IJARCCE

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

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 9, Issue 6, June 2020

A Model to Detect Phishing Websites using Support Vector Classifier and a Deep Neural Network Algorithm

P.S. Ezekiel[1], O. E Taylor[2], F. B. Deedam-Okuchaba[3]

Department of Computer Science, Rivers State University, Port Harcourt, Nigeria[1,2,3]

Abstract: Phishing refers is the process whereby an attacker pretends to be a legitimate one for the purpose of getting vital information such as personal information, credit card details and confidential passwords from user. Phishing are usually done through websites Urls, emails, text messages and phone calls. Once they successfully acquire user's vital information, they used it in gaining access to the user's account which can to financial theft and loss. This paper presents a model in detecting phishing websites using support vector classifier and a deep neural network algorithm. We used a urlset dataset which comprises of 48,009 legitimate website Urls and 48,009 phishing Urls making a total of 98,019 websites Urls. The dataset was pre-processed by removing all Nan and finite values therefore making it clean and fit for training. After processing, we used feature extraction in deducting the dataset dimension and some unwanted feature columns thereby reducing the dataset from 16 feature columns to 2 feature columns; with the domain feature column (this holds the domain name/website Urls) and the label feature column (this holds the binary values 0 and 1, where 0 represent a legitimate website Url and 1 represent a phishing website). We also used CountVectorizer in converting text documents (domain column) to a vector of term/token counts. CountVectorizer also enables the pre-processing of text data prior to generating the vector representation. After training, support vector classifier showed that the result of accuracy was 97.21% while our deep learning algorithm was 98.33% of the total 98,018 url dataset studied. Thereafter we saved and deployed both models to web using flask

Keywords: Phishing, Support Vector Classifier, Deep Neural Network, Machine Learning.

I. INTRODUCTION

Phishing refers is the process whereby an attacker pretends to be a legitimate one for the purpose of getting vital information such as personal information, credit card details and confidential passwords from user. Phishing are usually done through websites Urls, emails, text messages and phone calls. Once they successfully acquire user's vital information, they used it in gaining access to the user's account which can to financial theft and loss. Phishing is also the act of disguising to be a trusted organization/company and sending fraudulent mails to individuals, sometimes the content of this mails claim that you have won some thousands of cash, requesting that you reply them by sending some confidential information like account details and credit card pin. It can also be defined as an act of circumventing or entrap security with an alias [1]. Phishing is the most unsafe criminal exercises in the cyber space; since almost everybody accesses the services made available by financial bodies such as banks online. Phishing attacks have increased rapidly over the years with almost records of everyday victims. Various methods are used by phishers to attack the vulnerable users such as messaging, VOIP, spoofed links and counterfeit websites. The reason for creating these counterfeit websites is to acquire private data from users like account numbers, login id, passwords of debit and credit card, etcetera. Phishing attacks can be prevented by detecting the websites and creating users' awareness to identify the phishing websites [2]. Users commonly have many user accounts on various websites including social network, email and also accounts for banking. Typically, phishing attack exploits social engineering in luring victims by sending them website Url link which redirects them to the phishing website. The fake link is created in form of a legitimate one. This link are most times sent via email to the user or be displayed in popular web pages [3].

Phishing attacks are constantly growing as online transactions and digital media is growing, Anti Phishing Working Group (APWG) reported that in Q2, 2017, in phishing most targeted industries are payment system 45% followed by Financial Institutions 16%, webmail 15%, and Cloud Storage 9% (Phishing Activity Trends Report, 2010). Communities which are online based as well as technical companies are working endlessly with vital information and they also use some security authentication codes in other to reduce phishing attacks. Phishers creates website to look identical as the original one in other to mislead customers/users to the fake site for the purpose of stealing crucial information. Despite the fact that most users are aware of online fraud and are educated on social engineering attacks, they still fall victim because social engineering attackers always come with new strategies of gaining access to confidential information [4].

