.

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 6, June Month 2021

DOI 10.17148/IJARCCE.2021.106133

These memory cells are designed to hold the information without modification for long period of time. It consists of an "input gate" and "output gate", which are controlled by the learnable weights that are functions of the current observations and the "hidden units" from the previous timestep.

LSTM become more popular through continuous improvements after recurrent neural networks were discovered. Though they are used in processing a language and predicting of words they are not successful in keeping the data for long period of time. Another feature in LSTM is the use of gates for adding and deleting features.

The 3 gates are called

- The Input-gate
- The Output-gate
- The Forget-gate

The forget-gate determines the data that is unnecessary and not to be considered. The input-gate chooses the new important incoming information that needs to be saved. Finally, the output- gate chooses data needed for finding the output-activation of the LSTM units.

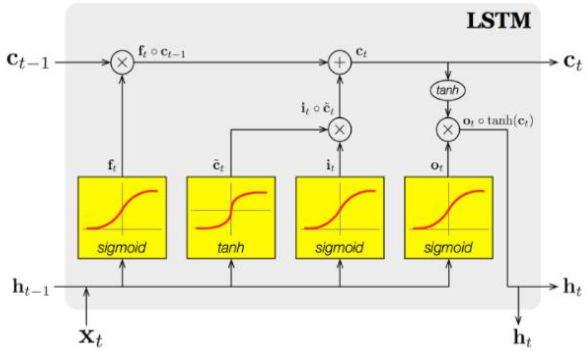

Fig 3.4 LSTM cell structure

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad \textbf{(1)}$$

As per the equation 1, the cell-state is termed as the long-term-memory. This condition related to the cell is changed by the forget-gate that is placed below the cell-state and also adjusted by the input-modulation-gate. From equation 1, the condition of the cell from the previous-cell state forgets the data by multiplying with the forget-gate and adding up fresh information through the input-gates output.

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right) \quad \textbf{(2)}$$

As per the equation 2, the forget-gate is named as the remember vector. The output which comes out of the forget-gate gives information to the cell-state regarding which kind of information it must forget by carrying out a multiplication of 0 with a position present in the matrix table. If the forget-gate outputs a 1, the data is retained in the cell-state. From equation 2, we can conclude that the sigmoid-function needs to be applied to the weighted-input and the previous-hidden-state.

$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right) \quad \textbf{(3)}$$

As per the equation 3, the input-gate is called as the saved vector. The input-gate determines what kind of information should pass through the cell-state or the long-term-memory. The crucial part is the activation-functions for each gate. The input- gate represents the sigmoid-function that has a range from [0, 1]. If we notice, the summation between the previous-cell-states is considered the cell-state equation and the sigmoid-function alone will perform the task of adding memory and will not perform remove or forget the memory. This is the reason why the input-modulation-gate contains a tanh activation-function. Tanh ranges from -1 to 1 and thus allows the cell-state to forget memory.

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 6, June Month 2021

DOI 10.17148/IJARCCE.2021.106133

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

(4)

When a floating number between the range [0, 1] is added, such numbers are never going to be zero or turned-off or forgotten. This is the reason why the input-modulation-gate consists of a function that is a tanh activation function as per the equation 4. Tanh function ranges from -1 to 1 that allows the cell-state to forget the memory.

$$o_t = \sigma\left(W_o \left[h_{t-1}, x_t\right] \; + \; b_o\right)$$

(5)

As per equation 5, the output-gate is termed as focus vector.

$$h_t = o_t * \tanh\left(C_t\right)$$

(6)

As per the equation 6, the hidden-state is the working memory.

4. IMPLEMENTATION

The data gathering and a complete knowledge of the dataset is the preliminary part of the project. Our dataset was gathered from the UC Irvine database repository and it consists of "11055" datasets that are pre-processed consisting of both legitimate and phishing websites data.

Details	Values
Total number of feature attributes	30
Complete websites data count	11055
Phished websites count	4898
Genuine websites count	6157

Table 1. Dataset description

ALGORITHM:

Step 1: The data collection or gathering of the dataset.
Data is collected from the UCI Irvine database repository.
Step 2: The data pre-processing
Pre-process the missing data from the dataset.
Step 3: The data modeling
Prepare the dataset with selected features called a feature selection and further proceed to feed it into LSTM.
Step 4: LSTM model building
List the parameters required, layers, epochs, learning rate before running the model.
Step 5: Process dataset with selected features in LSTM model.
Step 6: Split the URL dataset to a 70:30 ratio for the task of training and testing.
First, train some portion and then check it with different components for greater results. Partition as seen below:

- Identifying some portion from the dataset for testing.
- Take the remaining portion for training.

