Modelling of an Adaptive Network Model for Phishing Website Detection Using Learning Approaches

Aldo Tenis*, Santhosh R.

Department of Computer Science and Engineering, Faculty of Engineering, Karpagam Academy of Higher Education, Coimbatore, Tamil Nadu, India
Emails: aldoteni@gmail.com; santhoshrd@gmail.com

Abstract

Phishing links are spread via text messages, social media platforms, and email by phishing attackers. Social engineering skills are used to visit phishing websites to trick the users and enter critical information related to personal data. The confidential data is stolen to defraud legitimate financial institutions or general websites for illegally attaining the benefits. Many machine learning-based solutions are in the enhancements and the technology of machine learning applications to detect the suggested phishing. The rules are used for a solution which depends on the extracted features, and few features require to lies on the services of third-party that, creating time-consuming and instability in the service of prediction. A deep learning-based framework is suggested to detect website of phishing. A framework is established to determine if there is a risk of phishing in real-time during the web page is visited by the user to give a message of warming by the browser plug-in. The prediction service in real-time merges the various techniques for enhancing the accuracy to lower the fake alarm rates and the time of computation which has the filtering whitelist, interception of the blacklist, and prediction of deep learning (DL). Various models of deep learning are compared using the different datasets in the module of machine learning prediction. The greatest accuracy is obtained as 99.18% by the adaptive Recurrent Neural Networks (a − RNN) model from the results of experiments to demonstrate the suggested feasibility solution.

Keywords: Phishing; legitimate; deep learning; prediction; false alarm rate

1. Introduction

Internet services take a tremendous change in the life of people. Many services related to online handles the users via the membership system, which every user requires for registering and logging in for personal services [1]. Hence, users require for giving the information related to personal during the suitable and effective services. The storage and securely transmitted data in the network environment are protected using network security technology. Moreover, many cyber criminals require different methodologies for stealing and attacking information related to personal data [2].

One of the cyberattack techniques is phishing which stimulates common websites to provide personal data to trick the users. The experts in network security used the technical methodologies for intercepting the attacks since the phishing attacks have been emergent for ten years. The improvement and changing of anti-attack technology and technology of attack constantly. There is no efficient technology for preventing attacks of phishing. The phishing attacks because of economic losses largely from network security reports in recent days [3]. There are above 100,000 links of phishing each month based on the APWG reports regarding the activity of phishing which has been an increasing trend in the last year [4]. The internet crime complaint centre shows the 2020 annual report causes the loss of economy generated using phishing attacks across $54 million [5].

Phishing links are spread through text messages, social media platforms and emails. The attacker edits the content and copies it via social engineering, which is interested in clicking the phishing link after obtaining the data. Hence, the risk is detected via network security technology and alerts the individual who is very efficient anti-attack

Doi: https://doi.org/10.54216/FPA.120213
Received: May 25, 2023 Revised: June 16, 2023 Accepted: July 17, 2023

technology to prevent the individual from leaking personal data when the link of phishing is used in the browser [6]. The rules are used for the core of conventional methodologies to detect URLs phishing. The rules are generated to summarize the webpage and URL source code's parsing features and compare the feature value with the empirical thresholds. The academic report of the research considered that the number of efficient rules in 100 [7] and [8]. The strategies of new attacks developed by cybercriminals depend on the rules. The rules' logic is restricted, and the rules are interpretable. Hence, the detection methods depend on the rules to crack easily and utilize the attackers. For instance, HTTPS is the schema of URL feature utilized in most of the study papers and attains greater importance. Moreover, the average of 83% of websites related to phishing shown by the report of APWG utilizes the schema of HTTPS in the 2021 first quarter [9].

More applications are being used in the cyber security field with the sudden growth of machine learning. Few experts and scholars are used to detect the links of phishing in the proposed system depending on machine learning, and the articles in many academic journals show the solutions based on machine learning to obtain greater accuracy [7] to [10]. Moreover, the real-time system needs the time of response for the predictive services in the environment of real-time scenario application, which are considered difficulties in milliseconds, and the user's trust and experience are affected by the higher rate of false positives. A deep learning-based framework is suggested for detecting the links of phishing in the environment of real-time web browsing. The browser plug-in is developed in the proposed system for obtaining the information related to clients, and the background services of prediction are called, and the results of prediction are shown to the users. The present page is received as the warming prompt when the URL of the present browser tab is predicted as the link of phishing [11]. The core prediction service obtains the prediction result to call the trained machine learning model. Various models are introduced, having the various data sets to compare and back up in the proposed system [12] – [15]. The experimental solutions conclude that the adaptive Recurrent Neural Networks (a – RNN) model achieves 99.18% accuracy, which is greater than Support Vector Machine, Random Forest, and Logistic Regression. The proposed system has contributions. They are:

1) The phishing URL is detected by the framework based on deep learning. The models are trained and tested with the help of seven created custom datasets from four existing data sources, and the greatest accuracy is obtained as the 99.18% having the model of (a – RNN).

2) The Chrome browser extension is the suggested framework to implement the prototype.

The work is structured as follows: Section 2 provides the extensive analysis of the various prevailing approaches; section 3 is a methodology, and the outcomes are provided in section 4. The outcomes are provided in section 5.

2. Related works

Phishing attacks represent a severe issue, and the technology is used to intercept and detect the attacks of phishing that evolve constantly. The fast and accurate method for filtering the better URLs via the phishing URLs of block and whitelist via the blacklist is discussed in [16]. Moreover, methods of the list do not detect the new links of phishing, and due to the less cost of creating the URL of phishing, the attacker does not lie on the similar link of phishing various times [17]. Many reports of research depend on machine learning to publish, and the greater accuracy of solutions is achieved from the experimental results. Moreover, many attacks of phishing victims in the environment of the actual network cause the loss of economy [18]. A specific gap exists between the real network security solutions and the experimental information results. Hence, anti-phishing solutions for real-time studying in the environment [19].

