Detecting Malicious URL using Extreme Learning Machine Algorithm

MSc Research Project
MSc Cybersecurity

Ankit Kantilal Patel
Student ID: X21101850

School of Computing
National College of Ireland

Supervisor: Prof. Jawad Salahuddin

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: Ankit Patel

Student ID: x21101850

Programme: MSc Cyber Security **Year:** 2022

Module: MSc Research Project

Supervisor: Jawad Salahuddin

Submission Due Date: 15th December 2022

Project Title: Detecting Malicious URL using Extreme Learning Machine Algorithm

Word Count: 6403 **Page Count** 20

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Ankit Kantilal Patel

Date: 14th December 2022

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Detecting Malicious URL using Extreme Learning Machine Algorithm

Ankit Kantilal Patel
x21101850

Abstract

One of the major threats for various companies is phishing attacks which utilize malicious URLs as their payloads for a successful attack. Traditional malicious URL detection systems typically used blacklisting-based approaches or signature-based approaches. Such approaches can be bypassed easily by changing the signature pattern in the URL, hence insufficient for detecting newly generated phishing or malicious URLs. Hence many systems are implementing machine learning, deep learning, neural networks, and AI-based approaches in their detection systems to make the system more accurate and faster as compared to the traditional approach. Many machine learning and deep learning algorithms are used for such research but for this proposed model, the Extreme Learning Machine algorithm is used with the sigmoid function being used in the activation function instead of the reLu function. For training the model twenty characteristics of the URL are used. The model provided 84% accuracy with 1.48 minutes of time taken to train the model which is much faster as compared to using reLu function.

1 Introduction

1.1 Malicious URL

URLs that are created to either spam a victim or fraud the victim is termed malicious URL. These URLs also enable a user to download a virus into their system by just clicking on it. Phishing attacks can also be performed using malicious URLs. Phishing attacks are termed as one of the most dangerous cyber-attacks around the world, as the victim’s data can be comprised using that attack. Hence, it is essential to identify if the URL is malicious or not before clicking on it. The attacker often uses social engineering as a tool, to make the victim click on the malicious URL. Many malicious URLs appear to be very genuine which makes the victim assume that the URL is authentic and safe to click on. In many scenarios, shortened URLs are used for phishing attacks. Hence, it is necessary to make sure if the URL is malicious or not [1].

1.2 Detection of Malicious URL

Every malicious URL has five components namely a top-level domain, a subdomain, a second level domain, and a subdirectory with a resource name for some URLs. Clicking on such a malicious URL can result in various consequences such as a virus could be downloaded which can result in credential theft or encryption of data resulting in a ransomware attack. More than eighty-five percent of the emails received are spam. Multiple techniques can be used by various organizations to detect malicious URLs such as using an email gateway that provides filtering of the email body for malicious content, using various sandboxes, and using proxy-based plugins in the web browser. These technologies have various limitations as most of them rely on filtering or a blacklisting-based approach. In recent years, there has been an increase in the use of machine learning-based technologies, deep learning, and neural networks. These technologies help to analyze a large amount of data in a faster and an accurate manner [2].

1.3 Research Question

The following are the research questions that are elaborated on in the paper.

- How fast and accurate Extreme Learning Machine algorithm can perform if the reLu function is used in the activation function of the model replacing the sigmoid function?
- Is it possible to extract as many features as possible for training the model, which can be used to train the model?

The proposed model addresses the implementation of the reLu function in the Extreme Learning Machine algorithm instead of using the sigmoid function. The solution enables the model to be trained in less time and provides higher accuracy. Detecting a serious cyber security threat at an earlier stage is essential for a company. Initial detection helps the company to eliminate the threat earlier and faster which can prevent a huge amount of business loss to the company. The model proposed further in the paper contributes towards accuracy and efficiency in detecting malicious URLs and preventing serious cyber security threats in form of phishing attacks. The remaining part of the paper is outlined as follows: Section 2 mentions the literature review conducted for the research, Section 3 mentions about the design methodology implemented in the paper, Section 4 mentions about the model implemented in the paper, Section 5 mentions about the implementation, Section 6 mentions about the evaluation and results obtained from the model, Section 7 mentions about conclusion and future work, and Section 8 mentions about all the references used in the paper.

2 Related Work

The following section explains research conducted on multiple techniques used to detect malicious URLs. Machine learning, deep learning, and neural networks are the technologies majorly used by various security researchers. Lexical features of the URL are used for training the model by all the researchers. This section is divided into three subsections namely Malicious URL detection and classification, Approaches to detect Malicious URLs using machine learning and deep learning, and finally, a summary of the research conducted.

2.1 Malicious URL Detection and Classification

As most of the techniques rely on blacklisting based approach, where the url is filtered out or blocked by analyzing the signature of the URL. Deng [3] suggested using the Earth’s Mover’s Distance. The gap between a malicious URL and a genuine URL is calculated in the proposed method. A specific threshold is calculated of a genuine webpage, and it is measured by the threshold of a malicious webpage if the threshold is not met it was detected as malicious. Visual similarity was compared as a threshold. Since the malicious webpage or the URL can be almost similar to a authentic webpage, researchers moved to other forms of technique.

Feroz and Mengel [4] proposed the first use of machine learning to detect malicious URL using scalable machine learning algorithm, Apache, and Mahout. The proposed hybrid model could detect as well as classify the URLs into different categories. The detecting process was implemented using K-algorithm, while the classification process was achieved using Microsoft Reputation Services (MRS). The proposed model achieved the accuracy of more than nighty percent, but it required more filtering as the final analysis displayed levels of threat in a single URL only.

Nagaonkar and Kulkarni [5] proposed a similar hybrid model which was also able to perform classification and detection. The proposed model worked in two different stages. First the model, detects whether a URL given is malicious or not. Second, the model classifies the malicious URL as phishing, spam, or malware. The proposed model also classified the URL into multiple categories. Kumar [6] proposed the similar approach using multi-layer categorization, which resulted in efficient and accurate classification of the URL. The model performed better in classifying the URL as spam, phishing, or malware. Tan [7] proposed a similar approach which increased the accuracy for large network traffic.

URL is made up of three parts, protocol, host name, and resource identifier. Protocol is the technology used for communication, for example whether http or https is used. Host name is name of the website for example, facebook.com. Lastly, Resource identifier is the resource which we want to use, for example, facebook.com/all.txt. Here we are trying to visit all.txt file. Shantanu [2] proposed the model which used all these three component for analysis of the URL. Machine learning models such as Logistic Regression, Stochastic Gradient Descent, Random Forest, Support Vector Machine, Naive Bayes , K-Nearest Neighbors , and Decision Tree were used to develop the model. Sklearn library was used for implementation of all the algorithms in python. Forty five thousand URLs were used to train the dataset and then for testing the model. Linear discriminant was used to perform feature extraction and principal component analysis from the URLs. For standardization and normalization of the dataset URLs, feature scaling was used. The training of the model was performed using eighty percent of the dataset, while rest of the twenty percent was used to perform the testing of the model. Out of all the algorithms used Random Forest performed with higher accuracy.

Yang [8] proposed a model using neural networks for detecting malicious URL by implementing convolutional gradient unit or GRU algorithm. Text classification features were used by model to detect if the url is malicious. The model used three components, first Keyword-Based URL Character Embedding, second Feature Extraction Module, and lastly Classification Module. The model is more focused towards using feature extraction for training the model and then scaling the features. The model was more accurate than other model compared, but the model used a lot of computing resources comparison to other models.

