Convolution Neural Networks for Phishing Detection

Arun D. Kulkarni
Computer Science Department
The University of Texas at Tyler
Tyler, TX, 75799, USA

***Abstract*—Phishing is one of the significant threats in cyber security. Phishing is a form of social engineering that uses e-mails with malicious websites to solicitate personal information. Phishing e-mails are growing in alarming number. In this paper we propose a novel machine learning approach to classify phishing websites using Convolution Neural Networks (CNNs) that use URL based features. CNNs consist of a stack of convolution, pooling layers, and a fully connected layer. CNNs accept images as input and perform feature extraction and classification. Many CNN models are available today. To avoid vanishing gradient problem, recent CNNs use entropy loss function with Rectified Linear Units (ReLU). To use a CNN, we convert feature vectors into images. To evaluate our approach, we use a dataset consists of 1,353 real world URLs that were classified into three categories-legitimate, suspicious, and phishing. The images representing feature vectors are classified using a simple CNN. We developed MATLAB scripts to convert vectors into images and to implement a simple CNN model. The classification accuracy obtained was 86.5 percent.**

Keywords—Classification; convolution neural networks; machine learning; phishing URLs

I. Introduction

Convolution Neural Networks (CNNs) present a tool that enables the computer to learn from image samples, extract internal representations, and classify images. Study into CNN has increased in recent years as the computing power is being available. CNNs have several advantages such as they do not require any feature extraction technique. CNNs extract features through convolution and pooling. Through unique layer designs CNNs can extract higher order statistics and non-linear correlations. Today, many CNN models are available in practice that can be executed efficiently with recent advantages in hardware like Graphical Processing Units (GPUs). CNNs need image data as an input. Conventional Machine Learning (ML) techniques require samples in the form of a feature vector for classification. The purpose of feature extraction to reduce data by measuring certain features or properties that distinguish input samples. Samples belonging to the same categories form clusters in the feature space. The classification problem essentially reduces to partitioning the feature space. When classes overlap in the feature space classifiers such Naïve Bayes classifier makes decisions based on posterior probabilities. Commonly used parametric and non-parametric classification techniques in ML include decision trees, neural networks, minimum distance classifier, Support Vector Machine (SVM), Naïve Bayes classifier, etc. For these conventional ML techniques, input is presented in the table form and each sample is represented by a feature vector, whereas for CNN models, data is presented in the form of an image. To take advantage of CNNs one-dimensional feature vectors can be converted into two-dimensional images. A lot of data such as genomics, transcriptomic, methylation, mutation, text, spoken words, financial and banking are in non-image form and ML techniques used in these fields. Sharma et al. [1] have suggested a methodology to transform non-image data to image data. They have suggested a method to map a vector consisting of gene expression values to a feature matrix. In their method the location of a feature in the feature matrix depends upon similarity between feature values. In this paper, we suggest a new approach to map a feature vector to a feature matrix or the output image. In our approach, we divide the output image into regions and the gray value of each region is determined by a value in the feature vector. Each region in the output image represents a value in the feature vector. The number of regions in the output image is the same as the dimension of the feature vector. We have considered the problem of classification of Uniform Resource Locators (URLs) using a Deep Convolution Neural Network (DCNN). URL represents documents and other resources on the World Wide Web (WWW). Malicious web sites present a serious threat to cybersecurity. Malicious websites host unsolicited contents such as spam, phishing, viruses, etc. Many Machine Learning (ML) algorithms have been used to classify malicious URLs into classes such as legitimate, suspicious, or phishing. Common types of attacks using a malicious URL include Driven-by download, phishing, and spam. URLs consist of two components the protocol identifier and resource name. These two components are separated by a colon and two forward slashes. The common method to detect malicious URLs is the black-list method, which is database compiled over the period. ML approaches for URL classification use a set of URLs as training data and develop a model. To develop a model, one needs to extract features from URLs. In the present study, we have used a dataset consists of features 1,353 real world URLs that were classified into three categories-legitimate, suspicious, and phishing. The dataset contains ten attributes. The three classes: phishing, suspicious, and legitimate are denoted by -1, 0, and 1 respectively. We have converted the feature vectors into images that were stored in three folders. The folder names are the same as the class names. We developed a MATLAB script to map feature vectors into images, to implement a simple convolution neural network to classify URLs. The outline of the paper is as follows. The related work is provided in Section II. The proposed approach is described in Section III and implementation and results are provided in Section IV. Section V presents the conclusions and the future work.

