Journal of Theoretical and Applied Information Technology
30th June 2022. Vol.100. No 12

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

PERFORMANCE EVALUATION BY FEATURE REDUCTION USING DEEP LEARNING FOR IDENTIFYING MALICIOUS WEBSITES

SHAIK IRFAN BABU[1], DR.M.V.P. CHANDRA SEKHARA RAO[2]

[1]Research Scholar, Department of CSE, ANU College of Engineering, Acharya Nagarjuna University (ANU), Guntur, Andhra Pradesh, India

[2] Professor, Department of CSE, R.V.R. and J.C College of Engineering (A), Guntur, Andhra Pradesh, India

E-mail: [1]irfanbabushaik@gmail.com, [2]manukondach@gmail.com

ABSTRACT

Now a days, Internet activities are growing in exponential rate so are the criminal activities, with the growth of internet usage. Internet is also a source of malicious web pages. Automatic Malicious URL identification resulted a relative novel and sensitive security challenging area. The area would aim in aiding the users to overcome security threats due to the presence of malicious webpage's resulting in a better network security. The present study makes attempt in assessing and identifying malicious websites, a malicious identification model is proposed using deep learning ideas. The present work uses the URL and HTML based features to identify malicious websites. PCA is applied to reduce features, dominating features are identified. It is found that dominating features play vital role in segregating the URLs into malicious and non-malicious. Dataset from PhishTank and Alexa is used in this study. Seven Layer Neural Network has shown significant improvements resulting in accuracy of 94%. The proposed work gave true-positive rate 95.51 and False-malicious rate 9.51.

Keywords: *PCA, Neural Network, TMR, FMR.*

1. INTRODUCTION

The usage of Internet has grown exponentially and has become indispensable for humans. Such tedious usage has become vulnerable for hackers, intruders, attackers, etc., to perform non-social activities and for financial gains. Drive-by download, phishing, and social engineering & spamming are regarded as trivial attacks. If a user accesses the malicious webpages through their personal devices with no perception, malicious scripts sometimes launch attacks to put in scalawag programs, steal personal identities and credentials, or perhaps manage (take control) the victim's machine. Prevention is better than cure is more appropriate even for this menace. Identifying and isolate the malicious websites is need of the hour. Each time when the users decide to access unknown websites or click on an unfamiliar URL, a sanity check must be performed to evaluate the associated risk of visiting that website and the challenges that might be encountered.

Initially all the features in the dataset is used, each feature is given equal importance and features are not prioritized. The redundancy in the dataset is not identified and removed. As a result, training and testing of the models used additional computation power (in terms of epochs) and resources (in terms of memory). In this paper, explore a machine learning-based classification algorithm, capable of predicting whether a website is malicious or benign by analyzing HTML tags representing a Web page and URL components. To identify dominated features in two ways. By use of Deep Learning, focuses on automatically identifying the dominated features from the dataset. By use of PCA, focused on using linear algebra techniques to identify the dominated features form the data set. As resulted in reduction of the data set dimensionality for training without compromising on the performance results. Further reduced the number of epochs required

during training. These features are pre-processed and served to the Neural Network, SVM and KNN Machine Learning Classifier, which indicating the likelihood of the said URL being malicious or benign URL. Section 2 describes the literature survey, in section -3 proposed work is explored and in section-4 has discussion on the results obtained, the paper ends with conclusion in section-5.

2. RELATED WORK

Juan Carlos Prieto, Alberto Fernández-Isabel [1] et al. Discussed the DOCRIW (Domains Classifier based on Risky Websites) framework to classify based upon its domain. It can be split into two methods. The first method is based upon previous knowledge containing information of malicious websites. The second method focused on domain names of various websites such as malicious and non-malicious. In this paper focused URL based features like Host based and Domain based features and used different types of supervised classifiers used for experimental purpose.

Shreyas Rajesh Labhsetwar [2] et al., proposed models whether a website is malicious or benign based on application layer and network layer features. These features mainly focused on HTTP or HTTPS responses of website, upon trained supervised machine learning algorithms to check the given sites or malicious or benign. The URL parameters contain Server name, DNS query time, TCP details and many more to verify those websites.

Tariro Manyumwa [5] et al. Contributed towards exploring malicious websites using URL based features in a multiclass classification problem. Here, aimed three URL attacks such as phishing, spam, and malware. These included priority features like URL features, bag of words segmentation and other word-based features.

Ankit Kumar Jain [6] et al. Discussed that analyzing the hyperlinks found in the HTML content of the website. Here, aimed various new hyperlinks of HTML tags with this help train the supervised algorithms evaluated the performance of those classifiers, from those classifiers logistic regression classifier has given highest accuracy. A website can be converted into a DOM (Document Object Model) tree and extracted the hyperlink features with the help of web crawler. The hyperlink has categorized into 12 groups like total hyperlink, no hyperlinks, internal hyperlinks, external hyperlinks, null hyperlinks, internal error, external error, internal redirect, external redirect, login form link, external/internal CSS, and external/internal favicon.