Lin [9] used a model to detect URLs based on large amounts of datasets. Content-based solutions were not used in the model, as it was not able to handle a large amount of data. The model filtered all the benign URLs in the first stage, reducing the large number of URLs from the dataset. Then the training and detection are performed on the rest of the dataset. One million URLs were easily classified under two minutes showing higher accuracy and efficiency. Passive-Aggressive and Confidence Weighted algorithm was used for the model. The accuracy of the model was calculated using the measure of downloading rate and missing rate of the malicious URLs.

Bahnsen [10] used lexical and statistical features of the URL in the model to detect malicious URLs. First, the features were extracted from the URLs and then fed into Random Forest and recurrent neural networks algorithm. It also used eighty twenty percent for training and testing the model simultaneously. The results evaluated that Random forest was able to classify more accurately but recurrent neural networks consumed less memory as well as the performance of predicting a malicious URL was more.

2.2 Machine Learning and Deep Learning approaches

Saleem [11] proposed surveying multiple machine learning algorithms such as SVM, Regression, KNN, Naïve Bayes, and Random Forest for detecting malicious URLs. Lexical, Content-Based, Host-Based, and Reputation-Based features were used from the URL to the analysis. A confusion matrix was used for the evaluation of all the algorithms. Deep Learning algorithms were also used such as Logistic Regression, KNN, and SVM for the analysis. The deep learning models were more accurate compared to the machine learning models. Also, the training time of the deep learning models was less compared to machine learning models.

Junaid [12] proposed a model which utilizes machine learning algorithms such as Random Forest, Extra Trees, KNN, SVM, AdaBoost, and Gradient Boosting for detecting malicious URL. Datasets were taken from Kaggle, and UNB. Lexical features of the URL were used for training the model. The results included accuracy of more than nighty eight percent and three percent of low positives.

Basnet and Doleck [13] used a heuristic model by having binary methods to detect malicious URLs. The dataset used was from the phistank website. In the model along with lexical features, keyword, reputation, and search-based features were used for analysis. Machine learning algorithms such as SVM, NB, and Random Forest were used. Random Forest performed best among all the algorithms, while the Naive Bayes algorithm didn't perform well as compared to other algorithms. Though the lightweight approach was considered for the model, the model histogram resulted in an error rate of less than 0.3 percent, a false positive rate of 0.2 percent, and a false positive rate of 0.5 percent.

Lakshmanarao [1] made a model of NLP techniques such as TF-IDF Vectorizer, Hashing Vectorizer, and Count Vectorizer which focused on processing the text of URLs. Logistic Regression, K-NN, Decision Tree, and Random Forest machine learning algorithms were used for the model. In the proposed model Random Forest achieved the best accuracy among other algorithms with ninety seven percent. The model also embedded a web application using hashing vectorizer to detect malicious URL.

Manyumwa [14] proposed a model which was able to extract multiple features from the URLs such as word-based features, Whois database, HTML web source, and geographical locations. The model was able to detect as well classify the URL. Multi class attacks were also detected by implementation of ensemble learning technique. The URLs were classified as spam, phishing, and malware. The dataset was used from multiple sources such as DMOZ, Phistank, Webspam, and URLhaus. The accuracy for the model resulted in eighty nine percent. The proposed research was able to clearly show that detection of non-malicious URL is easier as compared to malicious URL.

Yang, Zhao and Zeng [15] proposed a model which implemented deep learning algorithm, also it was able to extract multidimensional features from the URLs. CNN Long Short-Term Memory Network algorithm was used to build the model. The model proposed included three stages, which are feature extraction, embedded representation of all the features, and then detection as well as classification of the URLs. The results produced more accuracy as compared to common machine learning algorithms. The false positive was also less in the model due to usage of multidimensional features.

Mohammad [16] proposed a model using a deep learning algorithm but it had not used trial and error methods while detecting which made the results much more accurate and faster as compared to other traditional deep learning and machine learning models.

Vazhayil [17] also implemented a model using deep learning algorithms such as Convolutional Neural Network (CNN) and CNN Long Short-Term Memory Network (CNN-LSTM). The model used URLs as input instead of using features of the URL. The model was quite complex to be implemented but the accuracy of the model was quite good. But [18] implemented a similar model using recurrent neural network (RNN), long short-term memory networks, and CNN-RN algorithms. The results produced were that CNN-LSTM had the highest accuracy among all the algorithms used.

Yuan [19] proposed a model which used semantic and visual features of URL as input to the neural networks such as capsule networks, independent recurrent neural networks, and attention mechanism. The model was able to detect and then classify the malicious URLs. The model resulted in more than ninety percent accurate in detecting the malicious URLs.

Ateeq [20] proposed a model which uses lexical features of the URL for analysis. The model used multi-layer feed forward network. The author implemented seventy percent of the data for training, fifteen percent for testing, and remaining fifteen percent for testing. The accuracy of the model was less, up to seventy five percent.

Sen, Ray and Chakrabarti [21] proposed a hybrid model using CNN and LSTM algorithm. The model used many features of the URL such as list-based approach, URL-based schemes, learning-based , and content based. The model achieved ninety percent accuracy.

Patil [22] proposed a model to detect clickjacking attacks by detecting malicious url. The model used extreme learning machine algorithm with the sigmoid activation function. The model used lexical features for analysis. It used eighty twenty percentage for training and testing the datasets in the model. The results achieved were more than ninety one percent.

2.3 Literature Review Summary

The results of literature review were mostly on machine learning models as well as neural networks or deep learning models. Most of the model used various pre-processing techniques to extract features of the URL, and then the data was given to the algorithm implemented for training and detecting. Most of the models in the research had high accuracy but more false positive in most of the cases in machine learning algorithm based models. However, in the proposed model the false positives are reduced by using reLu function and it utilizes all the lexical features for analysis.

3 Research Methodology

Due to lockdowns during covid times, there has been increase in the usage of internet. This has resulted in more work from home culture by almost every company. This made use of various technologies such as VPN, which makes the office network accessible through personal internet connection. Due to this, there has been increase in the various attack surfaces. There is increase in ransomware attacks on various companies especially after covid times. The first stage of the attacks is phishing for most of the cases, where the attacker crafts a malicious URL and with social engineering makes the victim visit it [21] [18]. In the proposed model, extreme learning machine algorithm is used with reLu function replacing the sigmoid function in the model. ReLu function makes the training time less and also makes the accuracy more as compared to sigmoid function. The model implemented is as Figure 1. The following subsections, mentions about the selection of the dataset, pre-processing of the URLs, data being visualized, modelling of the data, and evaluation of the proposed model.