Li et al. [11] suggested a deep autoencoder model for detecting the zero-day attacks of phishing and achieving an accuracy of 97.34%. The character level features are extracted from the executed experiments, and the strings of URL on the three various datasets gathered from the ISCX-URL-2016 [12], Phish Tank [13], and Phish Storm [20]. The N-fold cross-validation and curve analysis of receiver-operating characteristics are used to determine the results of experiments. In the reconstruction phase, the root means square error (RMSE) is compared between the URLs phishing and legal URLs, and the increased RMSE is identified for the considerable phishing URL.

Deep learning models are introduced by Sherubha et al. [21] to detect only phishing websites with the help of ten extracted features from the third-party services and the HTML. Three deep learning models are compared and computed the 18 weights of features. The experiments' results demonstrate that the greatest accuracy of 99.57% is achieved by the Long Short Term Memory (LSTM) model. Moreover, one published dataset is used, having instances of 3526. The too-small dataset is taken for the training of deep learning. The experimental results show the greatest accuracy rate because of test data's poor diversity and uneven distribution [22].

Doi: https://doi.org/10.54216/FPA.120213
Received: May 25, 2023 Revised: June 16, 2023 Accepted: July 17, 2023

The long short-term memory (LSTM) algorithm and convolutional neural network (CNN) are combined by Sherubha et al. [23] to classify the websites of phishing. The hybrid classifier achieves an accuracy of 93.28%, and the image, text, and frame features are used as the average calculation time of 25s. The Common Crawl and Phish Tank gather the URLs, and the image features are extracted from the URLs. Image features are used for feeding the model of CNN offline, and the contributions of text features are done to the LSTM classifier [24]. The new solution is to merge the text and picture characteristics. Moreover, the improvement from the experimental outcomes in the rate of accuracy and the time of computation is to satisfy the requirements of prediction products in real-time [25] – [29].

An anti-phishing protection system was initiated by Sheng et al. [30] that comprises the extension of a web browser, filters, and detection plug-in for email, and the phishing detecting server based on machine learning. The browser extension extracts the present URL to obtain the screenshot and save the visiting history of the user as the client end profile. The below processes are used by the server for detecting the links of phishing. They are (i) the third-party services of allowlist and blocklist for filtering the new URLs, (ii) the machine learning model is used depending on the 13 features for the prediction if the URL is the link of phishing, and (iii) the technology of computer vision is used for detecting the logos of website and the same web pages screenshots are compared. The logo detector in the article uses 20 well-known online banks and a few websites.

An anti-phishing system is initiated by Han et al. [31] that has the extension of the web browser. The plug-in browser obtains the present URL in real-time, and the features are extracted depending on the structure of DOM and identify if there is a phishing attack risk and the user to be prompted. The detection service is classified into three phases called blacklist filtering, machine learning model based on prediction, and whitelist matching. The URL is determined in the prediction phase, which satisfies the situation as the link of phishing depends on the features, which is character level. Consider an instance where there is no hyperlink for the webpage, and the number of hyperlinks crosses the particular percentage to the external domain names. These rules are severe to the attackers, and few general URLs are unfound. Additionally, the researcher enhances the accuracy using the combination of three fundamental classification models.

EPDB is developed in [32] as the web browser architecture with the smart engine to detect phishing websites. The brilliant engine by the EPDB is integrated with the machine learning model to detect in the environment of real-time when compared with the conventional architectures of the web browser. The UCI dataset is used for training the models of machine learning. The extraction rule framework is used in the predictive process that extracts the 30 website features. The Random Forest classifier achieves the greatest accuracy of 99.36% by the experimental outcomes. The solution consists of few difficulties and restrictions even though the experimental information accuracy is very large. Primarily, the highly complex task is to develop the browser. Few browser functions are needed to be comfortable with the functions of the browser in a mature way before being promoted to users. Additionally, the model has the dataset to be trained as single, and the model has robustness, which requires verification. Lastly, the framework for the feature extraction based on rules lies in third-party services.

A website designed by Abiodun et al. [33] for checking the link is URL phishing. JSoup HTML Parser (JHP) library and JAVA programming language implement the detector. There are three stages in the solution. The JSoup parses the structure of DOM for the website to detect [34]. The number of link tags is analyzed from the structure of DOM, and the attribute is analyzed as the value of "href". The attribute value classifies external, internal, and empty links. The linked calculator is an indicator with a value between 0 and 1. The URL needs to verify to consider the link of phishing when the value is more than 0.8. There is no training process since there is no introduction to the machine learning model. The researchers utilized URLs 300 to test the experiment's link calculator performance [35]. The testing results obtain an accuracy of 99.97%, and the false-negative rate is 0.03. A huge test data set will be used to check the solution. There is a misguidance for judging the risk of phishing from the analysis using the characteristics analysis of the tag link from the source code of the website, and this is easy to utilize the rule the circumvent the rules for the attackers.

3. Methodology

This section provides an extensive analysis of the proposed model.

3.1. Data acquisition

The core of the machine learning field is data. The considerable data quantity and quality affect the module's machine learning-based performances. The system's foundation is the module of data collection. There are two parts to the task of data collection to attain the data from various sources of data and analyze and store the data. The data is gathered from various open sources presented in Tab 1. The dataset of Phish Storm comprises the 96,018 URLs, as 48,009 phishing and 48,009 legal URLs. 99.65% of phishing and 35378 legal URLs are available

Doi: https://doi.org/10.54216/FPA.120213
Received: May 25, 2023 Revised: June 16, 2023 Accepted: July 17, 2023

in the dataset of ISCX-URL2016 [12]. Around 350,000 benign URLs are loaded from the project of open Kaggle [23]. Additionally, around 400,000 data is gathered to grab the new data daily from the platform of Phish Tank [13] in the proposed system. The URLs basic structure is analyzed, and the basic data is parsed like the domain, protocol, top-level domain, path, and subdomain [24]. The major fields of the table called URL are presented. The data is stored in the relational database. It is effective and flexible to provide data services using the reading depending on SQL. The services of data combine the various data sets. Consider an instance to choose the 20,000 good links from Kaggle and 20,000 links of phishing from the phish tank to merge with the 40,000 instances to the balanced data set.

