Detection of Phishing Websites Using Deep Learning Techniques

Md. Faisal Khan[a] and B. L. Rana[b]

[a,b] School of Science and Technology, Glocal University, Saharanpur, U.P. –
2471211, India. Email: mdfaisalkhan299@gmail.com, bansi.raina@rediffmail.com

Article History: Received: 10 January 2021; Revised: 12 February 2021; Accepted: 27 March 2021; Published online: 28 April 2021

Abstract: Phishing is a deceitful trick of cyber-attack designed and implemented by scammers and hackers with purpose of stealing personal data by impersonating the original websites. Phishing is like fishing in a lake wherein the users are very conveniently be fooled by scammers (phishers) by impersonating original websites and contents to leak their valuable personal and professional data. Currently a lot of anti phishing tools and techniques are being applied to detect and nullify the phishing cyber threat viz, heuristic feature, blacklist or white list and visual similarity-based approaches. In this research paper, we have anticipated robust and novel anti-phishing models via (I) Long Short-Term Memory (LSTM), (II) Deep-Neural Network (DNN) and (III) Convolution-Neural Network (CNN) using 10 features. The anticipated model achieves an accuracy of 98.67% for LSTM, 96.33% for DNN and 97.23% for CNN. The proposed techniques are highly efficient and robust which increases the phishing detection manifold.

Keywords: Phishing; Deep-neural network (DNN), Convolution-Neural network (CNN), Long short-term memory.

1. Introduction

Phishing can be expressed as a fraudulent technique to acquire or retrieve confidential personal data and related information by tricking a anonymous person into believing that the scammer is a righteous person and that the former person can trust upon him [1]. The easy prey of the scammers is careless and complacent individuals who are easily ensnared by the predators. Many a time a phishing attack may also be categorized as socially engineered attack or social attack. People are being targeted by phishing attacks on daily or regular basis. Phishing attacks have become more sophisticated and cynical in its nature which has evolved and increased manifold in recent years. The phishing attacks have become so much sophisticated that even some of world’s largest corporations are not immune and have been facing well over 1,000 attacks per month. According to a recent survey more than 65% organizations had to face phishing attacks in 2020; 30% of phishing messages had been opened by targeted users; in 2019, 32% of data threats were detected mainly due to phishing activity; phishing was the main culprit in 78% of cyber- espionage events; 51% of phishing attacks contained links to malware. An IBM report has revealed that the cost of a data breach due to phishing attack may go upto $4 million, however the stated figure alone is not sufficient enough to quantify the consequences and monitory losses due to phishing attacks. In US alone according to one of FBI’s internet crime reports US businesses had to incur more than $1.2 billion due to business email compromise attacks; scammers using fake gift cards, one form of spear phishing attack wherein the gift card is purportedly is sent which has been costing more than $ 70 million a year; another form of phishing attack is direct deposit phishing wherein the scammers are able to retrieve other person’s employee portal information thereby stealing their salaries which accounts for more than $100 million loss to businesses. It is important to know how phishing network works, figure 1 shows the process flow of phishing attack. The manner in which phishing attack takes place or phishing works has been discussed as below:

Planning

This initial step is used to collect confidential data of users in the form of e-mail lists, templates of scam pages as well as retrieving information from consumers of phishing identifications. Through various techniques and Trojan malwares the computers can easily be compromised (also known as Roots). Through various platforms the scammers get access to proof of notion exploits which enable the scammers to gain admittance to vulnerable computes.

Setup

The further steps involves ensuring the proper scam pages infrastructure on the compromised hosts used in the phishing attack.

Attack

There are millions of programs which have been written to handle mass mails, which enable a scammer to send out e-mails en masse using readily available right tools. The scammers learn through many online tutorials which provide easy explanations on how to send fake e-mails via different programs.

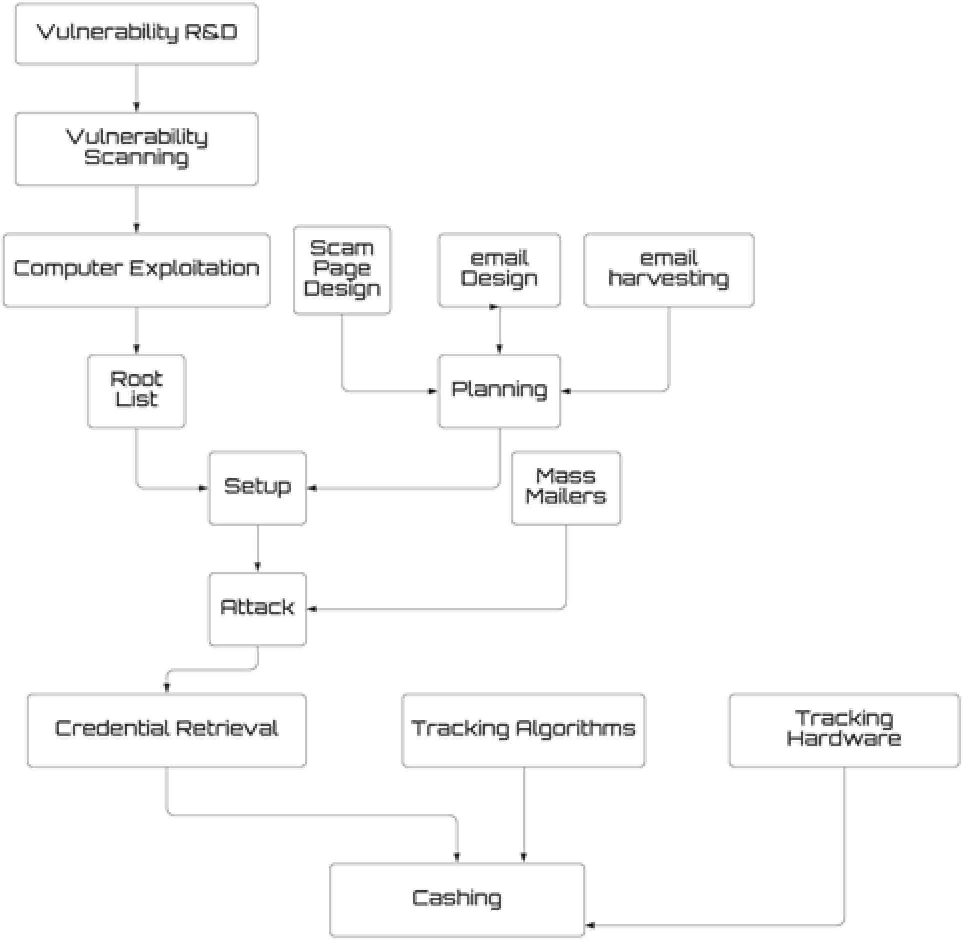


Figure 1 Process Flow of a Phishing Attack (Each step requires highly specialized skills from the Scammers involved in Phishing)

Collection

Time to time the phished information is sent to pseudonymous e-mail accounts using scam page hosting machines. A proxy server is used to retrieve this information.