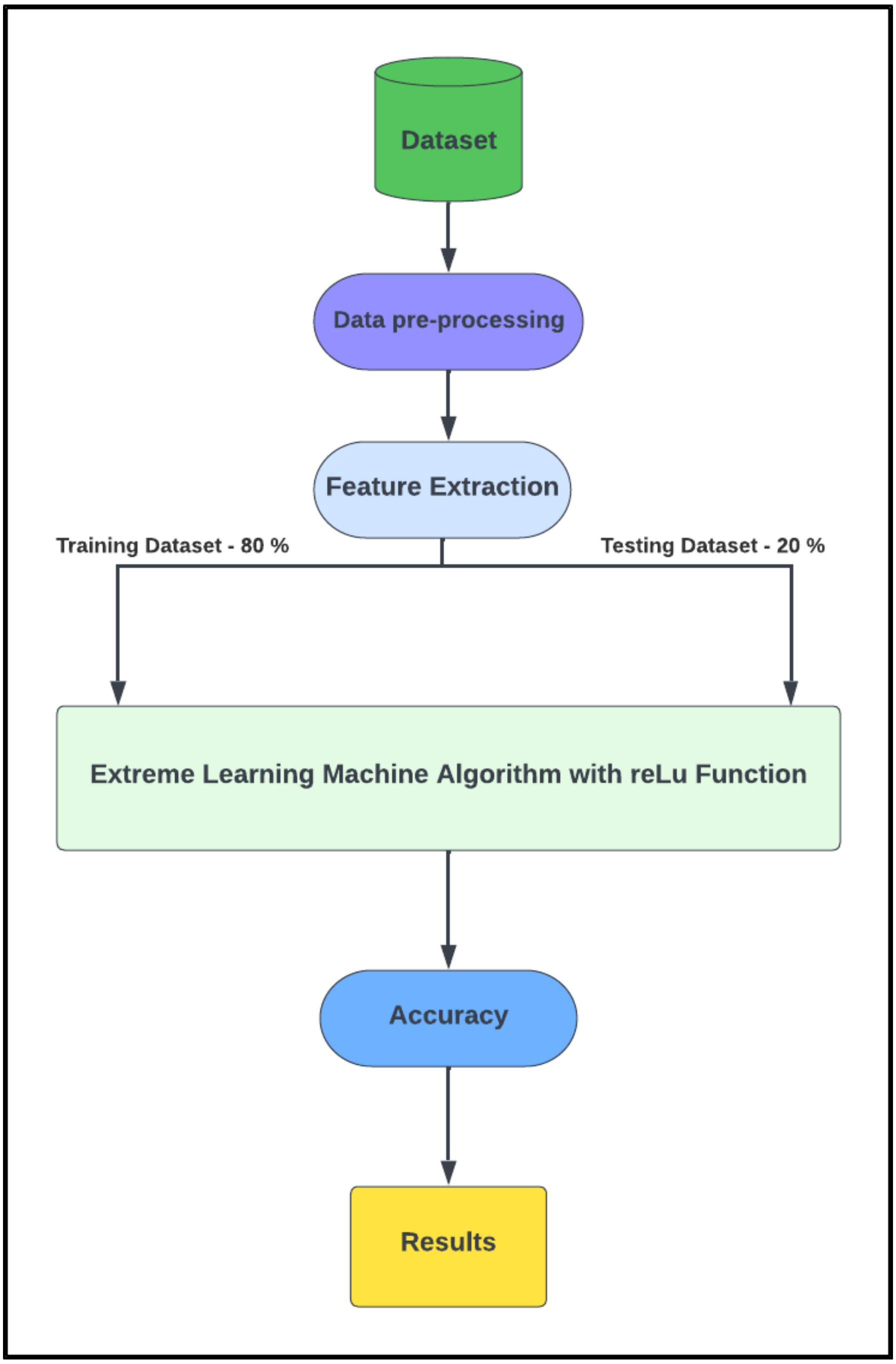


Figure 1: Proposed ELM Model for Malicious URL Detection.

3.1 Selection of Dataset

The dataset is selected from Kaggle [23] having 6,51,191 URLs. The URLs are classified into 4,28,103 as safe URLs, 96,457 as defacement URLs, 94,111 as phishing URLs, and 32,520 as malware URLs. The dataset is made from multiple sources such as phistorm, PhishTank, etc.

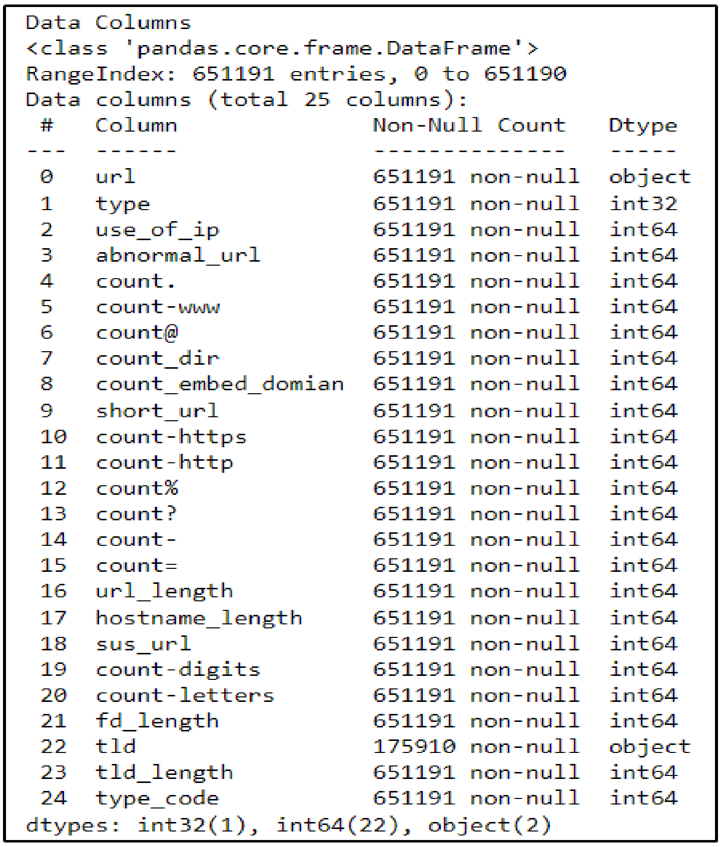
The classification in terms of percentage is as follows, about sixty-six percent of the URLs are safe URLs, fifteen percent of the URLs are of type defacement, and the rest are of type others. The dataset contains two columns, one containing the URL, and the second containing the label.

3.2 Dataset Pre-processing

This is the second stage of the model after, dataset selection. In this process, we adjust the dataset in an appropriate manner where the dataset is cleaned with removing the null values, balancing the dataset if some values are missing, and generalizing the data such as graph plots for various URLs. In the model, we have used bar plotting to make sure we analyze the dataset appropriately. Seaborn python library is used for plotting and analyzing the dataset.

3.3 Feature Extraction

After the pre-processing of the dataset, we try to extract features from the URLs, which is applied to model for training and testing. As the dataset is not provided with features, it is essential to extract the necessary features. In this stage, lexical features are extracted. Features such as does the URL contain any IP address, any abnormal URLs, Dots in the URL, www in URL, @ in URL, no of directories present in the URL, // in URL, any shortening service used in URL, https in URL, http used in URL, ? in URL, % in URL, presence of “-“ and “=” in URL, any suspicious words such as PayPal, login, or bonus in the URL, digits in URL, letter count in URL, length of the URL, and usage of top level domain in URL. Total of twenty five features were extracted from the URL. The features have integer type of value stored in the column. List of total features extracted are shown in the Figure 2.



```
Data Columns
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 651191 entries, 0 to 651190
Data columns (total 25 columns):
 #   Column               Non-Null Count   Dtype
---  ------               --------------   -----
 0   url                  651191 non-null  object
 1   type                 651191 non-null  int32
 2   use_of_ip            651191 non-null  int64
 3   abnormal_url         651191 non-null  int64
 4   count.               651191 non-null  int64
 5   count-www            651191 non-null  int64
 6   count@               651191 non-null  int64
 7   count_dir            651191 non-null  int64
 8   count_embed_domian   651191 non-null  int64
 9   short_url            651191 non-null  int64
 10  count-https          651191 non-null  int64
 11  count-http           651191 non-null  int64
 12  count%               651191 non-null  int64
 13  count?               651191 non-null  int64
 14  count-               651191 non-null  int64
 15  count=               651191 non-null  int64
 16  url_length           651191 non-null  int64
 17  hostname_length      651191 non-null  int64
 18  sus_url              651191 non-null  int64
 19  count-digits         651191 non-null  int64
 20  count-letters        651191 non-null  int64
 21  fd_length            651191 non-null  int64
 22  tld                  175910 non-null  object
 23  tld_length           651191 non-null  int64
 24  type_code            651191 non-null  int64
dtypes: int32(1), int64(22), object(2)
```

Figure 2: List of Features extracted from the URLs.