II. Related Work

This paper deals with phishing detection using convolution neural networks (CNN). CNN models are a part of artificial intelligence (AI). AI includes any technique that enables computers to mimic human behavior and reproduce or excel over human decision making to solve complex tasks independently or with minimum human intervention [2]. AI research deals with reasoning, knowledge representation, natural language processing. AI includes machine learning (ML) algorithms, Artificial Neural Networks (ANN), and deep learning networks. Machine learning evolved from pattern recognition and computational learning theory [3]. ANN models are biologically inspired. They learn from training samples and have used in pattern recognition since 1950s. Many ANN models with learning algorithms such as multilayer perceptron, backpropagation, Boltzmann machine, Hopfield net, neo cognition model etc. are available in practice [4,5,6,7]. Deep learning is a form of machine learning that enables computers to understand the world in terms of hierarchy of concepts. Convolution Neural Networks (CNNs) are special type of networks for processing data that have a known grid like structure [8]. DNNs discover in large datasets using the backpropagation algorithm [9]. CNNs are feedforward networks in that information flow takes place in one direction only, from their input to output [10]. CNN architectures in general consist of convolution and pooling layers that grouped in modules followed by fully connected layers. CNNs evolve into deep convolution neural networks (DCNN). DCNNs proven to be one of the best learning algorithms for understanding image contents and shown exemplary performance in image segmentation, detection, and retrieval tasks [11]. Recent developments in DCNN were possible because of availability of large data sets and graphical processing units (GPUs). With the ability of new programming framework, availability of data, and accessibility to GPUs many analytical models are developed [12]. DCNNs use gradient decent backpropagation algorithm. The use of Sigmoid activation functions leads to saturation resulting into slow convergence of gradient decent algorithm. The problem becomes sever as we go away from the output layers to hidden layers. The compound effect of saturation at multiple layers is known as vanishing gradient [13]. To avoid the vanishing gradient problem, DCNNs often use entropy loss functions with Rectified Linear Units (ReLU) in the output layer. Another issue with DCNNs is overfitting. Various regularization techniques such as dropout or bagging are used to overcome this problem [14].

Phishing URLs is one of significant threats in the world today. Commonly used technique for phishing URLs detection is blacklisting. Blacklists include sender blacklists and link blacklists. The effectiveness of using blacklists depends on update of databases that maintain blacklists. Phishing emails are sent from an Internet disguised as an email from a legitimate, trustworthy source. Many researchers have worked on phishing email detection. Gilehan and Taylor [15] used syntactic features for phishing detection. They presented the comparison of sentence syntactic similarity and the difference in subjects and objects of target verbs between phishing emails and legitimate emails. Fang et al. [16] suggested a framework to detect phishing emails based on improved recurrent convolution neural networks (RCNN) with multilevel vectors and attention mechanism. In their approach to extract features they divide each email into multiple levels, the character and word level of the email header as well as the character and word level of the email body. Rashid et al. [17] propose an efficient machine learning based phishing detection technique. They first extract lexical, host and word vector features and using the principal component analysis to reduce the number of features and use the SVM model for classification. They use five principal components and obtain the efficiency of 95.66 percent. Machine learning techniques for phishing extract features that distinguish legitimates from phishing websites. Features are extracted from various sources such as URLs, page content, search engine, digital certificate, web traffic etc. Software based approaches are classified into machine learning based, blacklist based, and visual similarity based [18].