DOI 10.17148/IJARCCE.2020.9632

II. RELATED WORK

Detection of Phishing Websites using Machine Learning Approach [4] developed a model that prevents users from being lured to phishing attack. They used three machine learning algorithms which are Linear model, Decision Tree, Random Forest and an Artificial Neural Architecture in detecting phishing websites on a dataset. They gathered their dataset from MillerSmiles archive, PhishTank archive and Google searching operators. The dataset consisted of 2456 instances and 30 features. Value of attributes was in the form of integer -1, 0, and 1, -1 represents phishing, 0 denotes suspicious and 1 denotes legitimate. First the dataset was processed to get mature data in desired format, then it was divided into two sections as training 70% and testing 30% respectively. They carried out their experiments using RStudio installed on windows 10. They compared their results was in terms of accuracy, error rate and precision. Random Forest had an accuracy of 95.7%, error rate of 4.3%, precision 93.7%, Linear Model had an accuracy of 92.10%, error rate of 8.0%, precision 92.00%, Neural Network an accuracy of 90.70%, error rate of 9.2%, precision 92.00%, Decision respectively. Also, Tree had an accuracy of 90.4%, error rate of 9.5%, precision 83.2%.

PhishGILLNET-phishing detection methodology using probabilistic latent semantic analysis, AdaBoost, and co-training [5], proposed a technique in detecting phishing attacks using vigorous server-side. This vigorous server-side technique integrates machine learning algorithm and natural language processing. This technique uses three layers approach in detecting phishing attacks. The first layer builds a topic model using Probabilistic Latent Semantic Analysis (PLSA), The second layer builds a vigorous classifier using AdaBoost and the third layer employs a classifier from both categorized and uncategorized example. They carried out some experiments using email data of about 400,000 emails. Their result shows that the third layer achieves F-measure of 100%.

A model of phishing attack detection was proposed in the paper, Detection of Phishing Website Using Machine Learning [6]. The proposed model focused on identifying the phishing attack based on checking phishing websites features, Blacklist and WHOIS database. Most features selected are used to distinguish between legitimate web pages and phishing web pages. They used two selected features which are domain name and website Urls in building and training their model. These used set of rules to identify website Urls of phishing webpages and that of website Urls of legitimate websites from their selected feature.

In the paper Phishing Webpage Detection for Secure Online Transactions, a novel approach for detecting phishing Websites was proposed [7]. proposed 3 layers in their model. They used Urls IP Address and Google Page Prank In the first layer. They used the website domain name in the second layer. They used quality website page in the third layer. They classify website page using fuzzy logic. They wrote fuzzy rules to assist the inference system in making logical wind-up about the legitimateness of the website page. In layer 2, there were 3 input values and 1 output value. In layer 3, there were 4 input values and one output value. The classification of their results was done using the weka data mining tool. The results for classification and integration of three layers were analyzed using graphical nodes in weka.

Web Phishing Detection Using a Deep Learning Framework [8] presented two techniques for detecting phishing website Urls. which are original feature column and interaction feature column. The original feature column is made up of website Url and domain name. The interacting feature has to do with the communications between websites. They used a real traffic flow dataset which includes traffic flow for 40 minutes and 24 hours. They constructed a graphical structure of traffic flow and analyzed the characteristics of web phishing from the view of the graph. They also introduced a Deep Belief Network (DBN) extract phishing features from a dataset. They selected Contrastive Divergence (CD) as training algorithm, which calculates the gradient through times of Gibbs Sampling. They analyzed their experiment by saying there are three parameters to affect the accuracy which are the number of DBN layer, the number of iterations per layer, and the number of nodes presents in the hidden layers. They first set the larger number of iterations T = 1000 and hidden layer nodes, such as layers = {top = 100, hidden = 50, . . ., 50, Vissible= 10}. In conclusion their DBN model achieve an approximately 90% true positive and 0.6% false positive.

Prevention of Phishing Attacks Based on Discriminative Key Point Features of Webpages [9], developed a technique which contrast images based on color values using image-based comparison. Only the company which created that website knows about the color range of the images present in the web page. None can design a fraudulent website which looks identical with the original website page having identical color range. So therefore, to have an accurate result, the images will be compared images using color values. They also presented an anti-phishing scheme using a particular key point features column in Website pages. Their technique achieves high accuracy and low error rates.

A Predictive Model for Phishing Detection [10] developed a model that enhance the performance of anti-phishing schemes using machine learning techniques. The machine learning model make use of selected module in establishing a good feature vector. They extracted the feature column from website Url pages. They made uses Naïve Bayes and Support Vector Machine which they trained on feature column set of size 15. There findings were established on a dataset of 2541 phishing instances and 2500 benign instances. They got 0.04% False Positive and 99.96% accuracy for both SVM and NB predictive models.