3.4 Visualization of the Datasets and Features

It is essential to visualize the datasets as well as the processing of the features to understand the dataset in more depth. The model has used multiple visualization formats such as pie charts, bar graphs, and many others. Even all the features of the dataset are also visualized after extraction from the URL to proceed accurately in designing the model.

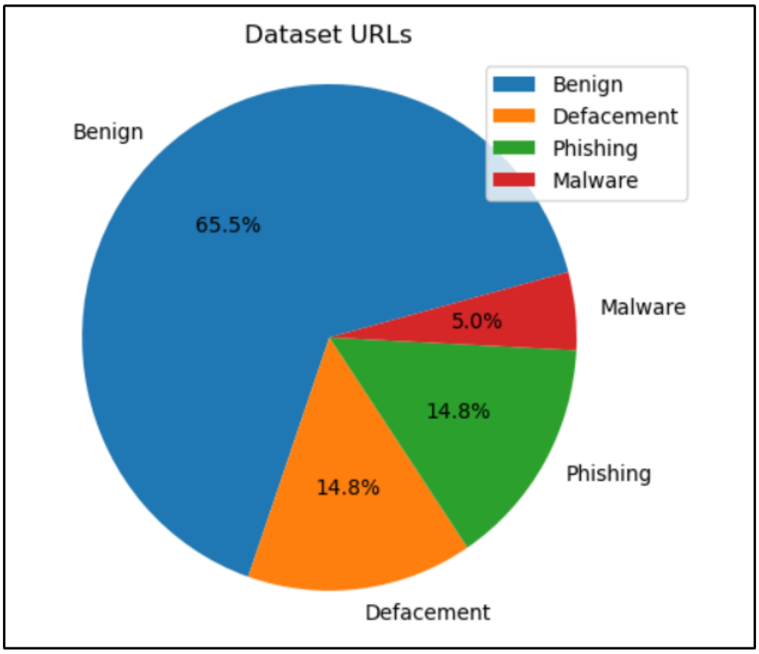


Figure 3: Types of URLs classified in the dataset.

Figure 3 shows the visualization of the URLs in the dataset, where more than sixty percent of the URLs are safe. The rest of the URLs are Malware, Phishing, and Defacement which is more than thirty-five percent. Hence the dataset has sixty percent of URLs which are safe, while forty percent of the URL are malicious.

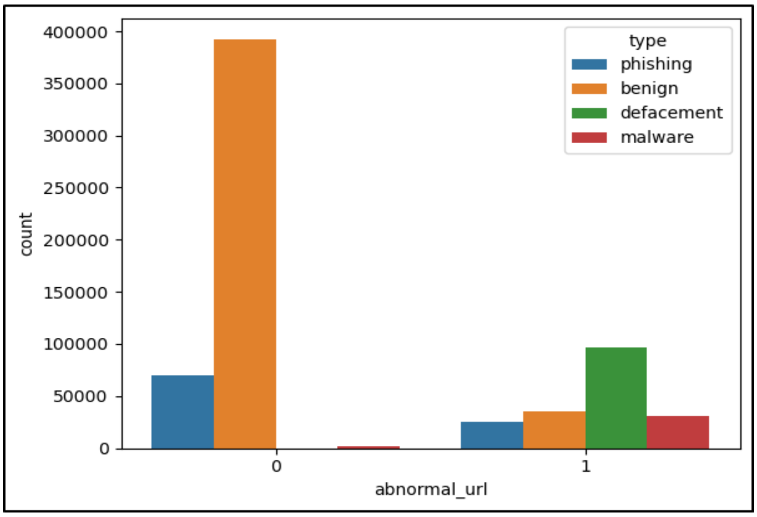


Figure 4: Abnormal URL extracted from the dataset.

Figure 4 shows abnormal URLs extracted from the dataset, the abnormal URLs are divided into one and zero. One represents, the URL is an abnormal URL. Zero represents, the URL is not an abnormal URL. Figure 4 shows that most of the abnormal URLs are from phishing, defacement, and malware. While some of the benign URL are also said to be abnormal. Abnormal URLs are checked using WHOIS library in python.

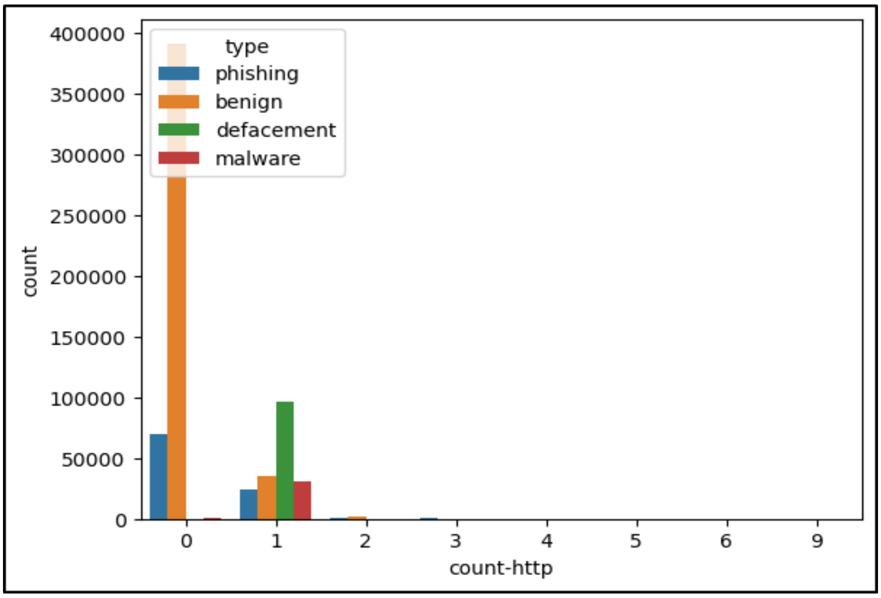


Figure 5: HTTP Features extracted from the Dataset.

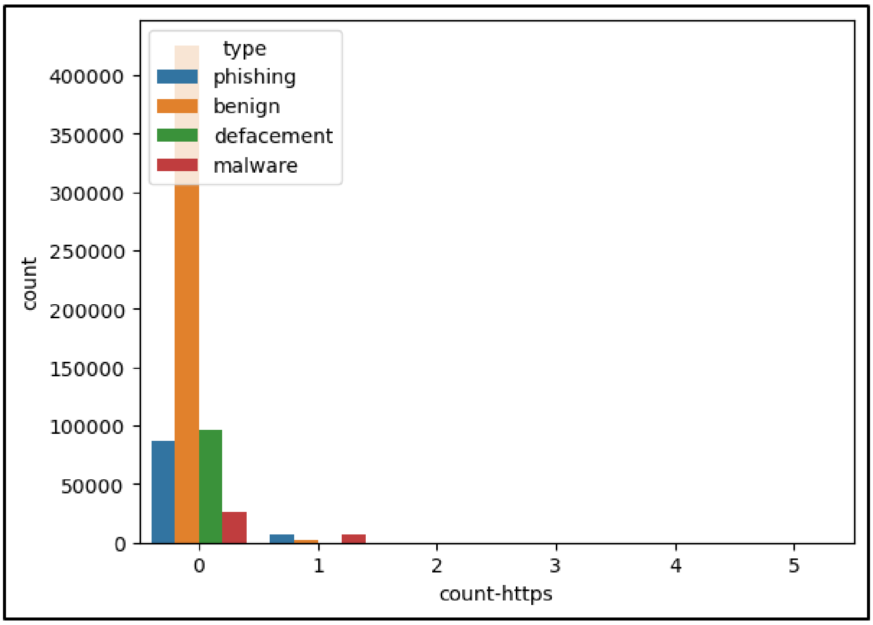


Figure 6: HTTPS features extracted from Dataset.