The module of machine learning is accountable for training and testing the model. The training model has the data to update in the regular frameworks, and all models' training and testing processes are triggered regularly and automatically. The system records every parameter of the run, and the types of data collection and the model are dumped into the file storage system. It has the flexibility of adding new models to the machine learning module.

3.2. Data loading

The dataset is utilized to obtain to train the model via the data service from the database. The flexible selection of various combinations of the data source is supported by the service of data and the datasets to vary the volumes of data. A label and the URL string are included in every instance of data to sign the URL is the legal link or the link of phishing. The values of the label are normalized between 0 and 1.

3.3. Feature extraction

The URL string is treated as the document with the semantics, and the technology of NLP (Natural language processing) is used for extracting the features. The feature extraction process transforms the text documents collection to the token count's matrix, and every token waits for one word. The tokenization process transforms the URL string into lists of words in the traditional models of machine learning. Hence, the feature number that equalizes the vocabulary size is identified using the data analysis. In the deep learning model, the tokenization process parses the URL string to the characters list as the tokens at the character level. The URL characters are obtained from the character set of ASCII. The most common 100 characters are chosen as the dictionary for the character set for the research.

There are 2083 characters as the maximum URL length. The maximum number of URL characters to 200 due to the time of calculation for the deep learning model and the statistical data is analyzed from the previous data set. Hence, every URL is converted to a matrix of $200*100$. The dictionary's position is relevant to every character is fixed as 1, and the balance values are 0.

3.4. Network configuration

The process of parameter configuration introduces the model's parameters based on the configuration file. The parameter grid is included by the configuration file relevant to every model, and every parameter contains the number of values in discrete ways. One of the combinations and permutations of the values of parameters are chosen for every training in the model's training process. The optimum parameter combination is achieved by comparing the models' accuracy when all the combinations are used for the model to complete the training.

3.5. Adaptive feature learning

The general notion of the auto encoder is introduced in the proposed system for the data reduction purpose. A tuple $m, h, n, R, T, S, X, Y, X', D$ defines the m/h/m auto-encoder in the mathematical form presented in Fig 1.

- set of real numbers is R.

- The positive integers are m and h to present the X and Z length. Case $0 < h < m$ is concerned with the data reduction.

- The mapping function from R^m to R^h is T.

- The mapping function from R^h to R^m is S.

- The set of n training vectors in R^m is $X = \{x_1, x_2, \ldots, x_n\}$.

- $X' = \{x_1', x_2', \ldots, x_n'\}$ denotes the equal set of target vectors in R^m and $Y = \{y_1, y_2, \ldots, y_n\}$ represents a set of n compressed vectors in R^h when the existence of external targets.

Doi: https://doi.org/10.54216/FPA.120213
Received: May 25, 2023 Revised: June 16, 2023 Accepted: July 17, 2023

- The function of dissimilarity or distortion, such as the Hamming distance or L_p norm, defined over R_m is D.

The auto encoder converts into output vectors $T \circ S(x_i) \in R^m$ from the input training vectors $x_i \in R_m$ for any transformations $T \in T$ and $S \in S$. The transformations biases and weights of S and T are learned by the auto encoder using the reduction of the function of distortion with the training vectors, which is defined below Eq. (1):

$$E(T,S) = \sum_{i=1}^{m} E(x_i) = \sum_{i=1}^{m} D(S(T(x_i; w, b); w', b'), \quad x_i) \qquad (1)$$

Here, b is the bias vector and w as the weight matrix is the T parameters. w' is the weight matrix and b' is the bias vector is the S parameters. Data compression and reduction are performed, and the data is projected by the auto encoder on the low dimension space when $h < m$. Various feature representations are attained that depend on the better option of mapping T and S, the D is the distortion function, and the usage of extra constraints like the regularization and generalization by using the regular auto encoder architecture. The numbers of auto encoders are stacked by defining the deep neural network. Non-linear sigmoid activation is used on the hidden layers for different classification issues. The auto encoders work like relevant T and S, which are the functions of linear transformation classes in the case of mappings to R_h from R_m. Hence, the matrices of the sizes $h \times m$ and $m \times h$ are T and S. The D is used for the linear transformation of R_m and R_h, the squared Euclidean distance designed. However, the complex-valued linear auto-encoders theory is presented.

3.6. Classifier model

The novel idea behind the suggested model is presented in this section. Thus, the suggested model of RNN is described for the intrusion detection of the network in a detailed way. The two public datasets are introduced briefly, which are utilized for evaluating the model. Lastly, the proposed system explains pre-processing the two datasets before describing the comparisons and the experiment's simulation using modern techniques.

The suggested model of RNN has the fundamental unit, the AE (auto-encoder). The feed-forward neural network is essential in auto-encoder that is the same as the MLP (multi-layer perceptron), which is comprised of three main layers called the output layer, input layer, and one or more hidden layer or layers with the numbers of neurons present in the output layer is same as the number of neurons present in the input layer. The auto-encoder has the learning approach, which is unsupervised learning due to the learning of abstract data representation, which is compressed by the auto-encoder to reconstruct the original input data rather than predicting the targeted output from the inputs. A simple auto-encoder has a typical architecture, shown in Fig 1, with a single hidden layer. The auto-encoder has two processes. They are (i) the process of encoding that performs between the hidden layer and input layer and (ii) the process of decoding that performs between the output layer and the hidden layer.

Consider the auto-encoder as the input layer, which has m nodes for h nodes in the hidden layer and m nodes for n input data vectors x_i for the equal set of y_i, abstract compressed data vectors.