Cashing

The phishers after retrieving the valuable credentials transfer the same credentials to Cashers. Now the Cashers use these phished identifications to attain money directly from the accounts involved to these credentials.

The number of methods have been established to detect and nullify phishing. The same techniques can be synopsized as follows.

- Deep-learning based detection Method: Deep learning method is a sub set of search engine or machine learning technique wherein the features are extracted directly from the given data. The data may include texts, sound or images. Deep learning method needs a huge amount of labeled data sets to enable Graphical Processing Unit (GPU) to train itself towards deep networks in a fraction of a minute. Availability of precisely curated data points has enabled high capacity supervised/discriminative deep learning techniques to farther working performance into its usability for myriad of applications. Many new techniques such as multi-layer feed forward network (MLFN) [2], Recurrent-Neural Network (RNN) [3] and Convolution-Neural Network (CNN) [4] have been employed to maneuver and exploit Deep-Neural Network (DNN) which is able to detect and nullify the malicious effects of phishing attacks. The above networks are trained using multi featured data sets which can be obtained through heuristic approaches. Le et al. [4] have proposed a CNN based deep neural network, URLNet for detection of malicious URLs. They have used character based and word based CNN techniques jointly optimizing both network techniques to detect phishing. They have also used more advanced word embedding techniques which enables to detect even rare words. The mentioned approach allowed URLNet to learn embeddings using never seen words allowing exploiting subword information.
- Heuristic-feature method based detection: This technique identifies phishing through prediction by using a set of features which exist either in page content or URL. It trains itself to classify a page as malicious. The heuristic technique has one major drawback wherein the detection rate decreases if the phishing web sites do

not have heuristic features moreover the technique can easily be bypassed if the related algorithms or uncovering features are notorious well in advance [5].

- Conventional-Machine learning based detection method: The accuracy to detect phishing using existing heuristic based techniques is not appropriate for practical use [6][7][8]. Nevertheless by developing innovative heuristics and improving the calculation algorithms, the detection accuracy can be improved many fold. In conventional-machine learning based detection technique the machine learning model is trained over prepared datasets; in practice these datasets are the extracted features using an innovative heuristic technique. The few algorithms used in conventional machine learning based detection of phishing are Support-Vector Decision Tree (SVDT), J48 Tree, Random-Forest (RF), Sequential-Minimum Optimization (SMO), Principal Component Analysis Random Forest (PCARF), Multilayer Perceptron, AdaBoost, Bayesian Additive Regression Trees (BART), Naïve Bayes (NB) etc. The algorithms are highly useful in detection of phishing attacks even if training data is large in number as these algorithms are able to learn possible variations that phishing sites may show.
- Visual-similarity method based detection: Phishing websites are illegitimate clone websites very similar to corresponding legitimate websites. Users are deceived to believe that they have been searching or browsing correct websites. The Visual similarity based phishing detection methods use the feature set alike text format, text content, HTML tags, image, Cascading-Style Sheet (CSS) etc. to make the verdict. These methods are being used to compare the apprehensive websites with those of legitimate websites by using numerous features; phishing is success, if the semblance is greater than the pre-defined threshold value [9].
- Listing method based detection: Internet browsers like Google, Mozilla, Chrome, Opera, Microsoft edge, etc. have maintained blocked and permitted URLs which are known as blacklist and whitelist URLs respectively. Sometimes in whitelist databases, legitimate websites are missing from the databases which might have been the victim and they are blocked being accessed through these browsers. Whereas in blacklist based databases phishing URLs are maintained. Blacklist based method fails when it encounters 0-day phishing sites. Updating the list more frequently may help overcome these problems, but it may seem really hectic work.

The advancement in machine learning techniques and their robustness which have been trained over millions of datasets has led to the development of many optimized methods which are able to deal with phishing sites very efficiently. Although the is advancement in phishing detection but still there are phishing sites which may last longer than a day, it means there must be a strong and more robust mechanism to avert and nullify phishing attacks with higher exactness. The more advanced heuristic methods capture specific features which are highly robust to detect even 0-day phishing websites.

In this research work, we applied Deep-Neural Network (DNN) deep learning algorithm for the detection of phishing sites. Here, we have utilized an information gain (IG) system to classify phishing websites by selecting best performing features. In the present work we have achieved high accuracy by selecting 10 best features.

2. Related Work

Mohammad et al [10] proposed self-structuring neural network model (ST-NNM) to predict phishing websites. They used 600 legitimate and 800 phishing websites to build their model using seventeen features extracted from URLs and the source code of these websites. To overcome the problem of overlifting they used “hold-Out” validation method by dividing their datasets into training, validation and testing datasets. They have used “log-sigmoid” activation function for all the layers. Zhao et al [11] have proposed a Gated-Recurrent Neural Network (GRNN) algorithm and compared with Random-Forest method (RFM) wherein the former outperformed the later model with well selected features. However the GRNN model requires more time to be trained along with optimization of system architecture to perform better.

Le et al [4] have used URLNet an end-to-end deep learning method to perceive phishing URLs. They have applied CNN method to both characters as well as words of the URL string to train the URLs by embedding in a mutually optimized agenda. They have shown that their model outperformed the existing models. The proposed model may fail if the phishing websites have very short URLs.

Feng et al [12] anticipated a more innovative phishing detection model applying neural network classification technique. By adopting design risk minimization principle and the model was able to attain very high accuracy along with having excellent generalization ability. The authors used Monte Carlo algorithm for training their model for the avoidance of over-fitting. Their proposed detection model can achieve a very high accuracy and a low FPR indicating the model’s excellence.

Yi et al [13] proposed website phishing detection applying a deep learning framework using original and interaction feature sets. They extracted the original features from the URL analysis and interaction features from the source codes of the websites. Later, they applied Deep Belief Network (DBN) to the extracted features achieving an accuracy satisfactory.

Sahingoz et al [14] applied a machine learning based phishing recognition technique from URLs. They have used different-different types of seven classification algorithms along with natural language processing created features. Through the experimental outcome it has been demonstrated that Random-Forest (RF) algorithm with Natural-language processing (NLP) based features is able to achieve a very high accuracy of 97.98% to detect phishing URLs.