SSL certificates for a website is very essential, as it defines that the communication is secure and encrypted. Hence, URLs which have https can be considered safe and secure. HTTPS features must be considered while checking for safe URLs. In Figure 5 and Figure 6, we can see that benign URLs have used https whereas many malicious URLs have used http.

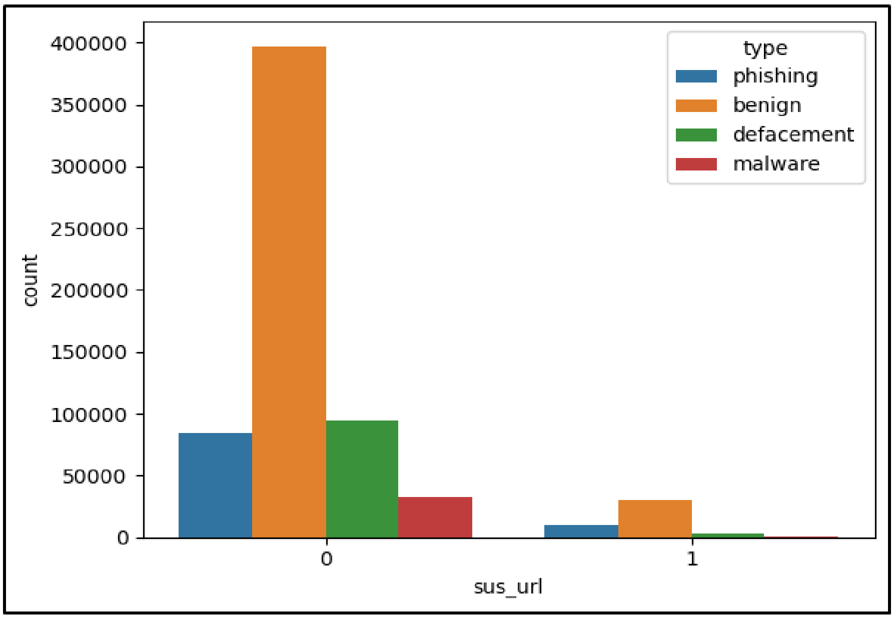


Figure 7: Suspicious keywords extracted from Dataset.

In Figure 7, suspicious keywords such as PayPal, bonus, update, and many others are searched in the URL. Figure 7 shows that some of the malicious URLs have such suspicious words, while most of the benign URLs don't have them. Zero represents no presence of suspicious keywords, and one represents the presence of suspicious keywords.

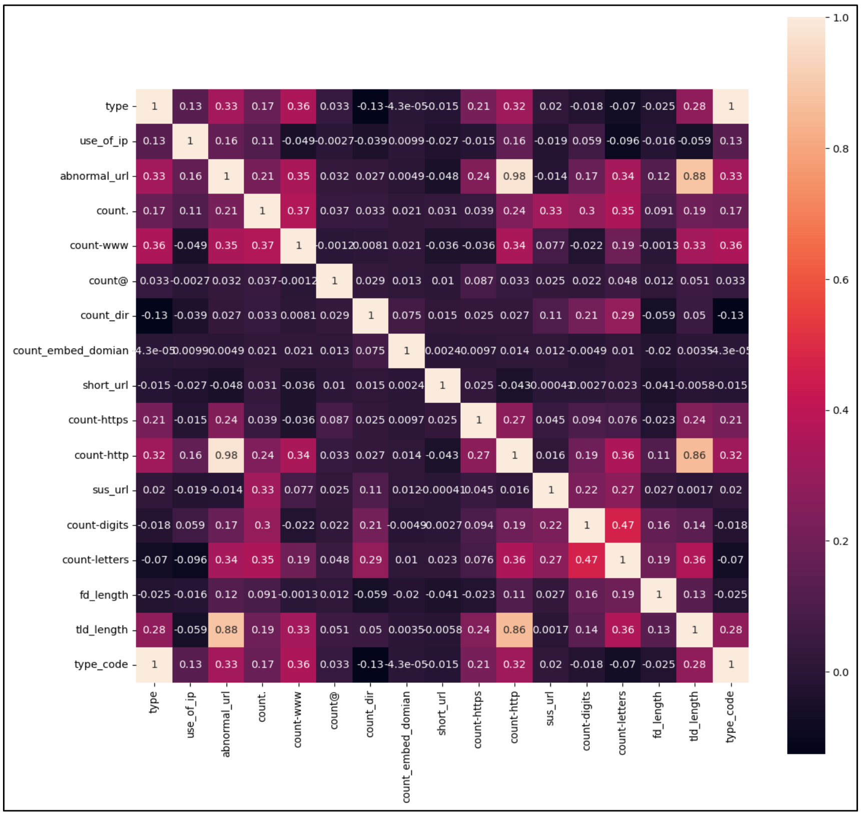


Figure 8: Correlation Matrix of features extracted from the Dataset.

The above Figure 8 shows the correlation between the features extracted from the URLs. Correlated features which are high are identified using correlation matrix. Such correlated features are removed from the dataset as they create a lot of noise while training the model. But in the matrix we can see there are not much high correlated features, hence we do not remove any features from the dataset. The correlation matrix is essential to make the dataset noise free and good for further analysis.

3.5 Training and Testing of the Dataset

After extracting the required features from the dataset, it is necessary to split the dataset into eighty-twenty percent ratio for training and testing simultaneously. To split the dataset, we are using train_test_split library. First, 177884 set of URLs are used for training, which accumulates eighty percent of the dataset. Second, 44600 set of URLs are used for testing the model, which accumulates to twenty percent of the dataset. The following figure shows the distribution of the dataset for training and testing the model.

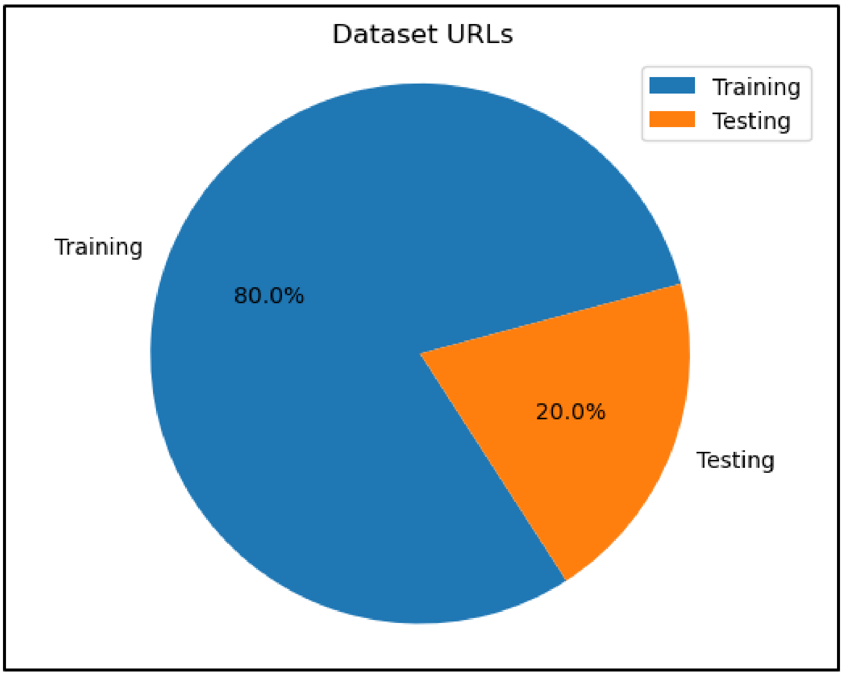


Figure 9: Splitting of Dataset into 80%-20%.