Step 7: Train the dataset by designing a prototype for training.
Step 8: The next stage is to assess the dataset for testing

- The model should be verified against the data selected for testing.
- Later calculate the performance scores and abandon the model.
- Lastly, compile an abstract of all skills the model has and finally return a rating.

Step 9: Predict the entered URL based on real-time attributes.
Step 10: Given input URL features are extracted.
Step 11: From the URL we extract the features, prepare a list form, and feed that list object that is the URL properties into the model that is trained.
Step12: Predict the input URL using LSTM by identifying the legitimate or phished sites.

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 6, June Month 2021

DOI 10.17148/IJARCCE.2021.106133

5. RESULTS

The project was conducted by using python as the programming language with the dataset taken from the UC Irvine database repository. The system performance is based on 8 GB RAM and 80 GB ROM.

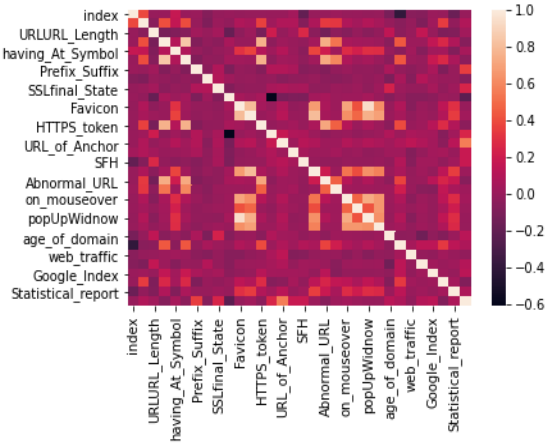

Fig 5.1 Heatmap for the different website features

The Fig 5.1, describes the Heatmap which is a graphical representation or in the form of a map showing up different color combinations for the data and corresponding values. This plot is drawn to visualize the dataset that we have collected by correlating the interested features with the available features.

Fig 5.2 Class Distribution

According to Fig 5.2, the class distribution consists of two classes the phishing class and the legitimate class.
The corresponding value for phishing is 4898 and the value for legitimate class is 6157 out of the total 11055 datasets. The number of classes are plotted against the number of observations.