Phishing Detection Based on Machine Learning and Feature Selection Methods [11] employed a novel dataset related to phishing detection, which contained website pages of 5000 legitimate websites and 5000 fake website pages. They tested different machine learning algorithms in other to get a better result. They choose Multilayer perceptron Random forest, J48 as their training algorithm. They used different tools in other to enhance the performance of the model. Random Forest Algorithm gave the highest accuracy result which is about 98.11%.

III. DESIGN METHODOLOGY

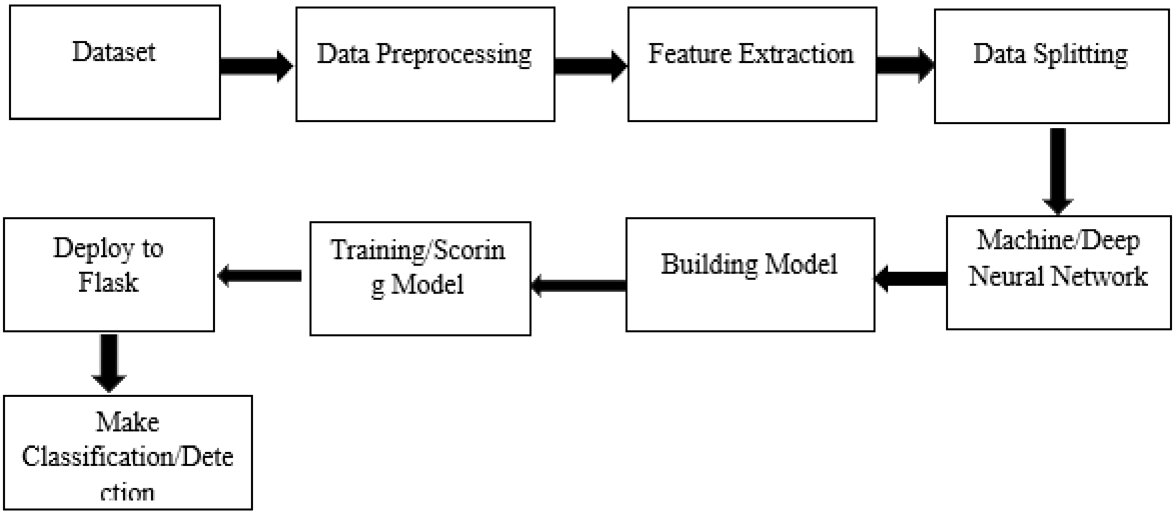


Fig. 1 Architectural Design of the proposed system

The design methodology and the system implementation process are as follows:
Dataset: we used a urlset dataset which contains 96,018 website urls. This dataset consists of 16 columns. The dataset contains 48,009 legitimate Urls websites and 48,009 phishing Urls websites. We processed the dataset by using just two feature columns which are the domain column and the label column. The domain column contains the websites Urls of both legitimate and phishing Urls while the label columns contains binary numbers (0,1) where 1 represent phishing website Urls and 0 represents legitimate website Urls. The dataset was created by [12].
Data pre-processing: This has to do with transforming the dataset into understandable format, and also removing inconsistency.
Feature Extraction: This has to do with the feature columns while accurately describing the dataset. We used feature extraction in reduction the dimention of the dataset from 16 columns to 2 columns. We also used **CountVectorizer** in converting a text documents (domain column) to a vector of term/token counts. CountVectorizer also enables the pre-processing of text data prior to generating the vector representation. This functionality makes it a highly flexible feature representation module for text.
Split Dataset into X_train and y_train: The datset was being splited into X_train and y_train variables which contains some array of numbers.
Machine /Deep Neural Network: We used a Support Vector Classifier as the machine learning model in training our model. We used Tensorflow framework and keras as our deep learning algorithm.
Building the Model: The model was built using Support Vector Classifier with SVC (kernel=i) where I represent the number of kernels (the set of mathematical functions) which are linear, poly, rbf and sigmoid. The model was trained using these four mathematical functions to have a better accuracy. The highest accuracy (97.21%) was found at linear, which is the first training step. We also trained thea deep learning algorithm with Keras and Tensorflow. The algorithm was made up of three connected layers and two hidden layers which we use ReLU as the activation and one output layer which we use Sigmoid as the activation. The network is made up of 123941 input shape.
Training/Scoring the Model: The deep neural network was trained using a batch_size=32 and an epoch value of 10 that is 10 iterative steps. The model was scored based on accuracy of about 98.33 on an epoch value of 10, that is having the highest accuracy at the 10th step of training/learning.
Deployed to Flask: we saved the Deep Neural Network model because it has the highest accuracy of about 98.33% while that of Machine Learning had 97.21%. The saved model was deployed to web using flask which is a web framework in python. We created a mini website where users can copy website Urls to check if it's a phishing one or not.
Make Classification/Detection: The trained/deployed model will collect input Urls from users and compare it with the Websites Urls of both legitimate and phishing urls it has in it's memory, if the collected input urls match the phishing urls, it classifies as phishing and if it matches the legitimate website urls it classifies it as legitimate website.