3.6 Evaluation and Result

For the model, we have used Extreme Learning Machine Algorithm using the reLu function instead of using the sigmoid function. For the evaluation of the model, a confusion matrix will be used. Other techniques such as F-1 scores, recall, and precision will be calculated for the model. We will calculate the time taken by the model to train using both the sigmoid and reLu functions, as well as the accuracy for the model with the sigmoid as well as reLu functions.

4 Design Specification

To accurately detect malicious URLs, we have implemented the Extreme Learning Machine algorithm which was implemented by [21] to detect clickjacking attacks. The ELM model is a neural network that is feedback in nature. The model uses the sigmoid function, but we have implemented the reLu function in the model for better accuracy and less training time.

Gradient-based algorithms are very slow in comparison to the ELM Mode in terms of learning which provides an extra advantage for the model as it does not face issues faced by gradient models such as local minima, stopping criteria, and also learning rate [24], which helps the ELM model to attain higher accuracy and performance.

4.1 Extreme Learning Machine Model

Extreme Learning Machine algorithm is a single layer model which have a hidden layer of neural networks which are feed forward in design. The ELM model has a single hidden layer between the input and the output layers. Weights can be chosen for the model in middle of the input and hidden layers [25]. High performance of the model and learning rate of the model depends on accurate values assigned to the weights, value of the threshold set, and the activation function. The neurons in the ELM model are activated using activation function such as reLu function [26]. Figure 10 shows the output, input, hidden nodes and activation function [27].

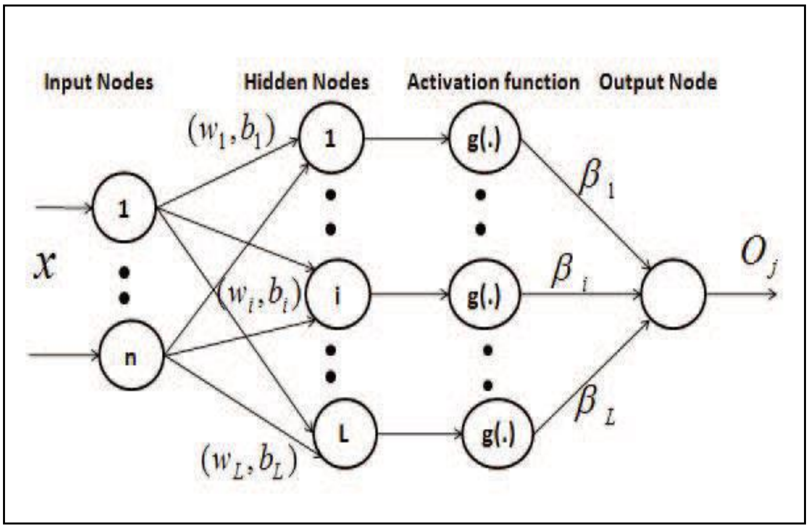
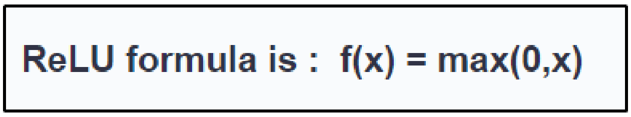


Figure 10: Extreme Learning Machine Model.

In the model implemented, we have used ELM classifier as the base class for the ELM model. ELM classifier class performs binary classification and provides results in form of either one or zero [25]. This is essential in the model as it produces the output as to either the URL is malicious or not.

4.2 reLu Function in the ELM Model

Extreme Learning Machine algorithm uses sigmoid function as an activation function, but we have implemented reLu function in the activation instead of using sigmoid function. ReLu function is a type of line piecewise function that produces output if the input is positive else it ill produces the output as zero. Nowadays many neural networks use reLu function as their activation function in the model as it is easier to train and achieves higher efficiency as compared to other activation function such as sigmoid function [28] The following Figure 11 shows the reLu function.



ReLU formula is : f(x) = max(0,x)

Figure 11: reLu Function.

5 Implementation

For the model implemented, main goal of the model is to detect malicious URLs with good accuracy and the training time must be reduced by using reLu function. With new techniques of phishing there are always new URLs generated which are very difficult to detect. Hence it is essential to make sure the data is accurate for the model to detect malicious URL. For this reason, many libraries have been implemented to make the dataset appropriate for the model such as pandas, NumPy, Sklearn, urlib, seaborn, requests, re, and many others. To visualize the features extracted from the dataset, matplotlib library is used to make various graph plots, correlation matrix, and pie charts. Pandas library is used to vie changes in the dataset as well to load the dataset into python jupyter. To analyze the statistical graphics seaborn is used along with pandas. For detecting abnormal URLs, a request is made to Whois database hence requests library is used. We also require searching for various features such as dots, exclamation marks, and so on, hence for such operation re library is used which performs regex search. After all the libraries are imported and the model is implemented, we detect the malicious URL using the model and then we compare the training time for both model with reLu function and model with sigmoid function, also accuracy for both the activation function is compared with the same model. Following are the specifications of the model being implemented.

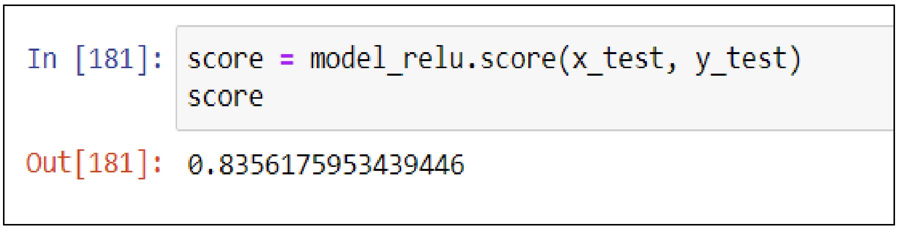
- RAM: 16GB
- Hard Disk Space: 50GB
- Operating System: Windows 11
- Programming Language: Python3
- Libraries: Pandas, NumPy, Matplotlib, Sklearn, Seaborn, Urllib, Os, itertools, and urlparse.

6 Evaluation

To efficiently detect malicious URLs, with the Extreme Learning Machine algorithm, various metrics are considered to calculate the efficiency and accuracy of the model which are accuracy, training time, F-1 scores, recall, and precision. The model is also evaluated with a confusion matrix for the reLu function.