Chia-Mei Chen [14] et al. Discussed Blacklist and whitelist mechanism, Blacklist mechanisms are not reliable for blacking malicious URLs in social environment. In this paper proposed two types of anomaly features: domain anomaly and social anomaly features. In Domain anomaly features are used to identify malicious domains based on lexical features. In social anomaly features represent anomalous user behaviors in social communications.

Wenchuan Yang [10] et al. introduced a neural network model Convolutional Gated-Recurrent-Unit (CGRU) for malicious URL detection. This model divided into three parts: 1. Keyword-Based URL Character Embedding 2. Feature Extraction Module 3. Classification Module. In Keyword-Based URL Character Embedding is used to map the original URL character into a low-dimensional vector, thereby encoding the original sequence as a two-dimensional floating-point matrix. In the character embedding, the malicious keyword in the URL is distinguished from the ordinary character. Such differentiation can highlight the key part in the URL, which is advantageous in allowing the feature detection module to extract the representative feature more quickly. In Feature Extraction Module is used the convolutional neural network to extract features on the abstract level of the URL and uses the GRU as a pooling layer, retaining the important features on the premise of preserving the context relationship. It uses a combination of different-length convolution windows to extract features more fully at each level. In Classification Module used to classify the detected features. In this model, a stochastic gradient descent is used to jointly optimize the model.

Seok-Jun Bu [13] et al. proposed an additional approach of deep learning with first-order logic programmed rules to insert the real-world restriction for phishing URL detection, designed weighting mechanism between the neural and logic components as β -discrepancy loss function.

3. PROPOSED METHODOLOGY

Malicious dataset which accounts to be 60% of the total dataset is used for the experiment, which is obtained from PhishTank (https://www.phishtank. com/) and remaining 40% benign URLs are obtained from Alexa (http:/www.alexa.com/). The number of collected URLs are around 11,000. The URLs in the data set are termed as primary URL's. The primary URL's may have other URLs embedded in it as a hyperlink, which are termed as secondary URLs. Class1 features are Lexical features that exist in the URLs (primary and secondary), such as length of URL, length of domain, dots, at the rate, double slash, underscore etc. The frequency of each special character is obtained and is explored in this experiment. The number of class 1 features considered in the experiment are 10. The URL is a webpage which is designed using different HTML tags. In this experiment we wish to find the frequency of each HTML tag used in designing the webpage. The frequency of the different HTML tags is referred as class2 features. Class2 features are HTML tags that exist in the URLs (primary and secondary), such as <a>, <abbr>, <b>, <href> etc. 99 class 2 features are considered in the experiment. The total number of features (class 1 and class2) are 109. 109 features are extracted from both primary and secondary URLs.

The figure1 depicts the proposed model for Identifying Malicious Websites Using Deep Learning.

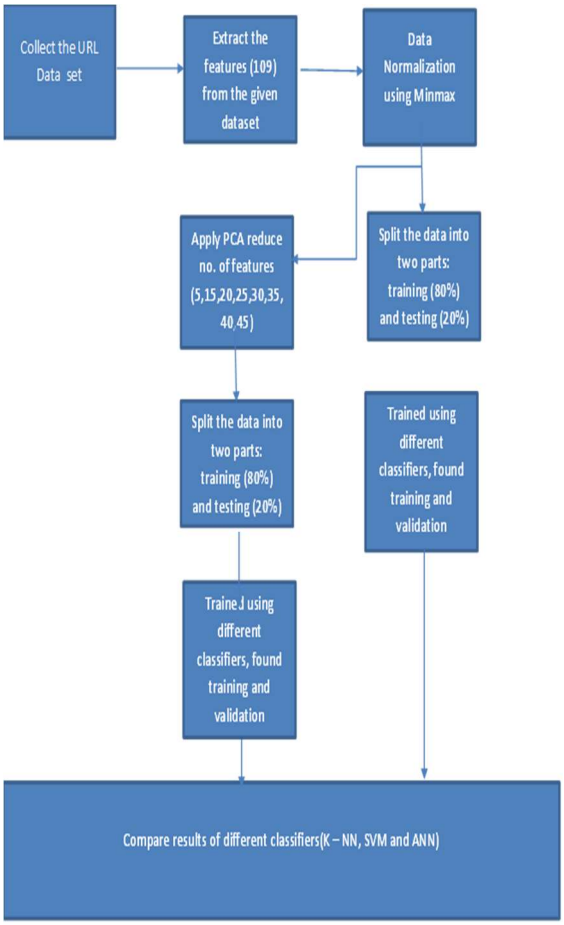

Figure 1: Block Diagram Of Identifying Malicious Websites

crawler method in python 3.6 is the best resource to collect the secondary URLs. The method *urllib.request.urlopen* retrieves the HTML tags from the given URL. 99 most frequently occurring tags are considered in the present experiment. Normalization is considered as a vital step in the pre-processing phase, all the features frequency are normalized. Min-Max normalization as given in the below equation (1) is applied to the obtained 109 features.

$$X_{Norm} = \frac{X - X\min}{X\max - X\min} \quad (1)$$

Where X_{max} and X_{min} are the max and min number of times the feature appeared. From the 11,000 URLs used 80% are used to train the model as training data, remaining 20% of the URLs are used as test data.