DOI 10.17148/IJARCCE.2020.9632

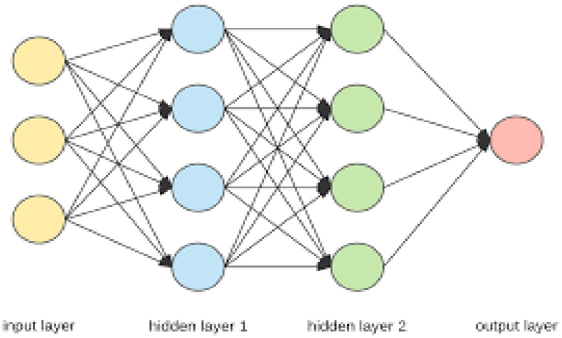


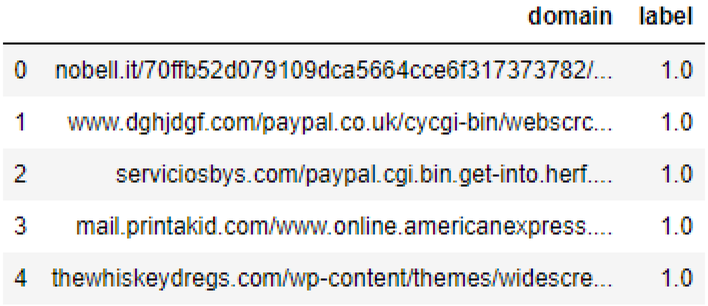
Fig. 2 Shows a Neural Network Architecture

This is a Neural Network Architecture with input neurons, three layers which comprises of two hidden layers and one output layer. Each neuron receives a set of x-values (numbered from 1 to n) as an input and computes the predicted y-hat value. Vector **x** actually contains the values of the features in one of *m* examples from the training set. The mathematical notation can be seen as follows:

$$Z= w1x1+w2x2+w3x3+\ldots WnXn = W^{T}-X \qquad \text{eqn. 1}$$

$$Z= W^{T}. x+b$$

$$Y^{/} = g(z) \qquad \text{eqn.2}$$



	domain	label
0	nobell.it/70ffb52d079109dca5664cce6f317373782/...	1.0
1	www.dghjdgf.com/paypal.co.uk/cycgi-bin/webscrc...	1.0
2	serviciosbys.com/paypal.cgi.bin.get-into.herf....	1.0
3	mail.printakid.com/www.online.americanexpress....	1.0
4	thewhiskeydregs.com/wp-content/themes/widescre...	1.0

Fig. 3 Shows training dataset

IV. RESULT AND DISCUSSION

In this paper, we used a urlset dataset which comprises of 48,009 legitimate website Urls and 48,009 phishing Urls making a total of 98,018 websites Urls. We pre-processed the dataset by removing all Nan and finite values therefore making it clean and fit for training.

label
label
1.08
1.04
1.00
0.96
0.92

Fig. 4 shows correlation matrix of the dataset

After processing, we used feature extraction in reducing the dataset dimension and some unwanted feature columns thereby reducing the dataset from 16 feature columns to 2 feature columns with the domain feature column (this holds the domain name/website Urls) and the label feature column (this holds the binary values 0 and 1 where 0 represent a legitimate website Url and 1 represent a phishing website. We also used CountVectorizer in converting a text documents

DOI 10.17148/IJARCCE.2020.9632

(domain column) to a vector of term/token counts. CountVectorizer also enables the pre-processing of text data prior to generating the vector representation. We splitted the dataset into X_train and y_train, X_test and y_test which holds 60% training data and 40% testing data. After data, we used support vector classifier and a deep neural network algorithm with a tensorflow framework and keras in training our model. We used support vector classifier with SVC(kernel=i) where i represents the mathematical functions in support vector classifier which are linear, poly, rbf and sigmoid. The support vector classifier had an accuracy of about 97.21%. at kernel=linear which is the first training step. We used a deep learning algorithm with a total input shape of 123941, batch_size of 32 and epochs value of 10. After training/learning for 10 iterative steps we had an accuracy of 98.33% out of the total trained model. We deployed our trained model (both support vector classifier model and deep learning model) to web using flask framework written in python to create mini website using Html, CSS and python for the backend to collect input Urls from users and classified them to be either legitimate Website Url or Phished Website Url.