6.1 Experiment 1 / Evaluation based on Accuracy

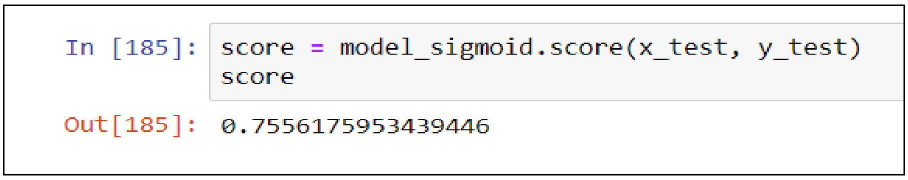
In the experiment, we trained both the activation functions with the ELM model on the same dataset, then we tested the model, and lastly calculated the accuracy for both the activation function. Here the results were that the reLu function had higher accuracy as compared to the sigmoid activation function. ReLu function performed with 83 percent of accuracy whereas the sigmoid activation function performed with 75 percent of accuracy. The following Figure 12 and 13 shows the accuracy of both activation function.



```
In [181]: score = model_relu.score(x_test, y_test)
          score

Out[181]: 0.8356175953439446
```

Figure 12: reLu Activation Function Accuracy.



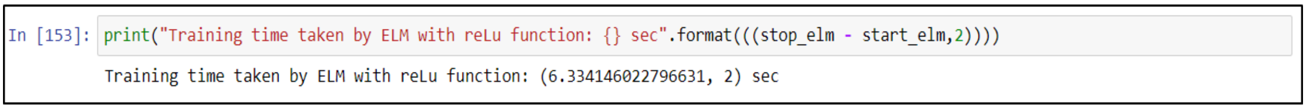
```
In [185]: score = model_sigmoid.score(x_test, y_test)
          score

Out[185]: 0.7556175953439446
```

Figure 13: Sigmoid Activation Function Accuracy.

6.2 Experiment 2 / Evaluation based on training time

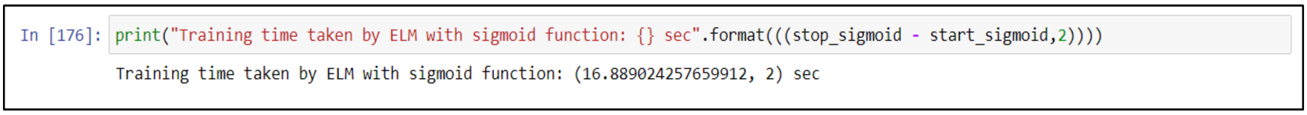
The training time of both the activation function were calculated using the time library of python by calculating the start of the training and then the end of the training of the datasets. The results accumulated showed that the reLu function trained in 6.3 seconds whereas the sigmoid trained in 16 seconds. The following Figure 14 and 15 shows the training times of both activation functions.



```
In [153]: print("Training time taken by ELM with reLu function: {} sec".format(((stop_elm - start_elm,2))))

          Training time taken by ELM with reLu function: (6.334146022796631, 2) sec
```

Figure 14: reLu Activation Function Training Time.



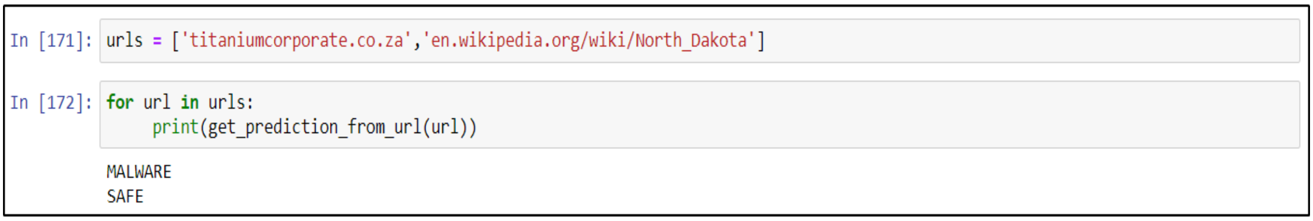
```
In [176]: print("Training time taken by ELM with sigmoid function: {} sec".format(((stop_sigmoid - start_sigmoid,2))))

          Training time taken by ELM with sigmoid function: (16.889024257659912, 2) sec
```

Figure 15: Sigmoid Activation Function Training Time.

6.3 Experiment 3 / Detecting Malicious URLs using reLu function

For efficient detection of the malicious, we created a function that will be called to extract the features from the URLs and then given to predict variables to predict whether the URL is malicious or not. We were able to detect the URLs provided to the model. The following Figure 16 shows the detection of the URLs.



```
In [171]: urls = ['titaniumcorporate.co.za','en.wikipedia.org/wiki/North_Dakota']

In [172]: for url in urls:
              print(get_prediction_from_url(url))

          MALWARE
          SAFE
```

Figure 16: Detecting Malicious URL using reLu Activation Function.

6.4 Experiment 4 / Confusion Matrix of the ELM Model with reLu Function

The following Figure 17 shows the confusion matrix of the ELM Model.

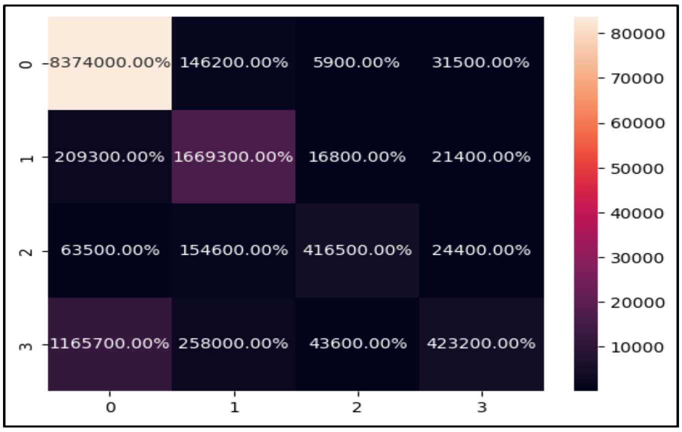


Figure 17: Confusion Matrix of the ELM Model with reLu function.

The performance characteristic of the model is calculated in the confusion matrix. The confusion matrix properties such as True Positive, False Positive, True Negative, and False Negative. The following Table 1 shows all the results calculated from the Confusion Matrix.

Table 1: Confusion Matrix of ELM Model

CONFUSION MATRIX VALUES			
True Positive	**True Negative**	**False Positive**	**False Negative**
883	1140	112	76

The performance matrix of the model calculated through Confusion Matrix is as follows.

Table 2: Performance Matrix

Accuracy	Precision	Recall	F1-Score	Training Time
83%	85%	90%	90%	6.3 Seconds

6.5 Discussion

Extreme Learning Machine model implemented with reLu function is compared with the sigmoid activation function with respect to various factors. Since, the sigmoid function is better than other activation functions, but it was observed that reLu function performed much better than sigmoid function in comparison to training time, and accuracy. The time taken by reLu function to train is 6.3 seconds whereas sigmoid function was trained in 16 seconds. In terms of accuracy reLu function is 83 percent while sigmoid function is 75 percent. Hence in terms of both the parameters accuracy and time, reLu function performs better and faster than sigmoid function and can be used for performing various other research in the neural network models.

7 Conclusion and Future Work

Malicious URLs are constantly emerging in various phishing attacks which are customized to bypass various technologies used to detect them such as blacklisting or filtering based approaches. This resulted into use of machine learning or deep learning based models to detect the malicious URL. The ML and deep learning models proved to be fast, and accurate as compared to the traditional technologies. The research proposed a model which uses Extreme Learning Machine algorithm with reLu activation function to detect malicious URLs. After successful implementation of the model, URLs were detected using the algorithm implemented in the model with eighty three percent of accuracy, proving the solution to be effective and fast.