In the process of learning from the training data, Seven Layer Neural Network (SLNN) model is proposed, which uses 109 features. Two different optimizer algorithms are explored namely *Gradient Descendent Malware* Websites

and *ADAM Malware Websites.*

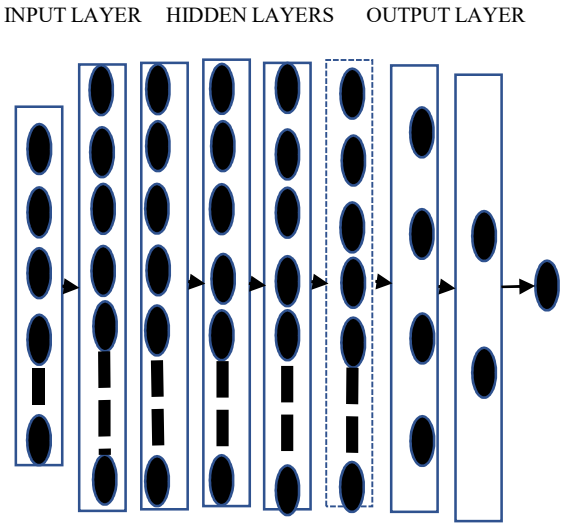

Figure 2: Seven Layer Neural Network (Slnn) Model

In figure 2 the description of, to the input layer give all class 1 and class 2 features connections from input layer given to the next consecutive hidden layers, which is mesh type of connections. The output has only one neuron it has a binary classifier, it will give whether the output is malicious or benign website. This model trained for different weights using trained data. This model has seven hidden layers all the layers are consecutive, and it is feed forward. The weights are updated every time using back propagation until it converges.

3.1. Gradient Descendent Malware Websites (GDMW)

Independent variables of class1 features are 10 and class 2 features are 99 altogether 109 features as input to the model and dependent variable as either 0 as malicious or 1 as non-malicious. A mathematical function which as $f(X_1,X_2,X_3,\ldots\ldots\ldots..,X_{109})$, let us consider $f(X_1,X_2,X_3,\ldots\ldots\ldots..,X_{109})$ to be $W_1X_1+W_2X_2+W_3X_3+\ldots\ldots\ldots+W_{109}X_{109}+W_0$ take input as independent variables and single dependent variable as output variable. The output variable is malicious or benign website in this case. In SLNN model to minimize error for that used Gradient Descendent Malware Websites optimization algorithm. Gradient Descendent Malware Websites is small steps to reach minimum value for that used a function such as f (Θ, X), where Θ refers the coefficients and X refers to input variables. To find coefficients used to Gradient Descendent Malware Websites to minimize the loss error based upon adjust the coefficient values.

The following formulas were used in Gradient descendent Malware Websites:

$$w_{t+1} = w_t - \eta \nabla w_t \quad (2)$$

$$b_{t+1} = b_t - \eta \nabla b_t \quad (3)$$

where, $\nabla w_t = \partial L\ (w, b)\ \partial w$

at $w = w_t$, $b = b_t$,

$\nabla b_t = \partial L\ (w, b)\ \partial b$ at $w = w_t$, $b = b_t$

The equation (2) describes the proposed algorithm *gradient descent malware websites* (GDMW). w_{t+1} is the updated expected weight of convergence, while w_t represents the current position. In the process of obtaining the updated weights, $\eta \nabla w_t$ is subtracted to obtain new weight, where η learning rate and ∇w_t represents the direction of the steepest descent.

Algorithm 1: Gradient Descendent Malware Websites (GDMW) ()

1. $t \leftarrow 0$.
2. max iterations $\leftarrow$ 150;
3. while t < max iterations do
4. $w_{t+1} \leftarrow w_t - \eta \nabla w_t$;
5. $b_{t+1} \leftarrow b_t - \eta \nabla b_t$;
6. end

3.2. ADAM Malware Websites

To deal with the nightmare of higher dimensionality, a tool for dimension reduction is indispensable. PCA (Principal component analysis) is a standard dimensionality reduction tool which can be employed in the process of feature reduction without the loss of information due to feature reduction. It does so by creating new uncorrelated variables that successively maximize variance. Applying PCA reduced the features from 109 to 40 even though the accuracy obtained has not affected.