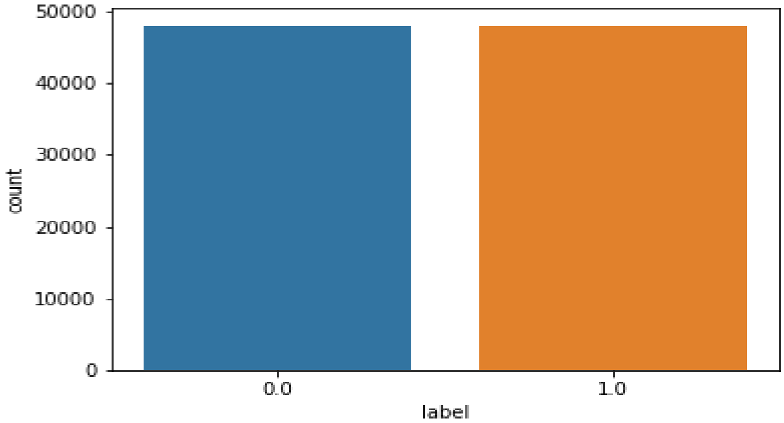


Fig. 5 shows countplot of legitimate and phishing where 1.0 represents phishing websites Urls and 0.0 represents legitimate website Urls

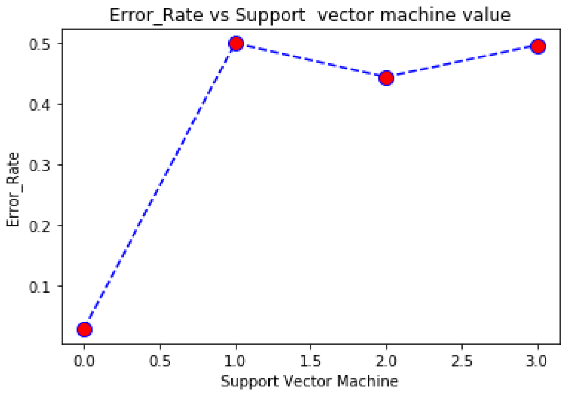


Fig. 6 shows error rate of support vector machine during training.