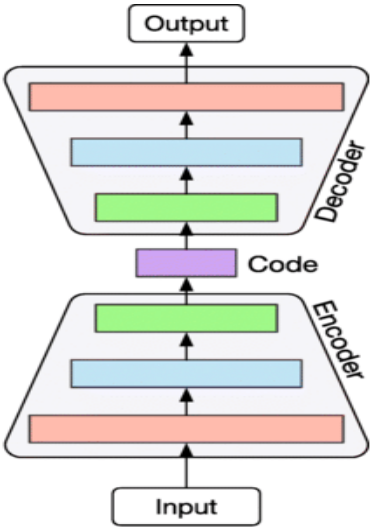

Figure 1: Auto-encoder

Doi: https://doi.org/10.54216/FPA.120213
Received: May 25, 2023 Revised: June 16, 2023 Accepted: July 17, 2023

The output layer has the number of nodes equivalent to the number of nodes present in the input layer, and x_i' is the reconstructed vector equivalent approximately to the input vectors presented below.

$$x_i \cong x_i'; \quad i = 1, 2, \dots, n \tag{2}$$

The abstract compressed data vectors y_i is calculated in the auto-encoder encoding process presented below.

$$T: y_i = g(x_i w + b); \quad \text{for } i = 1, 2, \dots, n \tag{3}$$

Here, the input layer presents the weight matrix, input vector, and bias vector as w, x_i, and b. At the same time, the auto-encoder has the process of deciding to use for calculating reconstructed data vectors x_i'.

$$S: x_i' = g(y_i w^T + b) \tag{4}$$

Here, b, x_i and w represent the bias vector, abstract compressed data vector, and transpose weight matrix are the $b, y_i,$ and w^T in the hidden layer. Eq. (3) and Eq. (4) represent the function g, a non-linear or linear activation function. The calculation of the sigmoid function is done, and it is used with the below representation.

$$S: x_i' = g(y_i w^T + b) \tag{5}$$

The values are used in the supervised learning for tuning the model with the values of b and w^T once the values of b and w^T are learned by using the auto-encoder in the unsupervised learning on unlabeled data with the help of a back-propagation algorithm having the classifier of softmax which depends on the vectors of training data.

3.7. Model explanation

The suggested model of adaptive RNN has the model of deep learning, which has two phases. They are (i) the initial decision phase and (ii) the final decision phase. The technique is used to construct the two phases called RNN (Recurrent neural network) because of the speed and performance in real-time classification. The network traffic is categorized as abnormal and normal, having the score value of probability in the initial phase. The value is considered the extra feature for training the final decision phase for the multi-class attack classification and normal. The technique of RNN is chosen in both stages, which merges the encoder with two hidden layers and the classifier of softmax layer on top. Two tasks are carried out in the stage of training to build the suggested model of RNN. The pre-training task is the first one where every layer of the encoder is individually pre-trained with the help of an unsupervised learning approach. The error is raised during reconstructing the minimized input features at every layer.

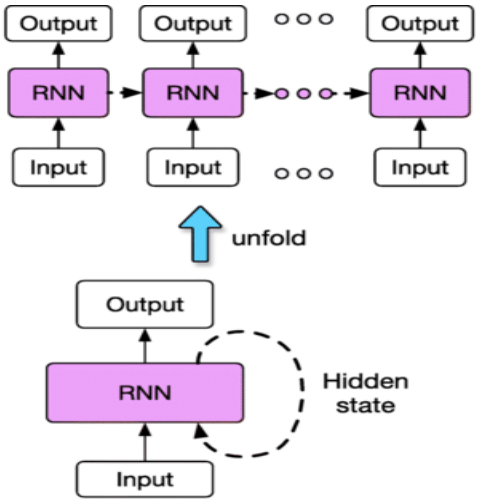

Figure 2: RNN model

Doi: https://doi.org/10.54216/FPA.120213
Received: May 25, 2023 Revised: June 16, 2023 Accepted: July 17, 2023

The first layer has the original unlabelled features, which are the inputs and the compressed features are the outputs which are the inputs to the next layer. The proposed system stacks them together and adds the classifier on top, a softmax layer for the upcoming fine-tuning task when the first and second layers are pre-trained individually. The supervised learning approach is used for tuning the pre-trained encoder model parameters depending on the back-propagation algorithm that is attained from the prior task. Then, the semi-supervised learning approach is used to train the suggested model. The fine-tuning task aims to lower the prediction error with the help of labelled features. The algorithm steps are summarized in two phases of the training phase of the model of RNN in Algorithm 1. The RNN model can classify the unknown network traffic instances after completion of the training stage. During the classification stage, the model of trained RNN has the steps which are presented in Fig 2. The over-fitting issue is reduced in the intention of the first phase, and the bias is mitigated to the normal traffic by locating more concentration on the abnormal traffic to categorize the various kinds of attacks. The suggested system gives the abstract feature space to differentiate between abnormal and normal flows of network traffic and discriminative.

Algorithm 1:

Input: l_x, l_t, x, t

Output: predict label and attack types; // pcl, pv

1. Begin //parameter initialization
2. initialize parameter (p);
3. //read training and testing feature sets
4. $y \rightarrow preTrain\ (x, IN1, H1);$
5. $v \rightarrow preTrain\ (y,\ IN2, H2);$
6. Tune the parameters
7. $Auto - encoder \rightarrow stacked\ layer\ (H1, H2);$
8. $RNN \rightarrow stacked\ layer\ (IN, encoding, OT);$
9. $adaptive - RNN \rightarrow fineTune\ (x, l_x, RNN);$
10. Perform classification
11. $[pcl, pv] \rightarrow classify\ (t, adaptive - RNN);$
12. return $adaptive - RNN, pcl, pv;$
13. end

4. Numerical results

The simulation is done in MATLAB 2020a. The storage capacity is 500 GB by the server. Here, 0.2 is the ratio of test data and 0.8 for training data. There are seven data sets and six models. Every model contains various parameters. The below steps are performed by the experiments to find the optimum model in a fast way. Primarily, the model is selected, which performs well from the novel analysis. The GRU model is preferred in the proposed model. Thus, the better performance data sets are compared from the experiments in the proposed system for training various models, and the results are compared in the second step. The hyperparameters are optimized for the model's dataset in the proposed system. The first methodology enumerated the parameters' discrete values option, and the cross-combination was performed to compare all the performance results from the experiment.