Update rule for Adam:

$$m_t = \beta 1 * m_{t-1} + (1 - \beta 1) * (\nabla w_t) \quad (4)$$

$$v_t = \beta 2 * v_{t-1} + (1 - \beta 2) * (\nabla w_t)^2$$
(5)

$$m_t = \frac{mt}{1-\beta_1^t}$$

$$v_t = \frac{vt}{1-\beta_2^t}$$

$$w_{t+1} = w_t - \eta \frac{m_t}{\sqrt{v_t + \in}} * m_t$$
(6)

In equation (6), w_{t+1} is the updated weight of convergence, w_t is the initial weight in the iteration, eta in the learning rate, m_t is exponential average of gradients along w_t and v_t exponential average of squares of gradients along w_t direction of the steepest descent.

Algorithm 2: ADAM Malware Websites ()

a. $m_0, v_0 \leftarrow 0$
b. while t = {1, …., T} do
   i. Evaluate gradients based on the probability objective at 't'

   instance of time

   ii. Update exponential moving 2nd moment
   iii. Update exponential moving 1st moment
   iv. Compute bias corrected moving average
   v. Evaluate SMA length to the nearest accuracy if it is

   tractable

   vi. Evaluate bias corrected moving 2nd movement
   vii. Update parameters with adaptive movement
c. else
   i. Update parameters with un-adaptive movement

END

This paper proposes Neural Network using ADAM optimizer, Neural Network using Gradient Descendent, SVM and K-NN classifiers, in this compare the results of all four classifiers found that Neural Network using ADAM optimizer has performed the rest of three algorithms.

4. RESULTS

The proposed model has given encouraging results which are furnished in this section. The evaluation parameters used for comparing various classifier models are False Malicious Rate (FMR), False Non-Malicious Rate (FNMR), Precision, Recall, F-score, and Accuracy along with some additional parameters.

Certain metrics which give the insights of the assessment of the classification models are explored. Among the model's malicious websites prediction, there could be websites which are malicious, this count is termed as True Malicious. The prediction of the model is malicious, but the actual is non- malicious, is called False Malicious. The expectation of the model is non-malicious, but the actual is malicious, is called False Non-Malicious. The prediction of the model is non-malicious, but the actual is non-malicious is called True Non-Malicious.

- **FMR (False Malicious Rate):** False Malicious Rate is measured to give the performance of the model. Which is the fraction of False Malicious and total Malicious websites.

$$False\ Malicious\ Rate = \frac{FalseMalicious}{FalseMalicious + TrueMalicious}$$

- **FNMR (False Non-Malicious Rate):** False Non-Malicious Rate is measured to give the performance of the model. which is the fraction of False Non-Malicious and total malicious and non-Malicious.

$$False\ Non-Malicious\ Rate = \frac{FalseNon-MaliciousRate}{TrueMalicious + FalseNon-Malicious}$$

- **Precision:** The Precision is one of the performance indicators of the model, this indicator gives the positive prediction of the model. Precision measures the total number of True-

Journal of Theoretical and Applied Information Technology
30th June 2022. Vol.100. No 12

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Malicious divided by total number of Malicious websites.

$$Precision = \frac{TrueMalicious}{TrueMaliciou + FalseMalicious}$$

- **Recall:** Recall is the measure the model ability to detect the True-Malicious, it calculates the ratio of the number of True-Malicious by True-malicious plus False Non-Malicious.

$$Recall = \frac{TrueMalicious}{TrueMalicous + FalseNon - Malicious}$$

- **F- 1 Score:** F-1 Score is the harmonic mean of precision and recall. It lies between 0 and 1, provides a simple way to compare classifiers.

$$F-1\,Score = \frac{2*TrueMalicious}{2*TrueMalicious + FalseMalicious + FalseNon-Malicious}$$

- **Accuracy (%):** Accuracy is the fraction of accurately distinguished webpages (both phish and benign) and total number of classified webpages.

$$Accuracy(\%) = \frac{TrueMlicius+TrueNon-Mlicius}{TrueMlicius+TrueNon-Mlicius+FalseMlicius+FalseNon-Mlicius} * 100$$

- **Confusion Matrix:** This Confusion Matrix, compares the actual malicious websites with predicted malicious websites. In the below Confusion Matrix X - axis labelled as Predicted malicious websites and Y- axis labelled as actual malicious websites.

Table1: Confusion Matrix

	Classifier Results		
Actual	**Class**	**Malicious**	**Non-Malicious**
	Malicious	TM	FNM
	Non-Malicious	FM	TNM

- **Receiver Operating Characteristic Curve (ROC):** ROC is a graph drawn by taking TMR on y-axis and FMR on x-axis, which gives the performance of the classifier. The classification predicts among the given URLs (Universal Resource locators) the malicious and non-malicious based on the URL and HTML tags. URLs are used in identifying a particular website, HTML tags are used in designing the webpages in websites. AUC (Area under the curve) metric gives the capacity of the model in classifying the given URLs into different categories based on the training dataset provided to the model in the process of learning.