Marchal et al [15] proposed "Off-the-Hook" a more robust application for the detection of phishing websites in real time as soon as they are visited by the web browsers. Off-the-Hook application preserves privacy of the users as well as it is able to identify the vulnerable websites being impersonated by phishing websites and immediately shoots warning to the user. To increase the detection rate the researchers have used Gradient Boosting algorithm to classify phishing websites. Faster decision making of "Off-the-Hook" application within milliseconds to block interaction with phishing websites and flashing warning in less than 2 seconds help users not to disclose their confidential data to these scammers.

Jain et al [16] applied the detection of phishing sites applying machine learning model at the client-side. The researchers identified 19 exceptionally predominant features to differentiate legitimate websites from those of phishing websites. The features were extracted from the URLs and the source code of the websites making the model faster, reliable and more intelligent giving accuracy.

El-Alfy [17] applied detection of phishing web based upon probabilistic-neural networks (P-NN) along with investigating the interaction of P-NN with K-medoids clustering to condense the intricacy in detection of phishing websites. To train the nodes, the proposed model combined both supervised and non-supervised algorithms. 11055 phishing websites were studied and their performance measures were evaluated giving an accuracy of more than 97% with low false errors.

Sophie et al [18] implemented decisive heuristics to detect phishing websites. 20 heuristic tests were selected based on the nature of phishing URLs and webpages. Phishark toolbar was used to implement the model testing the effectiveness of the heuristics. They showed that the combined effect of URL based and HTML based heuristics is appreciably more effective in detection of phishing websites. However, all heuristics are not at the same level in terms of detecting legitimate and phishing websites.

Aljofey et al [19] applied a comparatively fast-deep learning based model using character level Convolutional-neural network to detect phishing based upon the URLs of the websites. The planned model uses sequential pattern features to classify the legitimate URLs. Different feature sets such as hand-crafted, character level TF-IDF, character embedding and character level count vector features were evaluated and compared using traditional machine and deep learning methods. The model achieved more than 95% accuracy on the selected datasets and more than 98.5% accuracy on benchmark datasets.

3. Proposed Work

The aim is to detect malicious URLs using minimum features by applying deep machine learning techniques. Figure 2. shows the architectonics of the scheme being proposed which is comprised of feature extraction, selection and classification techniques. As an input web-page URLs are fed into the feature extractor. The feature extractor extracts the requisite features from the sources such as from URL, hyperlink and third party based and transfers them to Information Gain (IG) feature ranking algorithm. The IG algorithm supports in choosing the best performance features. The finest performance features are again trained over Deep Neural Network (DNN) to find out the output status and to differentiate between legitimate and phishing URLs.

A detailed explanation of the used model is as follows:

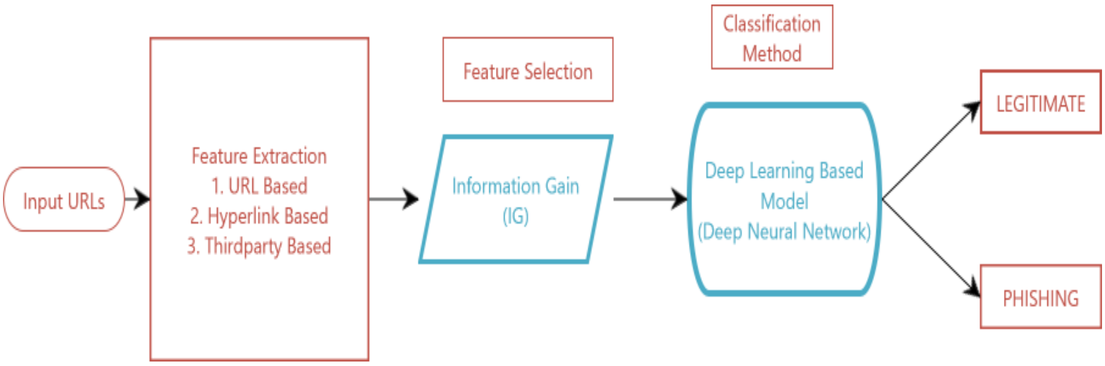


Figure 2 Architectonics of proposed model

3.1 Feature Extraction

The features have been extracted from URL obfuscation features, hyperlink-based features and third-party based features respectively. These features are extracted using Python with Selenium, a HTML parser and Beautiful soup for parsing the webpages. The choice of protruding features from the extracted features is carried out by using IG algorithm. In the below table 2 Rao and Pais [20] proposed the IG algorithm for the features.

3.1.1 URL obfuscation features

These are the appearances which are extracted directly from URLs, excluding website contents on third party services. Let us discuss the structure of a URL, a URL is a readable text, designed to replace IP addresses that computer systems use to interconnect with servers. A URL is easily able to determine and analyze a file structure on the given web-page. It consists of a etiquette, realm name and route having following format:

"protocol://domain-name.top-level-domain/path"

$$\underbrace{http://}_{\substack{\mathrm{Pr}\,otocol\\ Hyper\,Text\,Transfer}}\ \overset{Subdomain}{www.}\ \underbrace{shardauniversity}_{domainname}.\quad \overset{\substack{top\,level\,domain\\ TLD}}{com}$$

a) Scheme: Identification of the protocols to be used for accessing the resources freely available on the internet.

b) Hostname: Identification of the host where resource is located. It is a realm name which has been assigned to a host computer. A host name is a combination of host's local name with its parent domain's name. For example, www.glocaluniversity.com consists of host's machine name 'www' and the domain name 'glocaluniversity.com'.

c) Port Number: The servers frequently times bring more than one type of service, the browser must also tell the server the type of service required. The port number is used to make these requests for example the default port number for web service HTTP is 80 and HTTPS runs by default over port number 443.

d) Path: The specific resource which a web patron needs to access is identified by the path.

e) Query String: A inquiry string trails the trail component providing data information which can be used by the resource for some purpose. For example parameters for a search or data to be processed. The query string is a string of name and value pairs for example, term=bluebird&source=browser-search.

f) Fragment Identifier: A fragment identifier requires a location within page. It is presented by a hash character (#) which is an elective last part of a URL. For example http://foo/bar#frag, the string frag is the fragment identifier which can identify anything.

Rao and Pais have proposed 5 (UF1, UF2, UF3, UF4 and UF5) URL obfuscation features. Two features are ailing performance when analyzed applying IG algorithm as shown in the table 2. In our present work we have chosen the best 3 performance features, UF1 (dots in hostname), UF3 (lengthy URL) and UF5 (presence of HTTPS).