The standard statistical measures such as recall, precision, and accuracy are to evaluate the model of learning contains the better performance. The simple mathematical computations obtain the indicators for four atomic statistical indicators concerning the number of accurately found negative data points (TN), the number of accurately found positive instances (TP), the number of positive instances labelled as negative (FN), and the model

Doi: https://doi.org/10.54216/FPA.120213
Received: May 25, 2023 Revised: June 16, 2023 Accepted: July 17, 2023

predicts the number of negative predicted data points (FP). The F1 score is used to represent precision and recall. Also, a fake alarm affects the trust and experience of the user in the detection applications of cyber-security, and the alarms are leaked, causing the user's losses. Hence, the F1, accuracy, rate of false-negative and the rate of false-positive are used for measuring the models' effectiveness. Further, the metric called AP as Average Precision is used to evaluate the deep learning models' accuracy using the average precision value for the recall value across 0 to 1d, which is greater and better. The average of AP is the MAP as Mean Average Precision. The two is the number of classes.

The effectiveness accessed in the machine learning model is incomplete based on accuracy. It is customized for higher accuracy in the experiments in one dataset, which is not performed. The greater accuracy is obtained by the models not predicted by the new accurate data in real-time. Because over-fitting happens, these situations occur. The concept of over-fitting is analyzed if the trained model effectively predicts the unaware new Data in data mining. It is general for comparing the errors in the process of training having the errors in the validation process in the machine learning-based classification whether there are over-fitting having the epoch. The loss of validation and the loss of training is presented as having epochs in the model of a – RNN. Overfitting is avoided by one of the techniques called early-stopping. The epoch equalized to 6 is presented in the point of differentiation between over-fitting and under-fitting.

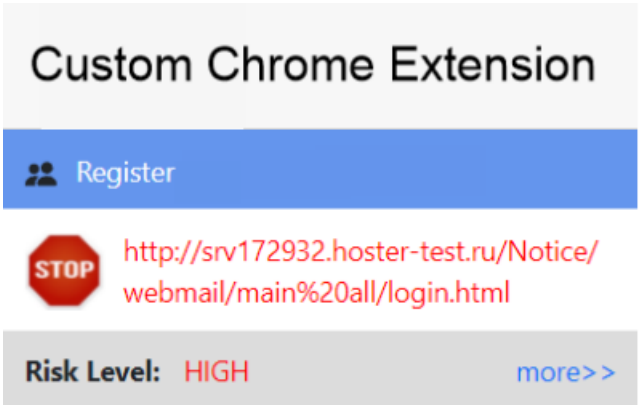

Figure 3: Chrome browser extension

4.1. Dataset analysis

The experiments use a similar data set for various models of machine learning from the analysis of the model having the better performance. Various researches presented that the random forest classifier outperforms the other conventional classification models to detect phishing networks. Hence, the classifiers selected in the proposed system as SVM, random forest, and logistic regression. Also, the architecture of the a – RNN model is unique in training the data sequence in the deep learning technique. The performance measure is compared with the six models in the proposed system. a – RNN obtains 99.18% accuracy, which is the greatest, and the low rate of false-positive of 0.0047% is obtained by the random forest presented in Table 1.

The F1 scores and the accuracy of other models are nearer in the experiments. Fig 3 presents this performance. It is not presented in the figure due to the accuracy of the mentioned model of a – RNN, which is lower than 0.9. The LSTM unit and the gate unit provide the effects on the data training sequence from the outcome data of three models of deep learning are presented again. The false-negative rate is greatest, and the summation of two error rates is higher even though the existing model obtains a low false-positive rate.

4.2. Parameter optimization

The dataset is used from the mentioned outcomes in the model of a – RNN to achieve better performance. The hyperparameters model is optimized by the third experiment having the best performance. 162 combinations of the option values are performed for all the parameters in total. The experiment is carried out after the deployment of the system to the cloud due to the experiment is run by a computer GPU do not support parallel computing, and a long time is taken to train the model having the dataset. The comparison is visualized by accessing the tool

Doi: https://doi.org/10.54216/FPA.120213
Received: May 25, 2023 Revised: June 16, 2023 Accepted: July 17, 2023

MATLAB 2020a for the execution results, and the performance measures obtain a better combination of parameters.

4.3. Comparison

The model a – RNN is compared with the existing results in the proposed system, which trains deep learning models for detecting phishing websites. Various dimensions show the comparison like data collection, indicators having performance, limitations, and indicators shown. Since there are no short links of the training model in the dataset and all the present prediction services are not detected accurately if the short links are at the phishing risk as the limitations of the suggested solution are implemented. Further, the first 200 URL characters are intercepted with above 200 characters, which is lost as the information and the detection results are affected. Also, the reporting process of the automatic review is judged presently depending on the rules like information about the client, the remote IP address, and the number of URL submitted. The phishing attackers used this strategy maliciously. More data is required to support the review results automatically in future. Consider an example, the current URL obtains the HTML and if there is an input box in HTML and identify the similarity between the website which is whist listed and the image of the logo.

5. Evaluation metrics

The below measurements related to performance are used for determining the suggested system and other models such as recall, accuracy, F-measure and precision. The number of legal URLs is accurately labelled as legal, plus the count of phishing URLs that are accurately labelled as phishing is termed as the accuracy of the total number of samples of the test set. The calculation of accuracy is provided in Eq. (14):

$$\text{Accuracy, A} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}} \tag{14}$$

The number of phishing URLs is accurately labelled as phishing, and then the total count of the labelled URLs as phishing is termed precision. Eq. (15) shows the calculation for precision.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{15}$$

The number of phishing URLs is accurately labelled as phishing rather than the total count of exact URLs of phishing is termed as recall, which is referred to as sensitivity and TPR. Eq. (16) shows the calculation for recall.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{16}$$

The rates of precision and recall have the weighted harmonic mean termed the F-measure. Methodology having the greater F-measure is more efficient. Eq. (17) shows the calculation for F-measure.

$$\text{F} = \frac{2 * \text{precision} * \text{recall}}{\text{precision} + \text{recall}} \tag{17}$$

The phishing URLs are accurately labelled as phishing URLs as indicated by true positives (TPs), and the legal URLs are inaccurately labelled as phishing URLs as indicated by false positives (FPs); the legal URLs are correctly labelled as legal URLs as indicated by true negatives (TNs), and phishing URLs are inaccurately labelled as legal URLs is indicated by false negatives (FNs).