4.1. DISCUSSION ON RESULTS

Three different classifiers KNN, SVM and proposed SLNN are explored as classification mechanism. A detailed study of different parameters of the above said classifiers are explored. Precision, Recall, Accuracy and F1-Score are standard metrics used as the performance measures of the classifier. Malicious URLs have more dangerous consequences. As recall is a vital metrics and other metrics explored are used for comparative study. To measure the performance analysis used standard performance analysis criteria like Confusion Matrix and Roc. The main objective is to highlight the reduction of dataset dimensionality and compare the performance for various sizes (5 to 40) of dominated features with respect to Accuracy and Number of epochs.
For the sake of an in-depth analysis the Receiver Operating Characteristic Curve (ROC), Confusion Matrix of each classification model.

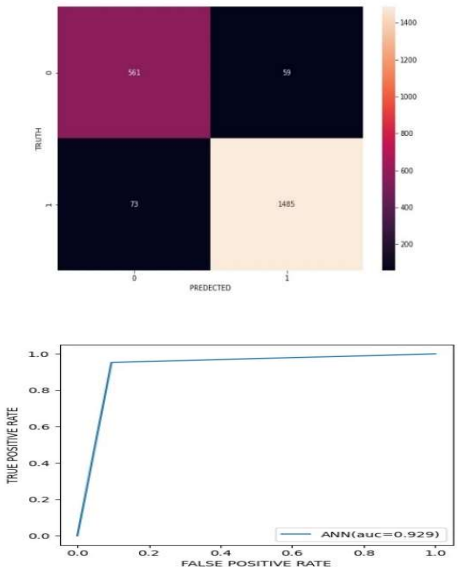

Figure 3: SLNN ADAM Malware Websites Confusion Matrix and Roc curve

Journal of Theoretical and Applied Information Technology
30th June 2022. Vol.100. No 12

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

In figure 3. shows out of the 2178 test cases, 1485 plus 73 the model has shown Malicious, the model correctly classified 1485 as Malicious and they are Malicious. In this case, 1485 true malicious or TM = 1485. The model predicted 73 are non-Malicious they are Malicious. The model incorrectly classified 73 as non-Malicious. In this case, 73 false non-malicious, or FNM = 73. Similarly, of 561 examples that were non-Malicious, this model predicted 561 are non-Malicious and they are non-Malicious. In this case, 561 true non-malicious correctly classified (561 true negatives or TNM = 561), and 59 were incorrectly classified (59 false positives, FM = 59).

In binary classification one has two class labels, in such binary classification Receiver Operator Characteristic (ROC) curve can be used as evaluation metric. In above figure the X-axis and Y-axis are shows False Malicious Rate and True Malicious Rate respectively. False Malicious Rate calculates total non-Malicious websites among total number of malicious and non-Malicious websites. True Malicious Rate calculates total Malicious websites out of total number of malicious and non-malicious websites. The ROC curve plots between TMR (95.31%) and FMR (9.51%). The Area Under the Curve explore the larger area covered the classifier between Malicious and non-Malicious. The maximum AUC (92.9%) measures the model performance.

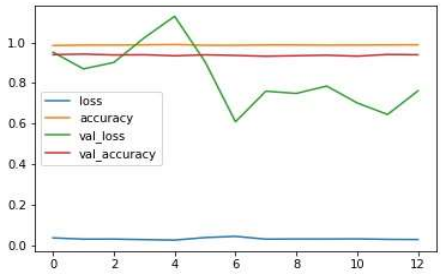

Figure 4: Graph of Loss function ADAM Malware Websites

Figure 4. shows that model error of the validation testing. Based upon this validation testing we should know the accuracy of our model. In above graph training accuracy is 98%, validation accuracy is 91%. To reduce the validation error used different types of regulation techniques are used.

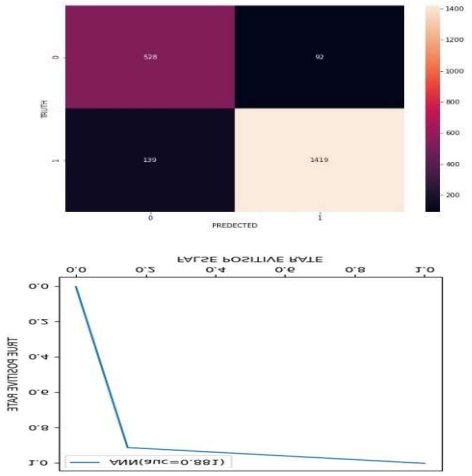

Figure 5: SLNN Gradient Descendent Malware Websites Confusion Matrix and Roc curve

In figure 5. shows out of the 2178 test cases 1419 plus 139 are Malicious, the model correctly classified 1419 as Malicious as originally, they are malicious. In this case, we say that we have 1419 true positives or TP = 1419. The model incorrectly classified 139 are non-malicious as they are malicious. In this case, we have 139 false negative, or FN = 139. Similarly, of 528 examples that were non-Malicious, 528 were correctly classified (528 true negatives or TN = 528), and 92 were incorrectly classified (92 false positives, FP = 92).