For the future work, we can implement various extensions on the browser which can detect the URLs real time before visiting it and does not allow the user to visit the site if its malicious. Other deep learning algorithms can also be included in the model with higher accuracy to make the model more efficient and have less false positive rate.

References

[1] A. Lakshmanarao, M. R. Babu, and M. M. Bala Krishna, “Malicious URL Detection using NLP, Machine Learning and FLASK,” in 2021 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES), Sep. 2021, pp. 1–4. doi: 10.1109/ICSES52305.2021.9633889.

[2] Shantanu, B. Janet, and R. Joshua Arul Kumar, “Malicious URL Detection: A Comparative Study,” in 2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS), Mar. 2021, pp. 1147–1151. doi: 10.1109/ICAIS50930.2021.9396014.

[3] A. Y. Fu, L. Wenyin, and X. Deng, “Detecting Phishing Web Pages with Visual Similarity Assessment Based on Earth Mover’s Distance (EMD),” IEEE Transactions on Dependable and Secure Computing, vol. 3, no. 4, pp. 301–311, Oct. 2006, doi: 10.1109/TDSC.2006.50.

[4] M. N. Feroz and S. Mengel, “Phishing URL Detection Using URL Ranking,” in 2015 IEEE International Congress on Big Data, Jun. 2015, pp. 635–638. doi: 10.1109/BigDataCongress.2015.97.

[5] A. R. Nagaonkar and U. L. Kulkarni, “Finding the malicious URLs using search engines,” in 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom), Mar. 2016, pp. 3692–3694.

[6] R. Kumar, X. Zhang, H. A. Tariq, and R. U. Khan, “Malicious URL detection using multi-layer filtering model,” in 2017 14th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP), Dec. 2017, pp. 97–100. doi: 10.1109/ICCWAMTIP.2017.8301457.

[7] G. Tan, P. Zhang, Q. Liu, X. Liu, C. Zhu, and F. Dou, “Adaptive Malicious URL Detection: Learning in the Presence of Concept Drifts,” in 2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE), Aug. 2018, pp. 737–743. doi: 10.1109/TrustCom/BigDataSE.2018.00107.

[8] W. Yang, W. Zuo, and B. Cui, “Detecting Malicious URLs via a Keyword-Based Convolutional Gated-Recurrent-Unit Neural Network,” IEEE Access, vol. 7, pp. 29891–29900, 2019, doi: 10.1109/ACCESS.2019.2895751.

[9] M.-S. Lin, C.-Y. Chiu, Y.-J. Lee, and H.-K. Pao, “Malicious URL filtering — A big data application,” in 2013 IEEE International Conference on Big Data, Oct. 2013, pp. 589–596. doi: 10.1109/BigData.2013.6691627.

[10] A. C. Bahnsen, E. C. Bohorquez, S. Villegas, J. Vargas, and F. A. González, “Classifying phishing URLs using recurrent neural networks,” in 2017 APWG Symposium on Electronic Crime Research (eCrime), Apr. 2017, pp. 1–8. doi: 10.1109/ECRIME.2017.7945048.

[11] S. R. A, M. R, R. N, S. L, and A. N, “Survey on Malicious URL Detection Techniques,” in 2022 6th International Conference on Trends in Electronics and Informatics (ICOEI), Apr. 2022, pp. 778–781. doi: 10.1109/ICOEI53556.2022.9777221.

[12] H. M. Junaid Khan, Q. Niyaz, V. K. Devabhaktuni, S. Guo, and U. Shaikh, “Identifying Generic Features for Malicious URL Detection System,” in 2019 IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), Oct. 2019, pp. 0347–0352. doi: 10.1109/UEMCON47517.2019.8992930.

[13] R. B. Basnet and T. Doleck, “Towards Developing a Tool to Detect Phishing URLs: A Machine Learning Approach,” in 2015 IEEE International Conference on Computational Intelligence & Communication Technology, Feb. 2015, pp. 220–223. doi: 10.1109/CICT.2015.63.

[14] T. Manyumwa, P. F. Chapita, H. Wu, and S. Ji, “Towards Fighting Cybercrime: Malicious URL Attack Type Detection using Multiclass Classification,” in 2020 IEEE International Conference on Big Data (Big Data), Dec. 2020, pp. 1813–1822. doi: 10.1109/BigData50022.2020.9378029.

[15] P. Yang, G. Zhao, and P. Zeng, “Phishing Website Detection Based on Multidimensional Features Driven by Deep Learning,” IEEE Access, vol. 7, pp. 15196–15209, 2019, doi: 10.1109/ACCESS.2019.2892066.

[16] F. Thabtah, R. M. Mohammad, and L. McCluskey, “A dynamic self-structuring neural network model to combat phishing,” in 2016 International Joint Conference on Neural Networks (IJCNN), Jul. 2016, pp. 4221–4226. doi: 10.1109/IJCNN.2016.7727750.

[17] A. Vazhayil, R. Vinayakumar, and K. P. Soman, “Comparative Study of the Detection of Malicious URLs Using Shallow and Deep Networks,” in 2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Jul. 2018, pp. 1–6. doi: 10.1109/ICCCNT.2018.8494159.

[18] A. Das, A. Das, A. Datta, S. Si, and S. Barman, “Deep Approaches on Malicious URL Classification,” in 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Jul. 2020, pp. 1–6. doi: 10.1109/ICCCNT49239.2020.9225338.

[19] J. Yuan, G. Chen, S. Tian, and X. Pei, “Malicious URL Detection Based on a Parallel Neural Joint Model,” IEEE Access, vol. 9, pp. 9464–9472, 2021, doi: 10.1109/ACCESS.2021.3049625.

[20] J. H. Ateeq and M. Moreb, “Detecting Malicious URL using Neural Network,” in 2021 International Congress of Advanced Technology and Engineering (ICOTEN), Jul. 2021, pp. 1–8. doi: 10.1109/ICOTEN52080.2021.9493481.

[21] M. Sen, K. S. Ray, and A. Chakrabarti, “Malicious URL Classification Using Deep Neural Network,” in 2021 IEEE 18th India Council International Conference (INDICON), Dec. 2021, pp. 1–6. doi: 10.1109/INDICON52576.2021.9691762.

[22] Y. Patil, “Detection of Clickjacking Attacks using the Extreme Learning Machine algorithm,” p. 19.

[23] “Malicious URLs dataset.” https://www.kaggle.com/datasets/sid321axn/malicious-urls-dataset (accessed Dec. 06, 2022).

[24] H.-J. Rong, G.-B. Huang, and Y.-S. Ong, “Extreme learning machine for multi-categories classification applications,” in 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), Jun. 2008, pp. 1709–1713. doi: 10.1109/IJCNN.2008.4634028.

[25] D. T. Várkonyi and K. Buza, “Extreme Learning Machines with Regularization for the Classification of Gene Expression Data,” p. 5.

[26] Y. Sönmez, T. Tuncer, H. Gökal, and E. Avcı, “Phishing web sites features classification based on extreme learning machine,” in 2018 6th International Symposium on Digital Forensic and Security (ISDFS), Mar. 2018, pp. 1–5. doi: 10.1109/ISDFS.2018.8355342.

[27] S. Baraha and P. K. Biswal, “Implementation of activation functions for ELM based classifiers,” in 2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), Mar. 2017, pp. 1038–1042. doi: 10.1109/WiSPNET.2017.8299920.

[28] K. Tachibana and K. Otsuka, “Wind Prediction Performance of Complex Neural Network with ReLU Activation Function,” in 2018 57th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE), Sep. 2018, pp. 1029–1034. doi: 10.23919/SICE.2018.8492660.