Table 1: Result analysis

Dataset	Accuracy	Precision	Recall	F-measure
MUD	96%	96%	96.8%	96.4%

Doi: https://doi.org/10.54216/FPA.120213
Received: May 25, 2023 Revised: June 16, 2023 Accepted: July 17, 2023

Synthetic phishing URL + MUPD (5000)	96.5%	95.8%	96.8%	96.2%
Synthetic phishing URL + MUPD (10,000)	96.54%	96.8%	96.3%	96.5%
Synthetic phishing URL + MUPD (50,000)	97.5%	98%	97.2%	97.6%
MUPD	98.5%	98.8%	98.3%	98.7%
(a – RNN)	99.2%	99%	99%	99%

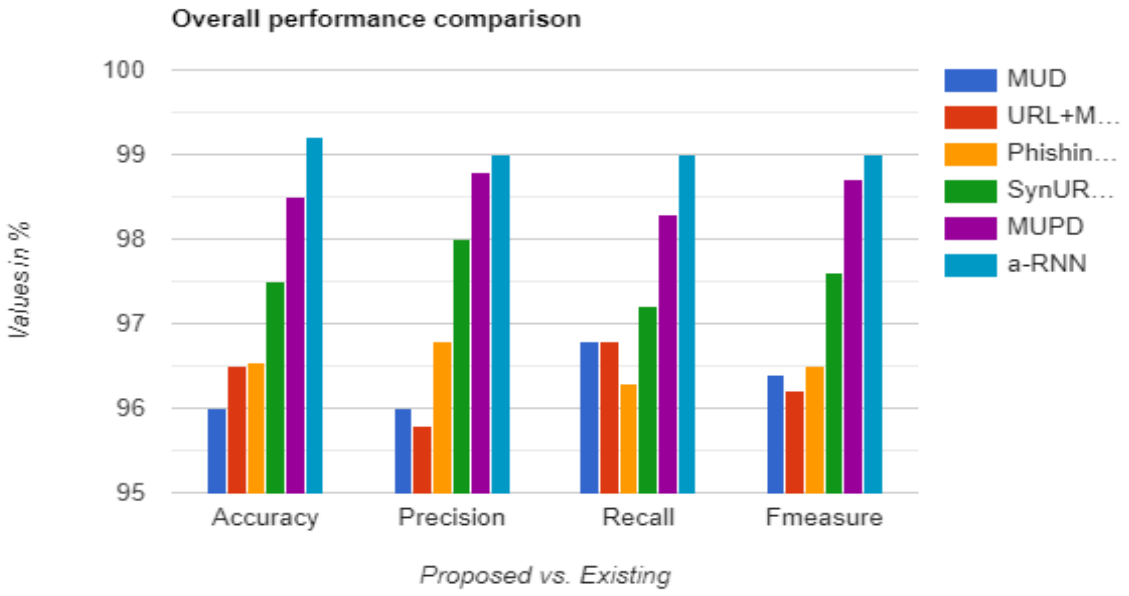

Figure 4: Result analysis

Table 2: Performance of a – RNN over different scales in the provided dataset

Dataset split	Accuracy	Precision	Recall	F-measure
10%	98	98.1	98.1	98.9
20%	98.90	98.2	98.5	98.7
30%	99.1	99	98.6	98.6
40%	99.2	99.1	98.7	98.4
50%	99.5	98.9	98.8	98.2
60%	99.6	98.5	98.1	98.5
70%	99	99	98.2	98.4

Doi: https://doi.org/10.54216/FPA.120213
Received: May 25, 2023 Revised: June 16, 2023 Accepted: July 17, 2023

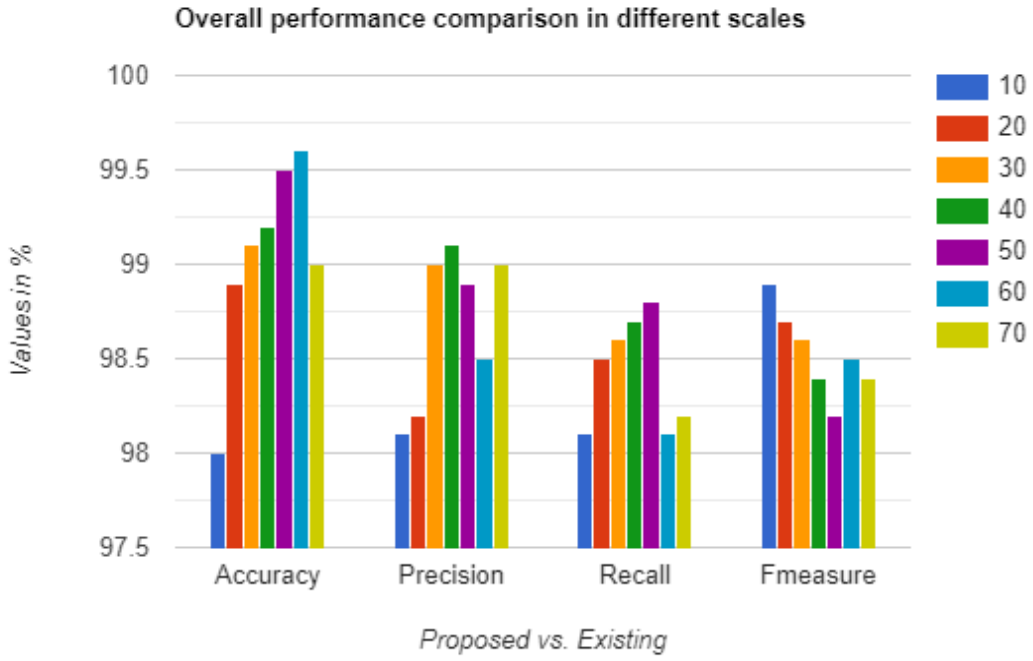

Figure 5: Comparative analysis

5. Conclusion

In recent years, multiple solutions based on machine learning have been suggested for dealing with phishing attacks, yet the outcomes are not checked in the live environments of browsing, and the analysis and the study regarding products are lacking in detecting phishing. A framework is suggested to detect phishing in the browsing circumstance in real-time. The theoretical framework has the features that are presented below.