3.1.2 Hyperlink based features

The hyperlink based features have been retrieved from hyperlinks in the source code of the web page. A hyperlink is a unidirectional link in an electronic document connecting two different documents including various sections in the same document. Rao and Pais [20] have used 8 hyperlink based features for the detection of phishing. In our present work we have chosen only 6 greatest performance features out of these 8 features which are:

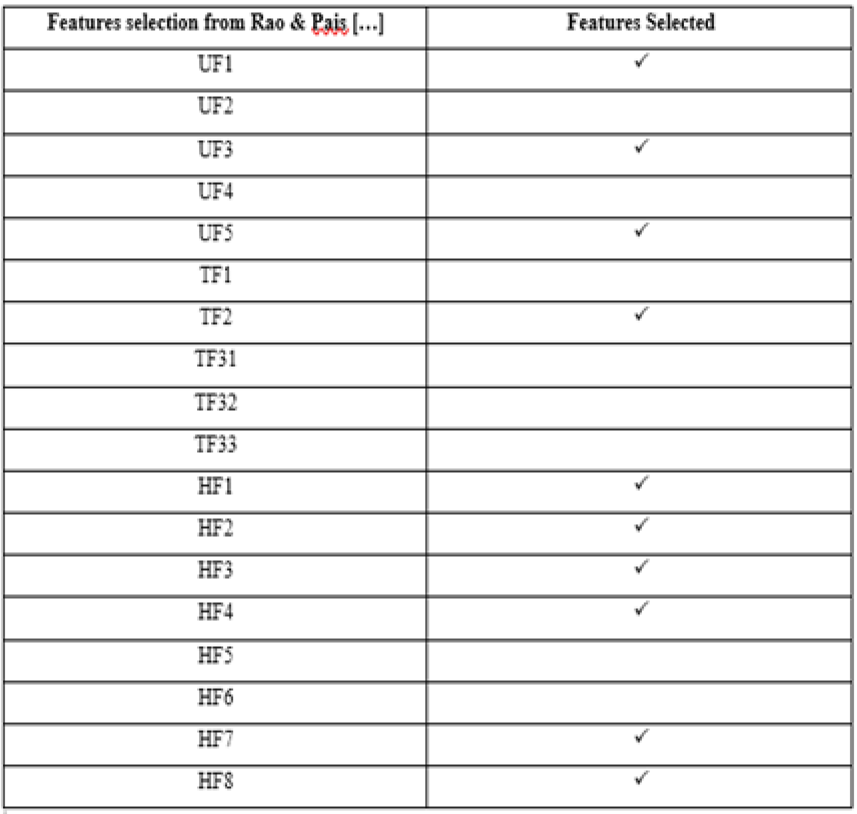
a. HF1 (presence of domain in anchor links)
b. HF2 (frequency of domain in image links)
c. HF3 (common page detection ratio)
d. HF4 (common page detection ratio in footer)
e. HF7 (presence of anchor links in HTML body)
f. HF8 (broken links ratio)

3.1.3 Third-party-based feature

Here, we have used Alexa to extract third party based features. We have excluded other third party based feature retrieving tools though they may perform well but this is just to condense the dependence on the 3rd party based amenities.

TF2:- Alexa rank is a 3rd party global ranking scheme which grades millions of websites in demand of their fame and thus can be utilized to detect and classify malicious phishing websites. The phishing websites are ranked low by Alexa which helps in classification and detection of phishing websites from the benign websites or high ranked websites. Combining a website's estimated traffic along with its visitor engagement in last three months, Alexa rank can be calculated. Estimation of traffic and engagement is performed from the data provided by the toolbar.

Table 3 is used to show the selected features, highlighting the above three features: Table 1 Features Selected



Features selection from Rao & Pais [...]	Features Selected
UF1	✓
UF2	
UF3	✓
UF4	
UF5	✓
TF1	
TF2	✓
TF31	
TF32	
TF33	
HF1	✓
HF2	✓
HF3	✓
HF4	✓
HF5	
HF6	
HF7	✓
HF8	✓

3.2 Feature selection

Information Gain (IG) has been used to filter out the predominant features from the given datasets and use it as a ranking tool. The ranking criterion has been used to analyze the relevance of these features which may help in classification and detection of malicious and phishing websites. The features which are mutually exclusive and dependent on class labels can be used to differentiate from those features which are independent of each other thus helping in detection of phishing websites. The (IG) is generally measured using the entropy of a system. The (IG) can measure the reduction in entropy of a system by splitting a dataset in accordance with a given value of a random variable. In this research work we have calculated gain over the entire available dataset for the ranking of the features. The significance and relevance of these features can be evaluated quantitatively by calculating the gain of each feature independently. The following two steps are used to calculate the gain of a given feature:

1) Entropy (class-label) calculated for the whole dataset, can be calculated from the below formula:

$$\text{info}(S)=\sum_{i=1}^{k} p_i \times \log_2 p_i \qquad (1)$$

where k = 2, the total unique number of class labels (phishing, legitimate) and S denotes a feature of the data set. Therefore, each feature carry few occasions fitting to one class and the rest to some other classes, pi shows the possibility of occurrences of S which fit in to the ith class. Probability pi can be calculated by enumeration the number of occurrences of S. After calculating pi for all i, the above equation is used to compute the entropy of S.

2) Calculation of conditional entropy for each inimitable value of that definite feature:

Conditional entropy wants an occurrence count of the class label. by feature value for its calculation. The feature value may be discrete or continuous.

i. Calculation of discrete-valued features:

$$\text{info}_A(S)=\sum_{i=1}^{y} \left|S_i\right| / \left|S\right| * \text{info}\left(S_i\right) \qquad (2)$$

where y is equal to the entire unique distinct values contained in feature value. Si shows a sum of ith type of value in feature and S is the total amount of feature value.

ii. To calculate the continuous-valued features, feature values have been segregated and divided into bins, the value accounts for the sum of the feature values. Since have different classes; can be termed as discrete-valued features. Hence, equation 2nd can be used to compute conditional entropy of specific features.