In binary classification problems Receiver Operator Characteristic (ROC) curve is regarded as evaluation metric. The curve gives the probability results of classification, which is plotted between TPR (91.07%) and FPR (14.83%). The integral value of the Curve (AUC) gives the assess ability of the classifier in the process of distinguish Malicious and non-Malicious websites. AUC is directly proportional to the accuracy of the classifier.

Journal of Theoretical and Applied Information Technology
30th June 2022. Vol.100. No 12

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

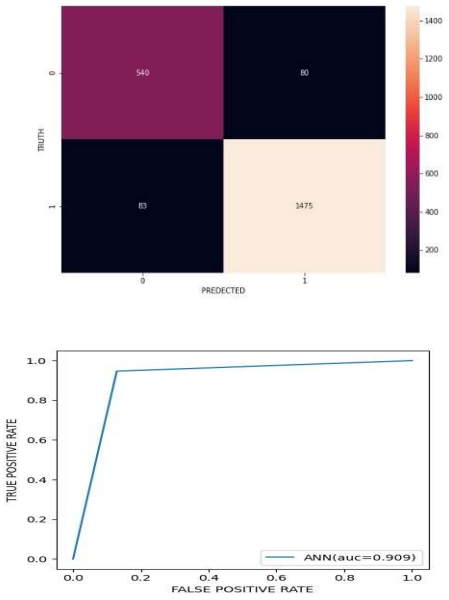

Figure 6: SLNN using PCA Confusion Matrix and ROC
curve

In figure 6. shows that of the 2178 test cases in that 1475 were Malicious, the model correctly classified 1475 as Malicious. In this case, we say that we have 1475 true positives or TP = 1475. The model incorrectly classified 83 examples as non-Malicious. In this case, we have 83 false negative, or FN = 83. Similarly, of 540 examples that were non-Malicious, 83 were correctly classified (83 true negatives or TN = 83), and 80 were incorrectly classified (80 false positives, FP = 80).

Table 2: Comparison Table of dominated features of PCA

Feature Reduction Technique	Dominated Features	Number of epochs	Accuracy
PCA	5	50	76
	10		82
	15		86
	20		87
	25		88
	30		90
	35		90
	40		92
	5	100	76
	10		82
	15		86
	20		87
	25		88
	30		90
	35		91
	40		92
	5	150	76
	10		82
	15		86
	20		87
	25		89
	30		90
	35		92
	40		93

In Table 2 results are tabulated, which gives as different dominated features with different epochs.

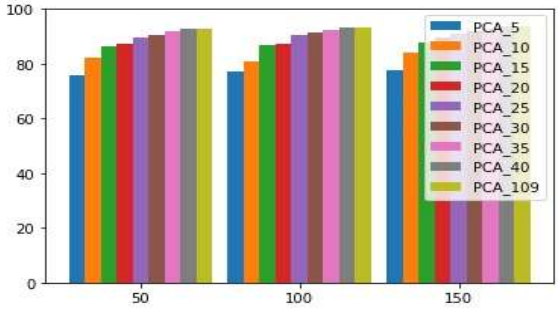

Figure 7: Comparison graph of dominated features of PCA

Principal component analysis (PCA) is a technique to reduce the number of dimensions in the given dataset. It increases interpretability with no loss of in the information but at reduce computational efforts. In above figure-7, X- axis represents the number of epochs while Y-axis represents accuracy of dominated features in every epoch. For different dominated features varying from 5 through 40, with step value 5 the experiment was repeated for different epoch values
of 50 through 150 with a step value of 50. The accuracy obtain with total features (109) was in-line with the accuracy using 40 dominated features. The accuracy obtained was 94%.

Journal of Theoretical and Applied Information Technology
30th June 2022. Vol.100. No 12

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 3: Comparison Table of Neural Network (GD), Neural Network (ADAM) and PCA

Models	Accuracy(A)	Precision (1)	Recall (1)	F-1Score (1)	Precision (0)	Recall (0)	F-1Score (0)
Neural Network (GD)	89	94	91	92	79	85	82
Neural Network (ADAM)	94	96	95	96	88	90	89
PCA	93	95	95	95	87	89	88

In Table 3 results has given for the experiment conducted by Neural Network (GD), Neural Network (ADAM) and PCA.