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
masking (Masking)            (None, 30, 1)             0
_________________________________________________________________
lstm (LSTM)                  (None, 30, 32)            4352
_________________________________________________________________
lstm_1 (LSTM)                (None, 32)                8320
_________________________________________________________________
dense (Dense)                (None, 3)                 99
=================================================================
Total params: 12,771
Trainable params: 12,771
Non-trainable params: 0
_________________________________________________________________
```

Fig 5.3 Training Log

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 6, June Month 2021

DOI 10.17148/IJARCCE.2021.106133

The Fig 5.3 describes the sequential-model that is used to process the tasks step by step starting with the processing of a sequence of integers and integrating each of them into a 64-bit dimensional vector and lastly process the order of vectors by making use of the LSTM layers.

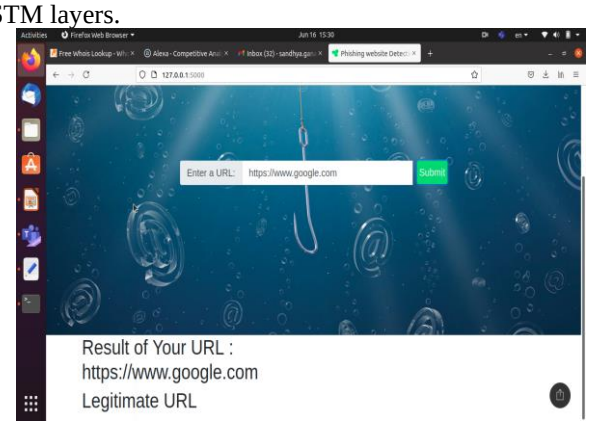

Fig 5.4 Legitimate URL

The Fig 5.4 displays the user interface where in the end user provides the input URL. Based on the LSTM prediction accuracy of 89% and the 30 features input into the LSTM model to identify the URL, the entered URL displays the result as “Legitimate URL”.

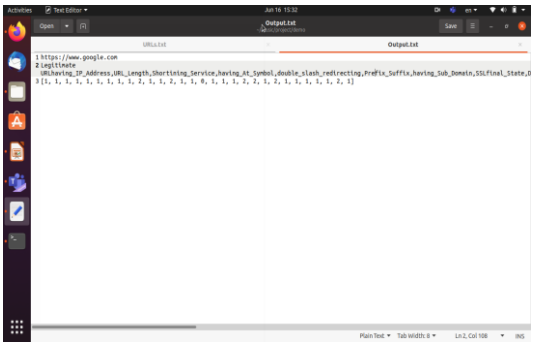

Fig 5.5 Legitimate URL output

The Fig 5.5 displays the output of the Legitimate URL which was shown as the resultant as per Fig 5.4
The output shows the 30 different feature headers and the corresponding values. The below values mean as follows.

- The value 0 indicates suspicious website.
- The value 1 indicates legitimate website.
- The value 2 indicates Phishing website.

Finally based on the weightage of each feature, the input URL is classified as “Legitimate” as per Fig 5.5

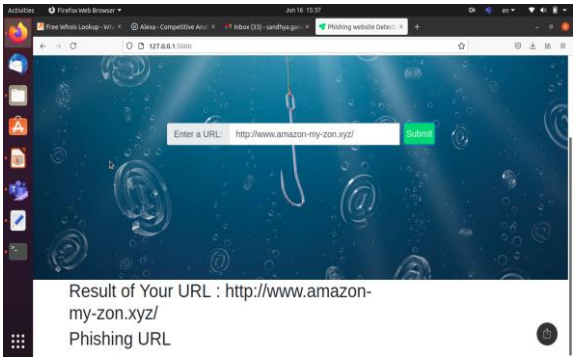

Fig 5.6 Phishing URL

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 6, June Month 2021

DOI 10.17148/IJARCCE.2021.106133

As per Fig 5.6, the input URL "http://www.amazon-my-zon.xyz/" is validated for the 30 different features, and based on the LSTM prediction accuracy of 89% of identifying the phishing websites the output is termed as phishing URL since the domain is not registered in the domain name server and there is no web traffic for the corresponding URL. The reasons for a URL to be phishing is decided based on the different heuristics.

Fig 5.7 Phishing URL output

The Fig 5.7 displays the output of the Phishing URL which was shown as the resultant as per Fig 5.6. A URL is decided as phishing based on the weightage of the features and the corresponding values showing up as 2 for phishing website.

6. PERFORMANCE EVALUATION EQUATIONS

Accuracy is measured with the aid of using suitable predictions of a classifier as opposed to the real classifications within side the dataset. The two assessment techniques "Recall" and "Precision" are calculated by using the "confusion matrix" as seen below:

Expected class	Grouped as Phishing	Grouped as Legitimate	Total
Phishing sites	Accurate Positive (15)	Inaccurate Negative (3)	18
Legitimate sites	Inaccurate Positive (2)	Accurate Negative (12)	14
Total	17(P')	15(R')	

Table 2. The "Confusion matrix"

By making use of the confusion matrix, the Accurate Positives, Inaccurate Negatives, Inaccurate Positives and Accurate Negatives are calculated by obtaining the testing results. Based on the above values the "Precision" and "Recall" values are calculated. Among 32 verified URLs which contains 18 phished URLs which further shows 15 URLs that are accurately classified as phished URLs and 3 URLs that are inaccurately classified as genuine websites. Out of 14 legitimate URLs, two URLs were mis-classified as phished URLs and 12 URLs were accurately classified as genuine sites. The system which made the prediction for websites classification has classified 15 URLs as phished and 12 URLs as genuine according to Table 2.

The project was evaluated for predicting the phishing sites by using the confusion matrix. The correctness of the classifier and also the classification model is determined by the confusion matrix.
The confusion matrices are built based on the actual-class versus the class that needs to be predicted called the predicted-class.
The expected-class**es** specify the expected results for phishing and valid categories, while the predicted-classes represent predictions for phishing and valid categories based on a heuristic network of detection system for identifying phishing