1) A closed-loop data is used in the proposed system to drive machine learning models with the best performance. The basic need to train the model is the dataset, and the data quality should be higher to enhance the model's performance. The data feedback is the high data quality with accuracy, sensitivity, and enhancement from users.

2) In a real-time environment, the system is running with no delays. When opening the web phase, the results of the prediction is displayed.

3) The tracking of experimental data is done. The model's training process is an automatic task, and every result from execution is saved in the database in real-time.

4) A browser extension is developed like a client's product that each common user can utilize.

5) The services of prediction are implemented that are extendable, and every service of detection is merged. Consider an instance where the service of blocklist filtering is introduced and the service of computer vision.

6) The feature extraction process is not dependent on the third-party services in the deep learning model.

References

[1] De', N. Pandey, and A. Pal, ''Impact of digital surge during COVID19 pandemic: A viewpoint on research and practice,'' Int. J. Inf. Manage., vol. 55, Dec. 2020, Art. no. 102171.
[2] J. A. Chaudhry, S. A. Chaudhry, and R. G. Rittenhouse, ''Phishing attacks and defences,'' Int. J. Secur. Appl., vol. 10, no. 1, pp. 247–256, 2016.

Doi: https://doi.org/10.54216/FPA.120213
Received: May 25, 2023 Revised: June 16, 2023 Accepted: July 17, 2023

[3] A. Alzahrani, ‘‘Coronavirus social engineering attacks: Issues and recommendations,’’ Int. J. Adv. Comput. Sci. Appl., vol. 11, no. 5, pp. 154–161, 2020.
[4] S. Bell and P. Komisarczuk, ‘‘An analysis of phishing blacklists: Google safe browsing, OpenPhish, and PhishTank,’’ in Proc. Australas. Comput. Sci. Week Multiconf., Feb. 2020, pp. 1–11
[5] Halgas, I. Agrafiotis, and J. R. C. Nurse, ‘‘Catching the phish: Detecting phishing attacks using recurrent neural networks (RNNs),’’ in Information Security Applications (Lecture Notes in Computer Science), vol. 11897. Cham, Switzerland: Springer, 2020, pp. 219–233
[6] Dou, I. Khalil, A. Khreishah, A. Al-Fuqaha, and M. Guizani, ‘‘Systematization of knowledge (SoK): A systematic review of software-based web phishing detection," IEEE Commun. Surveys Tuts., vol. 19, no. 4, pp. 2797–2819, 4th Quart., 2017.
[7] Cao, W. Han, and Y. Le, "Anti-phishing based on an automated individual whitelist," in Proc. 4th ACM Workshop Digit. Identity Manage. (DIM), 2008, pp. 51–59.
[8] Jain and B. B. Gupta, "A novel approach to protect against phishing attacks at client side using an auto-updated whitelist," EURASIP J. Inf. Secur., vol. 2016, no. 1, pp. 1–11, Dec. 2016.
[9] Jain and B. B. Gupta, ‘‘PHISH-SAFE: URL features-based phishing detection system using machine learning,’’ in Advances in Intelligent Systems and Computing, vol. 729. Singapore: Springer, 2018, pp. 467–474.
[10] Rao and A. R. Pais, ‘‘Detection of phishing websites using an efficient feature-based machine learning framework,’’ Neural Comput. Appl., vol. 31, no. 8, pp. 3851–3873, Aug. 2019.
[11] Li, Z. Yang, X. Chen, H. Yuan, and W. Liu, ‘‘A stacking model using URL and HTML features for phishing webpage detection,’’ Future Gener. Comput. Syst., vol. 94, pp. 27–39, May 2019.
[12] Xiang, J. Hong, C. P. Rose, and L. Cranor, ‘‘CANTINA+: A feature-rich machine learning framework for detecting phishing websites," ACM Trans. Inf. Syst. Secur., vol. 14, no. 2, pp. 1–28, Sep. 2011.
[13] Yang, J. Zhang, X. Wang, Z. Li, Z. Li, and Y. He, ‘‘An improved ELMbased and data pre-processing integrated approach for phishing detection considering comprehensive features,’’ Expert Syst. Appl., vol. 165, Mar. 2021, Art. no. 113863
[14] Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, ‘‘Distributed representations of words and phrases and their compositionality,’’ in Proc. Adv. Neural Inf. Process. Syst., vol. 26, no. 4, Dec. 2013, pp. 3111–3119.
[15] Al-Alyan and S. Al-Ahmadi, ‘‘Robust URL phishing detection based on deep learning," KSII Trans. Internet Inf. Syst., vol. 14, no. 7, pp. 2752–2768, 2020.
[16] Chiew, C. L. Tan, K. Wong, K. S. C. Yong, and W. K. Tiong, ‘‘A new hybrid ensemble feature selection framework for machine learning-based phishing detection system,’’ Inf. Sci., vol. 484, pp. 153–166, May 2019.
[17] Chen and C. Guestrin, ‘‘XGBoost: A scalable tree boosting system,’’ in Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min., vols. 13–17, 2016, pp. 785–794.
[18] Do, A. Selamat, O. Krejcar, T. Yokoi, and H. Fujita, ‘‘Phishing webpage classification via deep learning-based algorithms: An empirical study,’’ Appl. Sci., vol. 11, no. 19, p. 9210, Oct. 2021
[19] Chen, W. Zhang, and Y. Su, ‘‘Phishing detection research based on LSTM recurrent neural network,’’ Data Sci., vol. 6, pp. 638–645, Sep. 2018
[20] Chen, W. Zhang, and Y. Su, ‘‘Phishing detection research based on LSTM recurrent neural network,’’ Data Sci., vol. 6, pp. 638–645, Sep. 2018
[21] Sherubha, “Graph Based Event Measurement for Analyzing Distributed Anomalies in Sensor Networks”, Sådhanå(Springer), 45:212, https://doi.org/10.1007/s12046-020-01451-w
[22] Sherubha, “An Efficient Network Threat Detection and Classification Method using ANP-MVPS Algorithm in Wireless Sensor Networks”, International Journal of Innovative Technology and Exploring Engineering (IJITEE), ISSN: 2278-3075, Volume-8 Issue-11, September 2019
[23] Sherubha, “An Efficient Intrusion Detection and Authentication Mechanism for Detecting Clone Attack in Wireless Sensor Networks”, Journal of Advanced Research in Dynamical and Control Systems (JARDCS), Volume 11, issue 5, Pg No. 55-68
[24] Paul and S. Das, ‘‘Simultaneous feature selection and weighting— An evolutionary multi-objective optimization approach,’’ Pattern Recognit. Lett., vol. 65, pp. 51–59, Nov. 2015.
[25] Xue, M. Zhang, and W. N. Browne, "Particle swarm optimization for feature selection in classification: Novel initialization and updating mechanisms," Appl. Soft Comput., vol. 18, pp. 261–276, May 2014.
[26] Huang, S. Hao, L. Invernizzi, Y. Fang, C. Kruegel, and G. Vigna, ‘‘Gossip: Automatically identifying malicious domains from mailing list discussions,’’ in Proc. ACM Asia Conf. Comput. Commun. Secur. (ASIA CCS), Abu Dhabi, United Arab Emirates, Apr. 2017, pp. 494–505