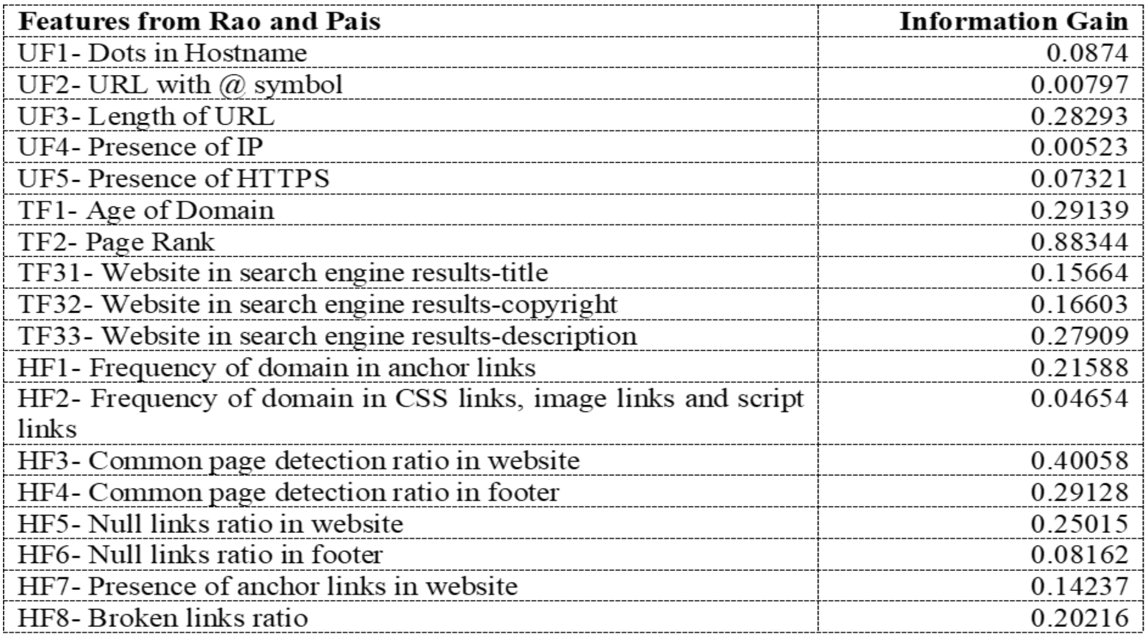
The calculation of the IG algorithmcan be carried out using the following formula:

$$\mathrm{IG}(\mathrm{A}) = \mathrm{info}(\mathrm{S})\text{-}\mathrm{info}_{\mathrm{A}}(\mathrm{S}) \qquad (3)$$

Table. 2

Techniques	**Image-based phishing**	**Common page-based phishing**	**Language independence**	**Broken links**	**Models**	**Features**
Mohammad *et al*	No	No	YES	NO	Neural Network	17
Zhao *et al*	No	No	YES	NO	Gated Recurrent Neural Network	Direct URLs
Le *at al*	No	No	YES	NO	Convolution Neural Network	Direct URLs
Feng *et al*	Yes	Yes	YES	NO	Neural Network	30
Yi *et al*	Yes	No	YES	YES	Deep Learning DBN	10
Sahingoz *et al*	No	Yes	YES	NO	Machine learning	Direct URLs
Jain *et al*	No	Yes	YES	NO	Machine learning	19
El-Alfy	Yes	Yes	YES	NO	PNN K-medoid clustering	30
Aljofey *et al*	Yes	Yes	YES	NO	Convolutional Neural Network	Direct URLs
Rao & Pais	Yes	Yes	YES	YES	Machine learning	18
Proposed Model	Yes	Yes	YES	YES	Deep Learning DNN	10

Table 3



Features from Rao and Pais	**Information Gain**
UF1- Dots in Hostname	0.0874
UF2- URL with @ symbol	0.00797
UF3- Length of URL	0.28293
UF4- Presence of IP	0.00523
UF5- Presence of HTTPS	0.07321
TF1- Age of Domain	0.29139
TF2- Page Rank	0.88344
TF31- Website in search engine results-title	0.15664
TF32- Website in search engine results-copyright	0.16603
TF33- Website in search engine results-description	0.27909
HF1- Frequency of domain in anchor links	0.21588
HF2- Frequency of domain in CSS links, image links and script links	0.04654
HF3- Common page detection ratio in website	0.40058
HF4- Common page detection ratio in footer	0.29128
HF5- Null links ratio in website	0.25015
HF6- Null links ratio in footer	0.08162
HF7- Presence of anchor links in website	0.14237
HF8- Broken links ratio	0.20216

4. Implementation

We have trained and cross validated an anticipated deep learning based method to classify as legitimate URL. or phishing URL from the given a list of website URLs by using MATLAB URL filter, to get the information of the URLs. Also to download the data of the site, which is further used to analyze the source code to extract the required features. The extracted data sets are further observed manually for authentic URLs; unwanted and duplicates URLs are removed from “PhishTank” data set. This procedure is to evade legitimate sites being used as phishing and reduce the processing time by avoiding annoying comparisons.

4.1 Tools Used

Used a MATLAB programme extract all features using URL and URL content. From the OPENPHISH platform, we gathered phishing URLs and from the ALEXA databases, legitimate pages. Many of the critical features are extracted and processed in MAT files as these URLs are load as an inputs in the MATLAB scripts. The extracted features are then passed to deep learning algorithms which are able to guess whether or not a URL is a legitimate website or a phishing scheme. We have introduces a deep learning algorithm by using Deep Learning Toolbox from MATLAB, which facilitates parallel computation:

4.2 Datasets Selection

The data set have been used from Rao and Pais [20] for experiment purposes. The dataset contains 3000 instances out of which 2000 are phishing sites obtained from “PhishTank” and 1000 are from Alexa database. For the assessment of model’s results, 70% datasets divided for training sets and 30% data sets for testing..

4.3 Deep learning algorithms

The performance of feature dataset has been estimated by training and cross validating the feature sets against various different parametric combinations. In case of multilayer feedforward neural networks, maximum accuracy in classification of phishing websites can be achieved by gathering data. based upon feature datasets and then tuning them to the constraints. The attainment of right value ensures that the phishing websites can then be classified with maximum possibility. In our work we have used MATLAB programming to implement deep learning algorithms. After applying a number of hidden layer combinations, the deep neural network (DNN) having five hidden layers gave the best result. The proposed DNN is comprised of 8 layers with 6 unseen layers along with one input and one output layer. The Rectified Linear Unit has been applied for the standardization of all the layers.

4.3.1 Brief description of deep-neural network (DNN)

The typical architecture of DNN has been shown in figure 3. DNN is a subset of machine learning models wherein it is composed of several common neural network layers along with one input, one output and one or more hidden layers.