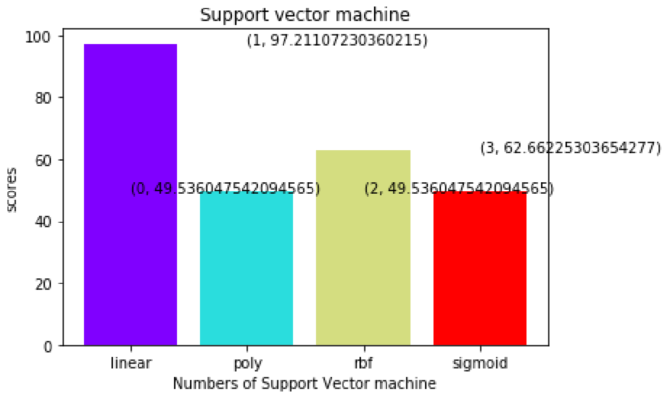
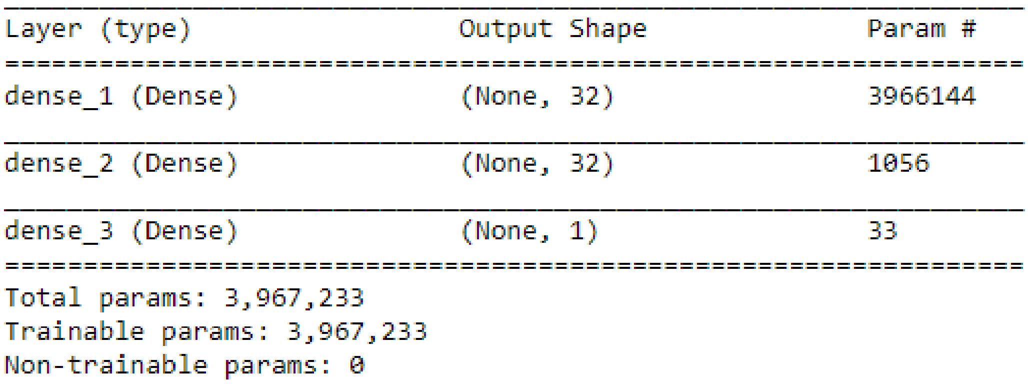
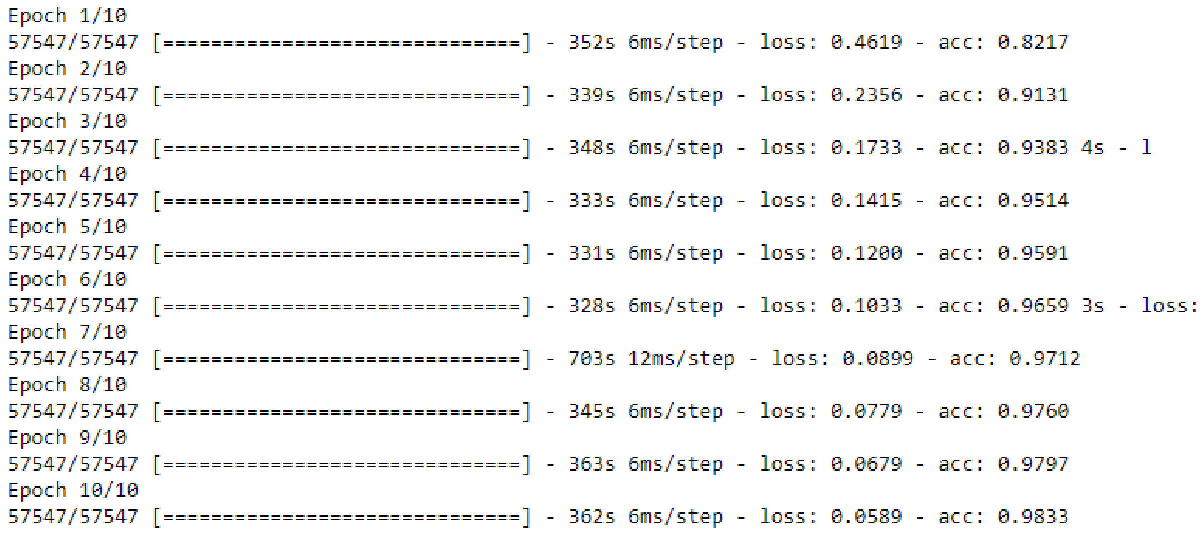


Fig. 7 shows accuracy of support vector machine at different kernel (mathematical functions).



```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_1 (Dense)              (None, 32)                3966144
_________________________________________________________________
dense_2 (Dense)              (None, 32)                1056
_________________________________________________________________
dense_3 (Dense)              (None, 1)                 33
=================================================================
Total params: 3,967,233
Trainable params: 3,967,233
Non-trainable params: 0
```

Fig. 8 shows the parameters used in the deep learning algorithm



```
Epoch 1/10
57547/57547 [==============================] - 352s 6ms/step - loss: 0.4619 - acc: 0.8217
Epoch 2/10
57547/57547 [==============================] - 339s 6ms/step - loss: 0.2356 - acc: 0.9131
Epoch 3/10
57547/57547 [==============================] - 348s 6ms/step - loss: 0.1733 - acc: 0.9383 4s - l
Epoch 4/10
57547/57547 [==============================] - 333s 6ms/step - loss: 0.1415 - acc: 0.9514
Epoch 5/10
57547/57547 [==============================] - 331s 6ms/step - loss: 0.1200 - acc: 0.9591
Epoch 6/10
57547/57547 [==============================] - 328s 6ms/step - loss: 0.1033 - acc: 0.9659 3s - loss:
Epoch 7/10
57547/57547 [==============================] - 703s 12ms/step - loss: 0.0899 - acc: 0.9712
Epoch 8/10
57547/57547 [==============================] - 345s 6ms/step - loss: 0.0779 - acc: 0.9760
Epoch 9/10
57547/57547 [==============================] - 363s 6ms/step - loss: 0.0679 - acc: 0.9797
Epoch 10/10
57547/57547 [==============================] - 362s 6ms/step - loss: 0.0589 - acc: 0.9833
```

Fig. 9 shows the training/learning process at 10 iterative steps.