Doi: https://doi.org/10.54216/FPA.120213
Received: May 25, 2023 Revised: June 16, 2023 Accepted: July 17, 2023

[27] Saxe, R. Harang, C. Wild, and H. Sanders, ‘‘A deep learning approach to fast, format-agnostic detection of malicious Web content,’’ in Proc. IEEE Symp. Secur. Privacy Workshops (SPW), San Francisco, CA, USA, Aug. 2018, pp. 8–14.
[28] Wu, X. Du, and J. Wu, ‘‘Effective defence schemes for phishing attacks on mobile computing platforms,’’ IEEE Trans. Veh. Technol., vol. 65, no. 8, pp. 6678–6691, Aug. 2016.
[29] Gowtham and I. Krishnamurthi, ‘‘A comprehensive and efficacious architecture for detecting phishing webpages,’’ Comput. Secur., vol. 40, pp. 23–37, 2014.
[30] Sheng, B. Wardman, G. Warner, L. F. Cranor, J. Hong, and C. Zhang, ‘‘An empirical analysis of phishing blacklists,’’ in Proc. 6th Conf. Email Anti-Spam (CEAS), Mountain View, CA, USA, Jul. 2009, pp. 1–20.
[31] Han, N. Kheir, and D. Balzarotti, ‘‘PhishEye: Live monitoring of sandboxed phishing kits,’’ in Proc. 23rd ACM Conf. Comput. Commun. Secur. (CCS), Vienna, Austria, Oct. 2016, pp. 1402–1413
[32] Rao and S. T. Ali, ‘‘PhishShield: A desktop application to detect phishing Webpages through heuristic approach,’’ Procedia Comput. Sci., vol. 54, pp. 147–156, Aug. 2015.
[33] Jain and B. B. Gupta, "A novel approach to protect against phishing attacks at client side using an auto-updated whitelist," EURASIP J. Inf. Secur., vol. 2016, no. 1, Dec. 2016, Art. no. 9.
[34] Kausar, B. Al-Otaibi, A. Al-Qadi, and N. Al-Dossari, "Hybrid client-side phishing websites detection approach," Int. J. Adv. Comput. Sci. Appl., vol. 5, no. 7, pp. 132–140, 2014.
[35] Varshney, M. Misra, and P. K. Atrey, ‘‘A phish detector using lightweight search features,’’ Comput. Secur., vol. 62, pp. 213–228, Sep. 2016.
[36] S. Hemamalini ,V. D. Ambeth Kumar ,R. Venkatesan,S. Malathi. (2023). Relevance Mapping based CNN model with OSR-FCA Technique for Multi-label DR Classification. Journal of Fusion: Practice and Applications, 11 (2), 90-110.
[37] C. S. Manigandaa,V. D. Ambeth Kumar,G. Ragunath,R. Venkatesan,N. Senthil Kumar. (2023). De-Noising and Segmentation of Medical Images using Neutrophilic Sets. Journal of Fusion: Practice and Applications, 11 (2), 111-123
[38] Sathya Preiya, V., and V. D. Ambeth Kumar. 2023. "Deep Learning-Based Classification and Feature Extraction for Predicting Pathogenesis of Foot Ulcers in Patients with Diabetes" Diagnostics 13, no. 12: 1983. https://doi.org/10.3390/diagnostics13121983
[39] Balakrishnan, Chitra, and V. D. Ambeth Kumar. 2023. "IoT-Enabled Classification of Echocardiogram Images for Cardiovascular Disease Risk Prediction with Pre-Trained Recurrent Convolutional Neural Networks" Diagnostics 13, no. 4: 775. https://doi.org/10.3390/diagnostics13040775.
[40] V. D. Ambeth Kumar,S. Malathi,Abhishek Kumar,Prakash M and Kalyana C. Veluvolu, “Active Volume Control in Smart Phones Based on User Activity and Ambient Noise” ,Sensors 2020, 20(15), 4117; https://doi.org/10.3390/s20154117
[41] V. Sathya Preiya,V. D. Ambeth Kumar,R. Vijay,Vijay K.,N. Kirubakaran. "Blockchain-Based E-Voting System with Face Recognition." Fusion: Practice and Applications, Vol. 12, No. 1, 2023 ,PP. 53-63

Doi: https://doi.org/10.54216/FPA.120213
Received: May 25, 2023 Revised: June 16, 2023 Accepted: July 17, 2023