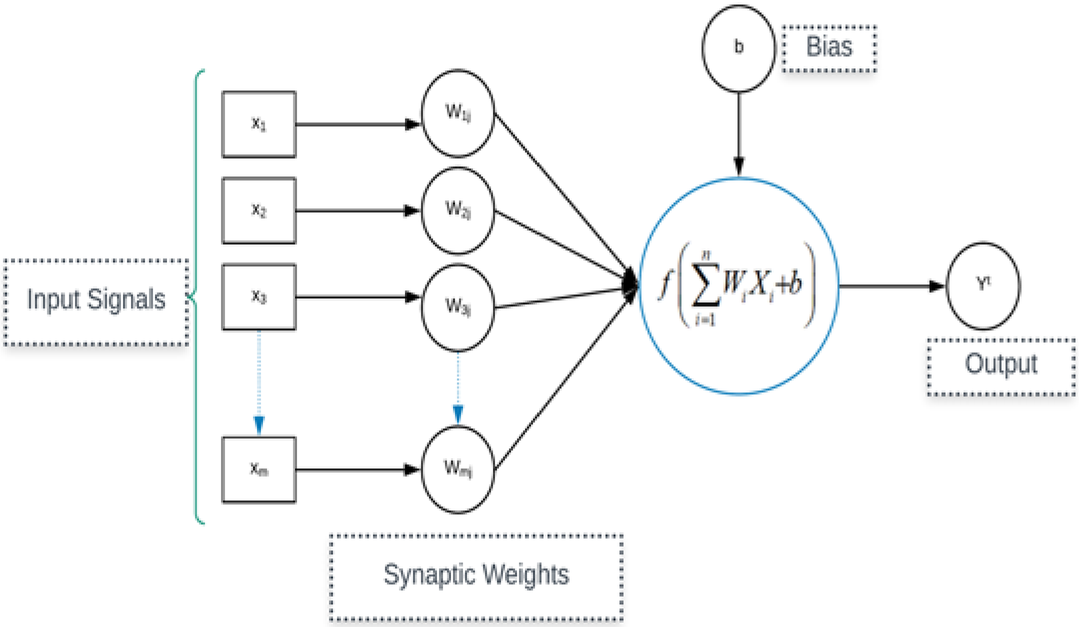


Figure 3 Architecture of Neuron

Neural networks (NN) are very complex structures made up of artificial neurons capable of mimicking the biological nerve cells (neurons). These artificial neurons can select in multiple inputs and it produces a single output. Hence, primary task of a neural network is to transform inputs into meaningful outputs. All the neurons in a neural network are capable of influencing each other and hence they may be termed as connected together. The neural network is highly efficient such that it is able to acknowledge and analyze every aspect of the dataset being handled along with filtering out data which are not related to each other. Thus these neural networks are highly proficient in finding enormously complex patterns in enormous measurements of data. The general mathematical representation of neuron's is:

$$Y^r = \Psi\left(\sum_{r=0}^{r=n} W_{rj} x_j + b_r\right) \qquad (4)$$

where Ψ is activation function, $W_r \in R^{L.B}$ is the weight function of r^th neuron and Y^r is the output of r^th neuron. The total number of neurons output layers generally dependent on number of desired output layers. The total number of neurons contained in hidden layers is hyperparameter which must be trained to obtain maximum result. Each and every deep-neural network is a complex mathematical function which is capable of adapting themselves according to the Nature of the data being analyzed.

Suppose, L = {0, 1, 2, 3, 4, 5} be the layers in our deep learning model. Y^{L-1} be the input to the layers {1, 2, 3, 4, 5} and Y^L is the output value of the layer. W^L is weight of layer j and has been used for linear transformation of inputs from r to n layers. B^L be the bias of the layer j and F^L be the related activated function of each layer. Y^0 be the input layer and $Y^{(L)}$ be the output layer.

$$\mathbf{Z^L = Y^{L-1} * W^L + B^L} \qquad \mathbf{(5)}$$

$$\mathbf{Y^L = F(Z^L)} \qquad \mathbf{(6)}$$

where * indicates matrix multiplication. Xavier initialization is taken to initialize W values, whereas B values has been initialized with 0. After, each iteration W and B have been updated using backpropagation algorithm. 0 layer is the input and 8 is output layer, 1-6 layers are hidden layers and are activated by the ReLU function, as shown in the below

$$Y_i^L = \begin{cases} o & if \quad Z_i^L \le o \\ Z_i^L & otherwise \end{cases} \qquad (7)$$

where i represents i^{th} iteration and L represents L^{th} layer. We used sigmoid activation function to compute intermediate output (Y^Θ) of the model as mentioned below:

$$Y^\Theta = \frac{1}{1+e^{-Z^L}} \qquad (8)$$

where L = 8.

The loss function $\left\{\ell\left(Y^\Theta, \hat{Y}\right)\right\}$ over the whole data set is the sum of cross entropy between the output and the actual output and can be described as below:

$$\ell\left(Y^\Theta, \hat{Y}\right) = \sum_{i=1}^{m}\left[\hat{y}_i * \log y_i^\Theta + (1-\hat{y}_i) * \log(1-y_i^\Theta)\right] \qquad (9)$$

Here, Y^Θ is intermediate output, $y_i^\Theta \in (0,1)$ is i^{th} row of Y^Θ whereas $\hat{Y}$ denotes to the labels of the datasets used and $\hat{y}_i \in \{0,1\}$ is i^{th} row of $\hat{Y}$. Here, the legitimate websites are denoted by 0 whereas the phishing websites are denoted by 1. Adam optimizer has been used to optimize the loss function.

5. Results and Discussions

We have appraise and investigate the performance of our DNN algorithms with different features and parameters. All the trials conducted with same data set of 3000 instances. Data was selected randomly from the available dataset by repeating each experiment. Accuracy and error rates respectively were used as the calculation metrics to test the concert of our model. Calculation has been performed by taking phishing websites as condition positive (ρ), ρ being the total number of phishing websites in the dataset and legitimate websites with condition negative (η). Wherein, η is the total number of legitimate websites in the data set. True positive (ρ_T) and true negative (η_T) represent accurately classified phishing websites and correctly classified legitimate websites.

i. Accuracy (A), calculated by the legitimacy and phishing rate to the total number of sites.

$$A = \frac{\rho_T + \eta_T}{\rho + \eta} \qquad (10)$$

ii. Error Rate (ER) defines the rate of legitimacy out of erroneously classified websites.

$$ER = 1 - \frac{\rho_T + \eta_T}{\rho + \eta} \qquad (11)$$

5.1 Validation of selected features using DNN

Deep neural network (DNN) has been used to validate feature selection and for the same we have performed three experiments; first being using all the 18 features mentioned by Rao and Pais. 98.13% accuracy is obtained by using all the 18 features. Table 4 shows the individual accuracy rate for the experiment. The individual accuracy rate varies from 61.06% to 97.07%. Two URL based features were detached whose accurateness rate was less than 66% along with two hyperlink based features (HF5 & HF6). The second experiment was performed after removing 8 features (UF2, UF4, TF1, TF31, TF32, TF33, HF5 and HF6) by minimalizing third party based features as well. Overall accuracy found 99.90% by selecting 10 best features. The experimental results clearly validate selection of features.

Experiment 1 – Deep neural network to evaluate individual heuristic features: Performance to each specific feature has been estimated and tabulated as shown in table number 4. Accuracy determination of individual contribution of each feature is important as to know the best performing features among them. Features having maximum accuracy in detection of phishing websites have more significance to class labels thus helping in the ranking process using information gain (IG) algorithm. The process will be used for the justification of inclusion and exclusion of individual features based on (IG). In this experiment, the overall accuracy obtained is 98.13%.