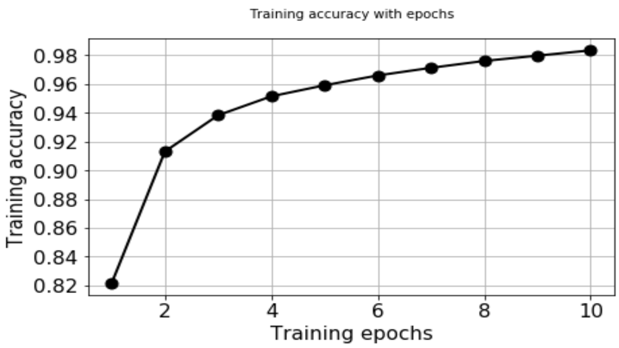


Fig. 10 shows training accuracy at different number of epochs which is the iterative steps.

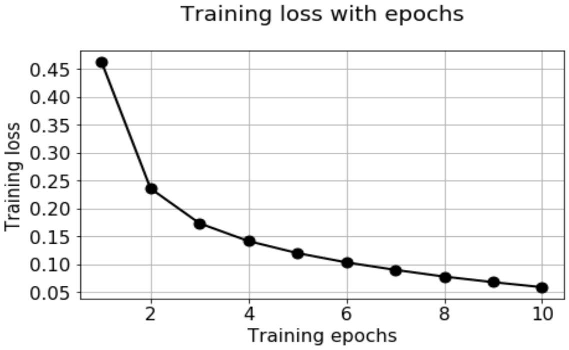
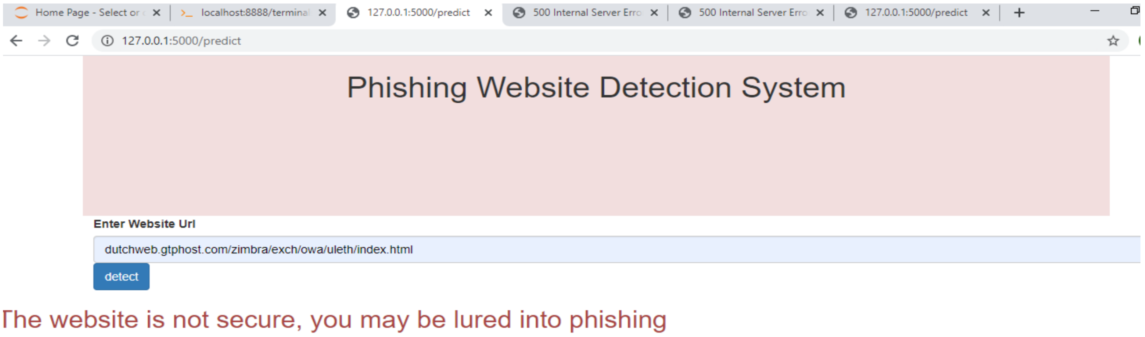


Fig. 11 shows the loss values at different number of epochs during training.



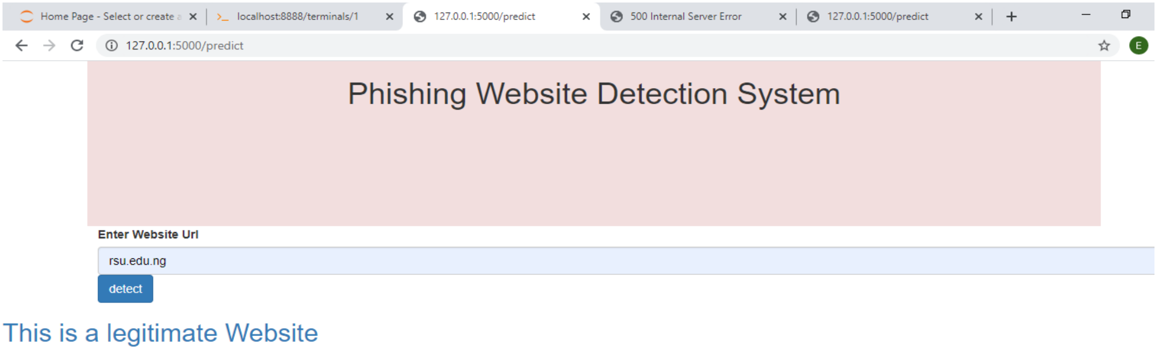
Phishing Website Detection System

Enter Website Url

dutchweb.gtphost.com/zimbra/exch/owa/uleth/index.html

The website is not secure, you may be lured into phishing

Fig. 12 shows a phishing url detected by our model in flask


Home Page - Select or create a
localhost:8888/terminals/1
127.0.0.1:5000/predict
500 Internal Server Error
127.0.0.1:5000/predict
127.0.0.1:5000/predict
detect