Zhang et al. [19] proposed a page content-based technique. Huang et al. [20] proposed an approach that is based on URL features. They have used 23 features from URL and used the SVM. The two classifier values are fed into the fusion model. Abdelhamid et al, [21] built a system for detecting phishing URLs based on associative classification. Hadi et al. [22] proposed an approach for detecting malicious URLs using only visible features from social networks. Kulkarni and Brown [23] have classified phishing URLs using machine learning techniques such as SVM, decision tree, Naïve based classifier, and ANN. Sahoo et al. [24] provide a comprehensive survey and structural understanding of malicious URL detection techniques using machine learning. Yang et al. [25] have proposed a spam filtering method based on multi-model fusion. During pre-processing they separate text and image data from an email. The text dataset to train Long-Short Term Memory (LSTM) and image datasets are used to train a CNN model. CNN architecture allows dealing with images effectively. CNN architecture employs a collection of neighborhood pixels as opposed to individual use of features by ML models [1]. Chiramdasu, et al. [26] explore the various ways of detecting malicious links from the host-based and lexical features of the URL to protect users from being subjected to identity theft attacks. We have used a CNN model for classification of URLs into three classes legitimate, suspicious and phishing. We used features that are extracted from URLs. To use CNN, we first converted feature vectors to images that were classified by the CNN model.

III. Methodology

In this paper, we propose a framework for classification of phishing URLs. In our approach we use the features that are extracted from URLs. Often phishing emails contain URLs of malicious websites. We use a simple DCNN to classify URLs from their feature vectors. The framework for the proposed approach is shown in Fig. 1. The second step in our approach converts the feature vectors into 2-D image matrices.

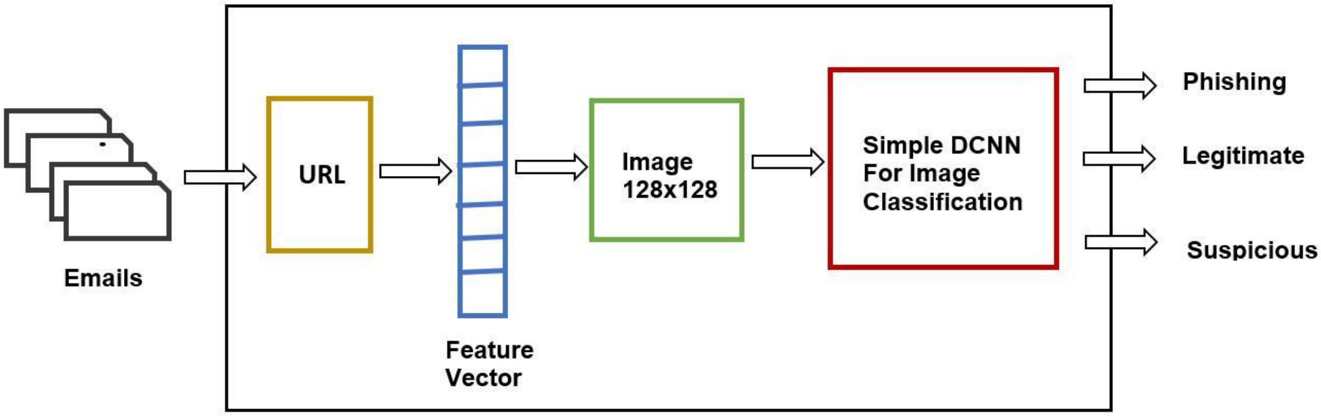


Fig. 1. Framework for phishing URL detection.

In our method we first normalize the values between 0 and 255 and create gray value regions based on the features in the feature vector, 0 corresponds to black and 255 corresponds to white. In between values are mapped to corresponding gray values. Fig. 2 shows the feature vector consists for four features and the output image. The gray values represent numeric values in the feature vector. There are four regions in the image, and each represents a feature value and the image represents the feature vectors. All feature vectors in the dataset are converted to the corresponding images. We created a datastore containing three folders one for each class. The labels of the folders are the same as the class labels. The images were of the size of 128 rows and 128 columns. The images were split into two datasets-training and testing datasets by randomly chosen images. The DCNN was trained with the training set images and was tested with images in the test dataset. Conventional neural networks with the backpropagation learning algorithms have been used for classifying feature vectors. Conventional neural networks during the learning phase use the mean squared error at the output layer and is propagated backwards to hidden layer to update the weights. That causes vanishing gradient problem. We use a simplified model of a DCNN as shown in Fig. 3.