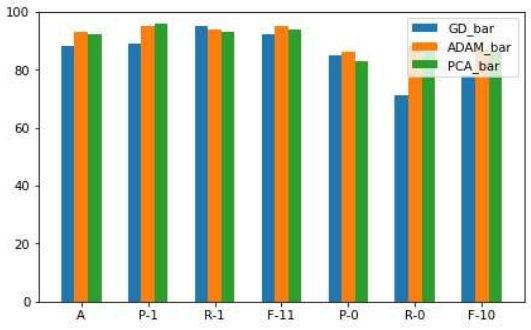

Figure 8: Comparison Graph of Neural Network (GD), Neural Network (ADAM) and PCA

Figure 8 provides comparison of performance metrics for different models. P-1, R-1,F-11 depicts the precision, Recall and F-1 score of malicious URL respectively. Similarly P-0,R-0 and F-10 gives the Precision , Recall and F-1 score of non-Malicious URL. A represents the accuracy of Neural Network (GD), Neural Network (ADAM) and PCA models.

Support Vector Machine (SVM):

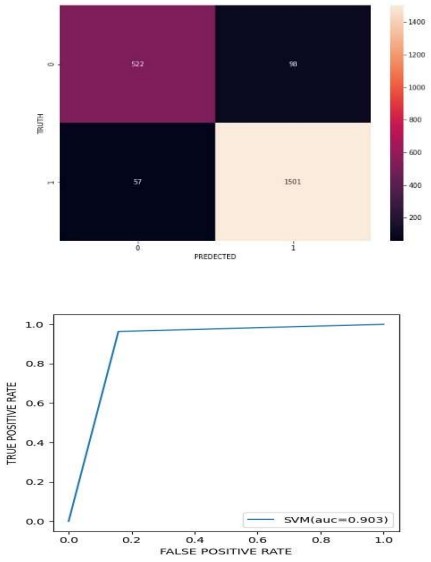

Figure 9a: Confusion Matrix and ROC of SVM

Figure 9 shows the results obtain using SVM. The test case has 2178 URLs, 1501 were found to be Malicious by the model, but the test cases have 1558 Malicious URLs. So, 1558-1501 (57) were mis-classified as non-Malicious. In other words, 1501 and 57 are classified as True Malicious and False Malicious respectively.

Similarly, out of 2178 test cases, 522 Non-Malicious URLs are identified by the model, but the test case set has 580 URLs which are non-Malicious. So, in conclusion 522 URLs were correctly classified (522 true non-malicious or TNM = 522), and the remaining 58 URLs were incorrectly classified (58 false non-malicious, FNM = 58).

Journal of Theoretical and Applied Information Technology
30th June 2022. Vol.100. No 12

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

K-Nearest Neighbor (KNN):

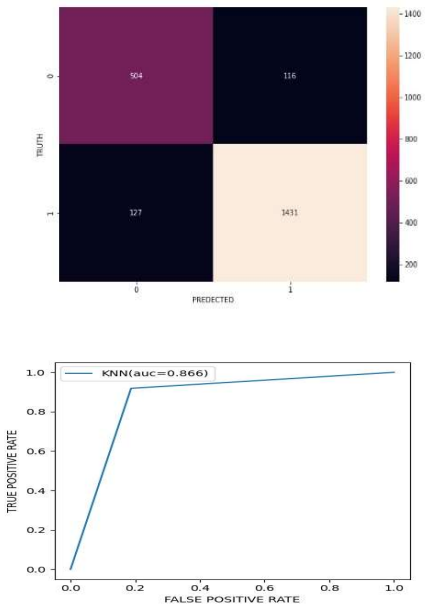

Figure 9b: Confusion Matrix and ROC of KNN

KNN model accuracy is given in figure 9, where among 2148 URLs, 1431 are found to be Malicious by the model. But the test case has 1558 Malicious URLs. In other words, 1431 and remaining 127 URLs are classified as True Malicious and False Malicious respectively.
Similarly, among the 620 non-Malicious URLs in the test dataset, 504 URLs were classified as non-Malicious by the model, rest of the 116 URLs are malicious with respect to the dataset, but the obtained classifier classified them as non-Malicious. In other words, 504 and 116 are classified as True non-Malicious and False non-Malicious respectively. ROC Results obtained for KNN model are TMR 91.84%, and FMR 1.87%. The AUC for KNN obtained is 86.6%.

Table 4: Results for different Classifiers

CLASSIFIER	PRECISION	RECALL	F1-SCORE	ACCURACY
KNN	93	92	92	89
SVM	94	96	95	93
SLNN GDMW	94	91	92	89
SLNN AMW	96	95	96	94

In Table 4, the performance of different models in the present study are provided. Comparisons of different measures such as precision, recall, f1-Score, and accuracy are tabulated in the table 2. From the tabulated metrics the overall performance of the proposed algorithm Neural Network using Adam optimizer has given encouraging results when compared to the other models.