Phishing Website Detection System

Enter Website Url

rsu.edu.ng

This is a legitimate Website

Fig. 13 shows a legitimate Url detected by our model in flask

DOI 10.17148/IJARCCE.2020.9632

V. CONCLUSION

Phishing refers is the process whereby an attacker pretends to be a legitimate one for the purpose of getting vital information such as personal information, credit card details and confidential passwords from user. Phishing are usually done through websites Urls, emails, text messages and phone calls. Once they successfully acquire user's vital information, they used it in gaining access to the user's account which can to financial theft and loss. This paper presents a model to detect phishing websites using support vector classifier and a deep neural network algorithm. We used a urlset dataset which comprises of 48,009 legitimate website Urls and 48,009 phishing Urls making a total of 98,018 websites Urls. We preprocessed the dataset by removing all Nan and finite values therefore making it clean and fit for training. After processing, we used feature extraction in reducing the dataset dimension and some unwanted feature columns thereby reducing the dataset from 16 feature columns to 2 feature columns which the domain feature column (this holds the domain name/website Urls) and the label feature column (this holds the binary values 0 and 1 where 0 represent a legitimate website Url and 1 represent a phishing website. We also used CountVectorizer in converting a collection of text documents (domain column) to a vector of term/token counts. CountVectorizer also enables the pre-processing of text data prior to generating the vector representation. After training, support vector classifier gave us an accuracy of 97.21% and our deep learning algorithm gave us 98.33%, thereafter we saved and deploy both models to web using flask. This paper can also be extended by deploying the trained model as a chrome extension.

REFERENCES

[1]. P. Syiemlieh, G. M. Khongsit, U. M. Sharma, B. Sharma "Phishing-An Analysis on the Types, Causes, Preventive Measuresand Case Studies in the Current Situation" IOSR Journal of Computer Engineering (IOSR-JCE), pp. 01-08, 2015.

[2]. R. Kiruthiga, D. Akila "Phishing Websites Detection Using Machine Learning" International Journal of Recent Technology and Engineering (IJRTE), 8(11), pp. 111-114, 2019.

[3]. H. Sampat, M. Saharkar, A. Pandey, H. Lopes "Detection of Phishing Website Using Machine Learning" International Research Journal of Engineering and Technology (IRJET), 5(3) pp. 2527-2531, 2018.

[4]. J. Kahksha, S. Naaz "Detection of Phishing Websites using Machine Learning Approach" International Conference on Sustainable Computing in Science, Technology & Management, pp. 1774-1781, 2019.

[5]. V. Ramanathan, H. Wechsler "PhishGILLNET—phishing detection methodology using probabilistic latent semantic analysis, AdaBoost, and co-training" Journal on Information Security pp. 1-22, 2012.

[6]. H. Sampat, M. Saharkar, A. Pandey, H. Lopes "Detection of Phishing Website Using Machine Learning" International Research Journal of Engineering and Technology (IRJET) 5(3) pp. 2527-2531, 2018.

[7]. S. Sathish, A. Thirunavukarasu "Phishing Webpage Detection for Secure Online Transactions" International Journal of Computer Science and Network Security, 15(3) pp. 86-90, 2015.

[8]. P. Yi, Y. Guan, F. Zou, Y. Yao, W. Wang, T. Zhu "Web Phishing Detection Using a Deep Learning Framework" Wireless Communications and Mobile Computing pp.1-9, 2018.

[9]. M. Rajalingam, S. A. Alomari, P. Sumari "Prevention of Phishing Attacks Based on Discriminative Key Point Features of WebPages" International Journal of Computer Science and Security (IJCSS), 6(1) pp. 1-18, 2012.

[10]. A.A. Orunsolu , A.S. Sodiya , A.T. Akinwale "A predictive model for phishing detection" Journal of King Saud University – Computer and Information Sciences, 31(7), pp. 1-16 2019.

[11]. A. Mohammed, A. A. Zuraiq, M. Al-kasassbeh, N. Alnidami "Phishing Detection Based on Machine Learning and Feature Selection Methods" International Journal of Interactive Mobile Technologies, 13(12), pp. 171-183, 2019

[12]. S. Marchal, J. Francois, R. State, and T. Engel. PhishStorm: Detecting Phishing with Streaming Analytics. IEEE Transactions on Network and Service Management (TNSM), 11(4):458-471, 2014.

DOI 10.17148/IJARCCE.2020.9632