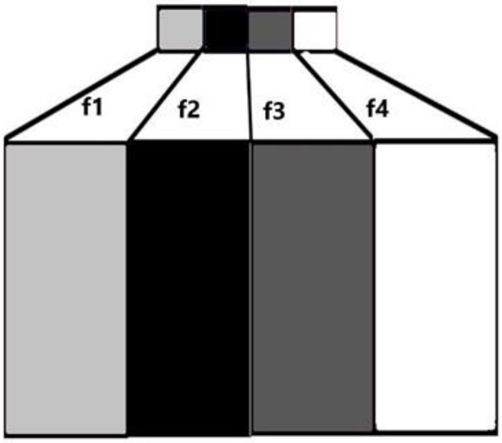


Fig. 2. Mapping a feature vector to an image.

The model consists of the input layer, convolution layer, batch normalization layer, ReLU layer, max pooling layer, fully connected layer, SoftMax layer, and classification layer. We can specify the input image size at the input layer. There are three convolution layers. The batch normalization layers normalize the activations and gradients propagating through the network, which makes the training an easier optimization problem. The batch normalization layers are followed by an ReLU layers. The max pooling layer is used to downsize the network and extract features. The fully connected, SoftMax and classifier layers map the feature vector to class labels. The output of the SoftMax layer consists of positive numbers that sum one that are used as class probabilities. To classify URLs, we have used three classes: phishing, legitimate, and suspicious.

IV. Implementation and Results

We used a dataset consists of 1,353 real word URLs that are classified into three categories a) phishing, b) legitimate, and c) suspicious. The dataset used in this paper is downloaded from University of California at Irvine (UCI) Machine Learning Repository [27]. The data set consists of ten features that are extracted from each URL. The features include URL of the anchor, Request URL, Server Form Handler (SFH), URL Length, Having "@" character, Prefix/Suffix, IP address, Sub Domain, Web Traffic, and Domain Age. These features are represented by numeric values such as -1, 0, and 1. We transformed each feature vector into a gray image by mapping numeric values in the feature vector to gray value regions. The images are classified using a DCNN model shown in Fig. 3. The image size for the input layer was set to 128x128. We used a 3x3 filter size and 8, 16, and 32 filters in the first, second, and third convolution layers, respectively. We used a 2x2 region size for the max pool layer. The number of units in the output layer was set to three as there three classes in the dataset.

We developed a MATLAB script to convert feature vectors into images. Three folders were created for three classes. The folder names are the same as the class names-phishing, legitimate, and suspicious. The output images were stored in the respective class folders. The images were classified by the simple DCNN. The total number of tuples in the data set is 1353 that represents 702 phishing, 548, legitimate, and 103 suspicious URLs. The samples were split into two datasets- the training and testing datasets. Randomly chosen seventy percent samples were used for training and thirty percent were used for validation. Fig. 4 shows randomly chose sixteen images from the training datasets. The classified images with class labels are shown in Fig. 5. The DCNN was trained using the training set data. Fig. 6 shows the accuracy and error curves with respect to epochs. The validation accuracy obtained was 85.47 percent in eight epochs.

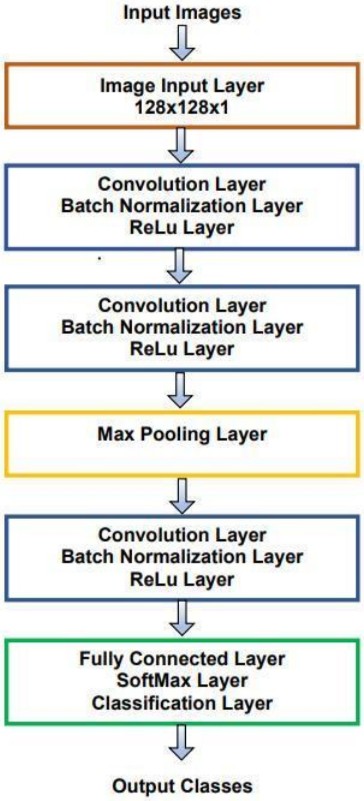


Fig. 3. Architecture for the simplified DCNN.

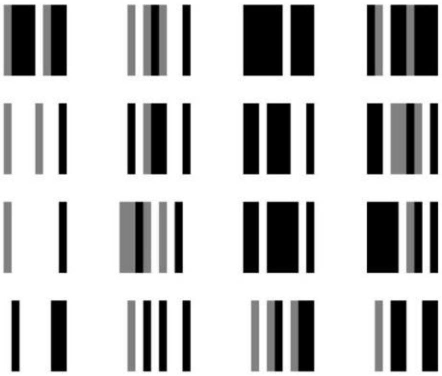


Fig. 4. Sample images from training set data.