Precision : # of Accurate Positive / (# of Accurate Positive + # of Inaccurate Positive)
Recall : # of Accurate Positive / (# of Accurate Positive + # of Inaccurate Negative)
Accuracy : (# of Accurate Negative+ # of Accurate Positive) / (# of Accurate Positive + # of Accurate Negative + # of Inaccurate Positive + # of Inaccurate Negative)
F1 score : 2 * P * R / P + R
Inaccurate Positive-Rate (called fall-out)**:**

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 6, June Month 2021

DOI 10.17148/IJARCCE.2021.106133

IPR = # of Inaccurate Positive/ N = # of Inaccurate Positive / # of Inaccurate Positive + # of Accurate Negative
See below,

- **Accurate Positive (AP):** The range of rightly labeled sites which are phished.
- **Inaccurate Negative (IN):** The range of sites identified as phished however they are valid sites.
- **Inaccurate Positive (IP):** The wide ranges of sites detected as valid however had been phishing sites.
- **Accurate Negative (AN):** The variety of efficaciously labeled valid sites.
- **Precision (P):** Measures the efficaciously detected phishing assaults approximately at all times that had been detected as phishing.
- **Recall (R):** Is an equivalent to Accurate Positive.
- **F1 score:** This measure is the average (harmonic mean) between precision and recall.
- **Accuracy (A):** It is the measures of the correctly identified phished sites and genuine sites with the total sites.

Algorithms	Accuracy (%)
LSTM	89%
MLP	56%
Logistic regression	83%

Table 3. Performance of the algorithms compared

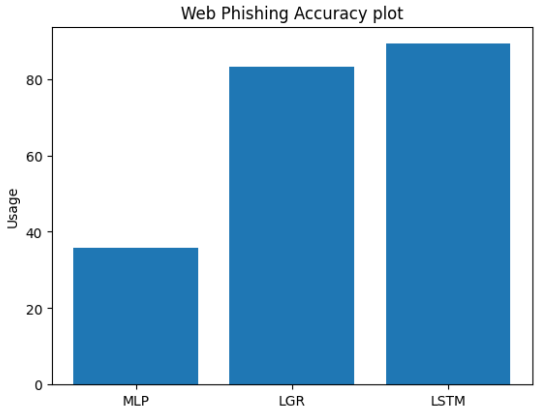

Fig 6.1 Accuracy plot

Fig 6.1 shows the comparison between the three algorithms and showing that LSTM performs best for prediction of phishing websites with 89% accuracy.

7. CONCLUSION

With reference to the project, a DNN model is used to find out the licitness of the input URL. The distinctive URL heuristics are used by the DNN model to train the LSTM algorithm. The feature attributes are tested against some deep learning-based models such as LSTM, MLP, and Logistic regression which is an ML model. Achieved a decent accuracy score of 89% with LSTM, 56% with MLP, and 83% with Logistic regression. The LSTM algorithm attained higher outcome after extracting 30 different feature attributes to detect the websites which were phished as compared with the other algorithms.

8. REFERENCES

1. G. Jaspher Willsie Kathrine, Paradise Mercy Praise, A. Amrutha Rose, and Eligious Kalaivani. C; "Variants of phishing attacks and their detection techniques", In IEEE Proceedings of the Third International Conference on Trends in Electronics and Informatics, ISBN: 978-1-5386-9439-8, 2019
2. Merlin .V.Kunju, Mrs. Esther Dainel, Heron Celestie Anthony and Sonali Bhelwa; "Evaluation of Phishing Techniques Based on Machine Learning", In Proceedings of the International Conference on Intelligent Computing and Control Systems, ISBN: 978-1-5386-8113-8, 2019
3. Mohammad Mehdi Yadollahi, Farzaneh Shoeleh, Elham Serkani, Afsaneh Madani, Hossein Gharaee; "An Adaptive Machine Learning Based Approach for Phishing Detection Using Hybrid Features ", In IEEE International Conference on Web Research, 2019
4. Yasin Sonmez, Turker Tuncer, Huseyin Gokal and Engin AVC; "Phishing Web Sites Features Classification Based on Extreme Learning Machine", In IEEE, 978-1-5386-3449-3, 2018
5. Mahdieh Zabihimayvan and Derek Doran; "Fuzzy Rough Set Feature Selection to Enhance Phishing Attack Detection", In IEEE, 978-1-5386-

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 6, June Month 2021

DOI 10.17148/IJARCCE.2021.106133

1728-1, 2019
6. Ankesh Anand, Kshitij Gorde, Joel Ruben Antony Moniz, Noseong Park, Tanmoy Chakraborty, Bei-Tseng Chu; "Phishing URL Detection with Oversampling based on Text Generative Adversarial Networks", In IEEE International Conference on Big Data, 2018
7. V.V.Ramalingam, Paras Yadav, Prakhar Srivastava; "Detection of Phishing Websites using an Efficient Feature-Based Machine Learning Framework", International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249 – 8958, Volume-9 Issue-3, 2019
8. Yongjie Huang, Qiping Yang, Jinghui Qin, Wushao Wen∗; "Phishing URL Detection via CNN and Attention-Based Hierarchical RNN", 18th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, 2019
9. Mohammed Alqahtani; "Phishing Websites Classification using Association Classification (PWCAC)", In IEEE, 978-1-5386-8125-1, 2019
10. Athulya A and Praveen K; "Towards the Detection of Phishing Attacks" IEEE Proceedings of the Fourth International Conference on Trends in Electronics and Informatics, 2019