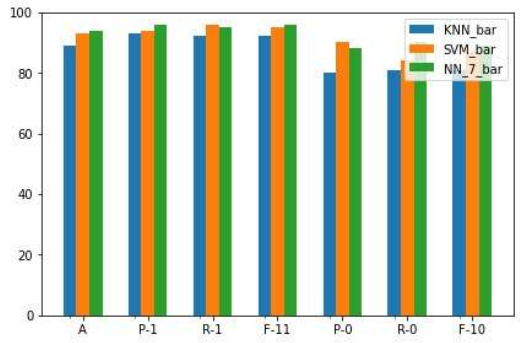

Figure 10: Comparison Graph of KNN, SVM and Neural Network

Figure 10 provides model accuracy comparisons of different metrics. P-1, R-1, F-11 depicts the precision, Recall and F-1 score of malicious URLs respectively on x-axis. In addition, P-0, R-0 and F-10 that give the Precision, Recall and F-1 score of non-Malicious URL are also shown on x-axis. The accuracy of the models in study are shown as legend '*A*' on x-axis of Neural Network (ADAM), SVM and KNN.

CONCLUSION:

In this paper, Web Classification is explored. One of the main objectives is to identify the dominated features and remove redundancy from the training dataset to save on computational cost. To deal with dimensionality reduction PCA is applied for different values of dominating features ranging from 5 through 40 with a step value of 5 in each iteration. In the process of web classification, a new classification model SLNN for identifying malicious website is proposed. The results obtained on applying PCA were almost like the results with all the features. To ascertain the results of the proposed model, the performance of SLNN was compared with SVM and KNN. The results obtained with SLNN were encouraging, the accuracy of SLNN obtained is 94% while SVM gave 93% and KNN gave 89% accuracy.

REFERENCES:

[1] Juan Carlos Prieto, Alberto Fernández-Isabel , Isaac Martín De Diego, Felipe Ortega , And Javier M. Moguerza , "Knowledge-Based Approach to Detect Potentially Risky Websites" , IEEE Access (Volume: 9)2021,Knowledge-Based Approach to Detect Potentially Risky Websites

[2] Shreyas Rajesh Labhsetwar, Piyush Aravind Kolte and Atharva Santhosh Sawant, "RakshaNet: URL - Aware Malicious Website Classifier", IEEE Access , 2021 2nd International Conference on Secure Cyber Computing and Communications (ICSCCC)

[3] Sanaa Kaddoura, "Classification of malicious and benign websites by network features using supervised machine learning algorithms", IEEE Access , 2021 5th Cyber Security in Networking Conference (CSNet)

[4] G. Kalyani, Dr. M.V.P.Chandra Sekhara Rao "Decision Tree Based Data Reconstruction for Privacy Preserving Classification Rule Mining" Informatica 41 289-304. 2017.

[5] Tariro Manyumwa; Phillip Francis Chapita; Hanlu Wu; Shouling Ji , "Towards Fighting Cybercrime: Malicious URL Attack Type Detection using Multiclass Classification", 2020 IEEE International Conference on Big Data (Big Data)

[6] Ankit Kumar Jain, B. B. Gupta, "A machine learning based approach for phishing detection using hyperlinks information", Journal of Ambient Intelligence and Humanized Computing (2019)

[7] Suleiman Y. Yerima and Mohammed K. Alzaylaee, "High Accuracy Phishing Detection Based on Convolutional Neural Networks", International Conference on Computer Applications & Information Security (ICCAIS 2020), 19-21 March, 2020.

[8] Carolin Jeeva and Elijah Blessing Rajsingh, "Intelligent phishing url detection using association rule mining", Human-centric Computing and Information Sciences, 2016.

[9] Samuel Ndichu, Sangwook Kim, Seiichi Ozawa, Tao Ban, Takeshi Takahashi and aisuke Inoue," Detecting Web-Based Attacks with SHAP and Tree Ensemble Machine Learning Methods", Applied Sciences(2022)

[10] Wenchuan Yang, Wen Zuo and Baojiang Cui, "Detecting Malicious URLs via a Keyword-Based Convolutional Gated-Recurrent-Unit Neural Network", IEEE Access (Volume: 7)2019.

[11] J . Heaton, " Artificial Intelligence for Humans: Deep learning and neural networks", Vol 3, Heaton Research, Incorporated, 323 pages, 2015

[12] Ahmet Selman Bozkir, Murat Aydos, "LogoSENSE: A companion HOG based logo detection scheme for phishing web page and E-mail brand recognition", Computers & Security, ELSEVIER, 2020

[13] Seok-Jun Bu and Sung-Bae Cho," Integrating Deep Learning with FIRST-ORDER LOGIC PROGRAMMED constraints for Zero-Day Phishing Attack Detection", ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)

[14] Chia-Mei Chen, D.J. Guan, Qun-Kai Su, "Feature Set Identification for Detecting Suspicious URLs using Bayesian Classification in Social Networks", Elsevier, Information Scieces 289 (2014), PG:133-147.

[15] Jyothy Mandala, M. V. P. Chandra Sekhara Rao,"Privacy preservation of data using crow search with adaptive awareness probability", Journal of Information Security and Applications 44 (2019) 157-169.