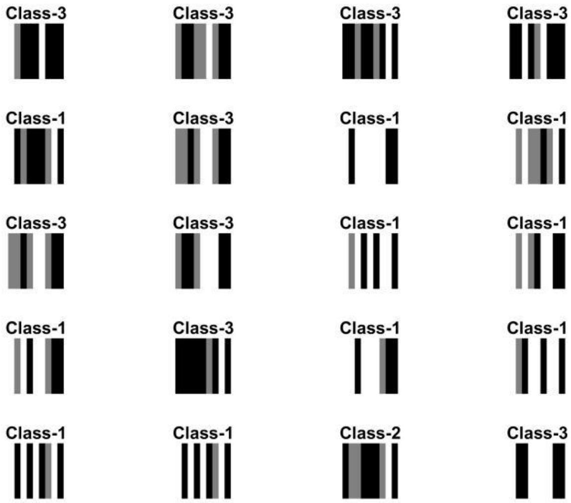
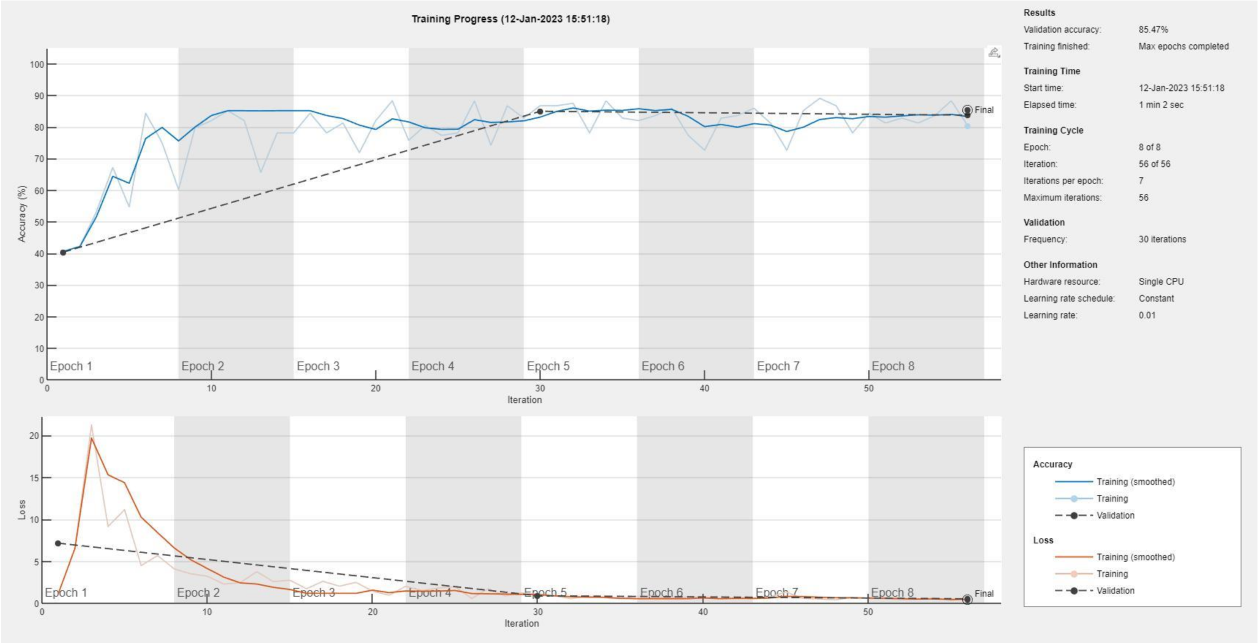


Fig. 5. Classified output images.