Experiment 2 – Exclusion of third party based features and evaluation of best performing features: Third party based feature extraction from URLs is a time taking process which is a hindrance in phishing website detection within a given time frame. In this experiment, all the third party based features were removed to appraise and analyze the robustness of the model; the accuracy plummeted below 90%. After this we included one, third party based feature (TF2) which has the maximum gain and attained an accuracy of 99.90% with 3100 epochs.

1.1 Results with DNN

In the previous section, the two experiments were carried out employing deep neural network (DNN) technique to validate the features. Now we have selected the ten best performing features; the individual accuracies of these ten best performing features have been shown in table 4. Experiment 3 has been performed employing deep neural network on the available data set from Rao and Pais [20]. Selecting, the ten features wherein the tuning of hyperparameters is made for the optimization of the model by choosing learning rate (r), number of hidden-layers, number of nodes, number of epochs and optimizer.

Experiment 4 – Tuning parameters to evaluate model: In this experiment to optimize the deep neural network (DNN) model the parameters were fine-tuned, which can be written as

1. ***Learning rate:-*** Started with r = 0.001 by keeping the other features as has been shown in table number 5, on this value the model's loss function started converging quickly which is shown in the figure 4. Herein we received the training accuracy is 99.71% and testing accuracy is 99.13%. Finally, we obtained 99.90% deep neural network training accuracy for chosen 3100 epochs as shown in figure 5.

Table 4: Individual features accuracy

Features	Training Accuracy (%)	Testing Accuracy (%)
UF1	66.82	66.08
UF2	61.23	61.06
UF3	80.01	79.34
UF4	62.14	61.48
UF5	68.28	67.43
TF1	78.51	78.44
TF2	97.28	97.07
TF31	83.62	83.35
TF32	77.26	77.01
TF33	66.41	66.08
HF1	73.62	73.51
HF2	66.24	65.88
HF3	84.41	84.07
HF4	81.48	80.52
HF5	77.02	76.52
HF6	64.73	64.08
HF7	70.2	69.88
HF8	78.82	78.30

2. Optimizer: Here we have used Adam optimizer to obtain training and testing accuracy.

Table 5: DNN Parameters

Layers	Number of units in Layers	Learning Rate (r)	Optimizer	Epochs	Activation Function
8	10,20,100,200,300,400,500,1	0.001	Adam Optimizer	3100	ReLU

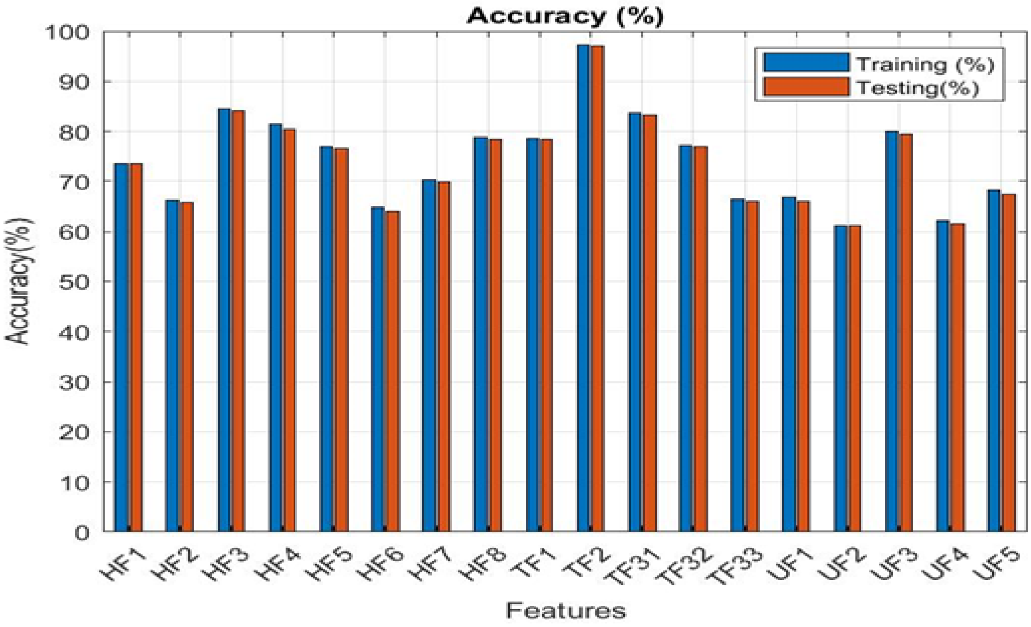
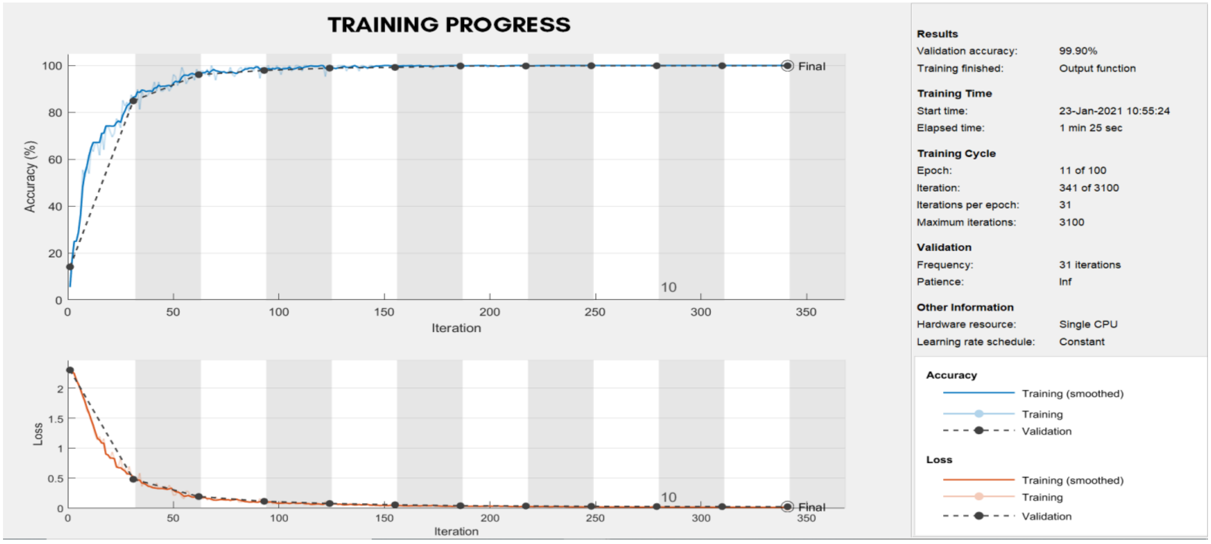


Figure 1 Individual feature accuracy using DNN



Results
Validation accuracy: 99.90%
Training finished: Output function

Training Time
Start time: 23-Jan-2021 10:55:24
Elapsed time: 1 min 25 sec

Training Cycle
Epoch: 11 of 100
Iteration: 341 of 3100
Iterations per epoch: 31
Maximum iterations: 3100

Validation
Frequency: 31 iterations
Patience: Inf

Other Information
Hardware resource: Single CPU
Learning rate schedule: Constant

Figure 5 DNN Training Progress with 10 features achieved 99.90% accuracy

3. Number of epochs: The iteration process was employed to find the total number of epochs to make our model perform better. We have started the iteration from 31 epochs and increased up-to 100 until minimum loss was obtained. Had we kept repeating the model, the loss might have continued to decrease to a minimum value, hence for the same reason we had to stop at that point.

4. Number of hidden layers: To remove network complexities only one hidden layer was used because increasing the hidden layers resulted in deteriorated performance of the model.

After conducting the aforementioned experiments, it can be concluded that deep neural network (DNN) is able to achieve a training accuracy of 99.90% with ten best performing features. The accuracy graph of best performing features using deep neural network (DNN) is shown in figure 5.

6. Limitations

The model is dependent on third party features and if these features are not available it will lead to limitations wherein the validation of the model may not be analyzed accurately. The proposed model may also fail to detect malicious phishing websites if these websites use embedded objects to replace texts.

7. Conclusions

Here, a robust system based on deep learning neural network (DNN) is proposed which is highly efficient in detecting phishing websites. To train the deep learning model, URL heuristics and third party based features have been used. Here we have minimized the number of features as compared to Rao and Pais [20], thereby reducing the dependence on third party based amenities which is able to attain an accuracy of 99.90%. In future we would like to use more heuristic features which may help in detection of phishing websites faster and more accurately even if the website includes embedded objects**.**

References

1. C. Abad. (2005). The economy of phishing: A survey of the operations of the phishing market. First Monday. [Online]. Available: https://firstmonday.org/ojs/index.php/fm/article/view/1272/1192.
2. Zhang Y.Y.N. (2011). Phishing Detection Using Neural Network. (pp. 95), [Online]. Available: http://cs229.stanford.edu/proj2012/ZhangYuan-PhishingDetectionUsingNeuralNetwork.pdf.
3. Bahnsen, A. C., Bohorquez, E.C., Villegas, S., Vargas, J. and Gonzalez, F. A. (2017). (pp. 1–8). "file:///C:/Users/somesha/Downloads/Deep Learning/10.1007%252Fs00521-017-3305-0.pdf," eCrime Res. Summit, eCrime.
4. Le, H., Pham, Q., Sahoo, D. and Hoi, S. C. H. (2018). URLNet: Learning a URL representation with deep learning for malicious URL detection. arXiv, no. i.
5. Da-Silva, C. M. R., Feitosa, E. L. and Garcia, V. C. (2020). Heuristic-based strategy for Phishing prediction: A survey of URL-based approach. Computional Security. 88(pp. 101613). doi: 10.1016/j.cose.2019.101613.
6. Zhang, Y., Hong, J. I. and Cranor, L. F. (2007). Cantina: A content-based approach to detecting phishing web sites. 16th Int. World Wide Web Conf. WWW2007, (pp. 639–648). doi: 10.1145/1242572.1242659.

7. Capgemini. (2019). Reinventing Cybersecurity with Artificial Intelligence.
8. Parveen, J. R. (2017). Neural Networks in Cyber Security. International Research Journal of Computer Science. (9)(4) (pp. 2015–2018).
9. Jain, A. K. and Gupta, B. B. (2017). Phishing detection: Analysis of visual similarity based approaches. Security Communication Networks, (2017)(1). doi: 10.1155/2017/5421046.
10. Mohammad, R. M., Thabtah, F. and McCluskey, L. (2014). Predicting phishing websites based on self-structuring neural network. Neural Computing Application. (25)(2), (pp. 443–458). doi: 10.1007/s00521-013-1490-z.
11. Zhao, J., Wang, N., Ma, Q. and Cheng, Z. (2019). Classifying malicious urls using gated recurrent neural networks. (773). Springer International Publishing.
12. Feng, F., Zhou, Q., Shen, Z., Yang, X., Han, L. and Wang, J. Q. (2018). The application of a novel neural network in the detection of phishing websites. Journal of Ambient Intelligent Humaniz Compuing. (pp. 1–15). doi: 10.1007/s12652-018-0786-3.
13. Yi, P., Guan, Y., Zou, F., Yao, ., Wang, W. and Zhu, T. (2018). Web phishing detection using a deep learning framework. Wireless Communication Mobile Computing. doi: 10.1155/2018/4678746.
14. Sahingoz, O. K., Buber, E.., Demir, O. and Diri, B. (2019). Machine learning based phishing detection from URLs. Expert System Applications.(117), (pp. 345–357). doi: 10.1016/j.eswa.2018.09.029.
15. Marchal, S., Armano, G., Grondahl, T., Saari, K., Singh, N. and Asokan, N. (2017). Off-the-hook: An efficient and usable client-side phishing prevention application. IEEE Transaction Computing. (66)(10), (pp. 1717–1733). doi: 10.1109/TC.2017.2703808.
16. Jain, A. K. and Gupta, B. B. (2018). Towards detection of phishing websites on client-side using machine learning based approach. Telecommunication System. (68)(4). (pp. 687–700). doi: 10.1007/s11235-017-0414-0.
17. El-Alfy, E.S. M. (2017). Detection of Phishing Websites Based on Probabilistic Neural Networks and K-Medoids Clustering. Computional Journal. (60)(12), (pp. 1745–1759). doi: 10.1093/comjnl/bxx035.
18. Gastellier-Prevost, S., Granadillo, G. G. and Laurent, M. (2011). Decisive heuristics to differentiate legitimate from phishing sites. 2011 Conference on. Network Inference. System and Security (SAR-SSI 2011), Proc.11. doi: 10.1109/SAR-SSI.2011.5931389.
19. Aljofey, A., Jiang, Q., Qu, Q., Huang, M. and Niyigena, J. P. (2020). An effective phishing detection model based on character level convolutional neural network from URL. Electron. (9)(9), (pp. 1–24). doi: 10.3390/electronics9091514.
20. Rao, R. S. and Pais, A. R. (2019). Detection of phishing websites using an efficient feature-based machine learning framework. Neural Computer Applications. (31)(8), pp. 3851–3873. doi: 10.1007/s00521-017-3305-0.