V. Conclusions

In this paper, we suggested a new approach to map non-image data to two-dimensional images so that data in the feature vectors form can be classified using CNN models. We mapped values in the feature vector to regions with different gray shades that are determined by feature values. We developed MATLAB script to convert feature vectors to images and classify them using a simple CNN model. The model was trained to classify real life URLs into three classes legitimate, suspicious, and phishing. We used randomly chosen seventy percent samples for training and thirty percent for testing. We obtained an accuracy of 85.56 percent. There are many ways to improve classification accuracy. In our method we used rectangular regions to map values in the feature vectors to corresponding gray regions in the output image. It is possible to use more complex shapes to define the regions. It is also possible to define shapes in the output image as a function of feature values. Furthermore, we can use DCNN models with a greater number of layers such as Alex Net, Res-Net, etc. to classify images obtained from the feature dataset. Our future work includes classifying data with DCNNs and testing the models with big datasets. In the present data set attributes consists of only three discrete values -1, 0, and 1. We plan to test the algorithm for features with multiple discrete values and explore complex shapes for mapping feature vectors to images and evaluate the suggested algorithm by comparing it with other machine learning algorithms.



Results
Validation accuracy: 85.47%
Training finished: Max epochs completed

Training Time
Start time: 12-Jan-2023 15:51:18
Elapsed time: 1 min 2 sec

Training Cycle
Epoch: 8 of 8
Iteration: 56 of 56
Iterations per epoch: 7
Maximum iterations: 56

Validation
Frequency: 30 iterations

Other Information
Hardware resource: Single CPU
Learning rate schedule: Constant
Learning rate: 0.01

Fig. 6. Training progress plot.

REFERENCES

[1] A. Sharma, E. Vans, D. Shigemizu, K. A. Boroevich, and T. Tsunoda, “Deep Insight: A methodology to transform a non-image data to an image for convolution neural network architecture”. Sci Rep 9, 11399, 2019.

[2] S. J. Russell and P. Norvig, Artificial Intelligence: A Modern Approach, Global Edition. Pearson, Harlow, UK, 2022.

[3] W. L. Hosch, Machine Learning [Online] Retrieved 2017-06-01 from http://www.britannica.com/EBchecked/topic/1116194/machinelearning.

[4] J. J. Hopfield, “Neurons with graded response have collective computational properties like those of two state neurons,” Proceedings of the National Academy of Sciences, 1984, vol. 81, pp. 3088-3092.

[5] D. E. Rumelhart, J. L. McClelland, and the PDP Group, Parallel Distributed Processing, vol. I, MIT Press, Cambridge, MA, 1986.

[6] R. P. Lippmann, “An introduction to computing with neural nets,” IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 32, pp. 4-22, 1987.

[7] K. Fukushima, “Neural networks for visual pattern recognition,” Computer, pp. 65-74, March 1988.

[8] I. Goodfellow, Y. Bengin, and A. Courville, Deep Learning. The MIT Press, Cambridge, MA, USA, 2016.

[9] Y. LeCun, Y. Bengin, and G. Hinton, “Deep learning,” Nature, vol. 521, pp. 436-444, 2015.

[10] W. Rawat, and Z. Wang, “Deep convolution neural networks for image classification: A comprehensive review”, Neural Computation, vol. 29, pp. 2352-2449, 2017.

[11] D. Cireşan, U. Meier, J. Schmidhuber, “Multi-column Deep Neural Networks for Image Classification,” Computer Vision and Pattern Recognition, 2012, pp. 3642-3649.

[12] E. Brynjolfsson, and A. McAfee, “The business of artificial intelligence”, Harvard Business Review, pp. 1–20, 2017.

[13] M. Tan and Q. V. Le, “Efficient Net: Rethinking Model Scaling for Convolutional Neural Networks,” Proceedings of the 36th International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019.

[14] A. D. Kulkarni, “Deep Convolution Neural Networks for Image Classification”, International Journal of Advanced Computer Science and Applications, Vol. 13, No. 6, pp 18-23, 2022.

[15] P. Gilchan, and J. M. Taylor. "Using syntactic features for phishing detection." arXiv preprint arXiv:1506.00037, 2015.

[16] Y. Fang, C. Zhang, C. Huang, L. Liu, and Y. Yang., “Phishing email detection using improved RCNN model with multilevel vectors and attention mechanism,” IEEE Access, vol. 7, pp. 56329-56340, 2019.

[17] J. Rashid, T. Mahmood, M. W. Nisar, and T. Nazir., “Phishing Detection Using Machine Learning Technique” 2020 First International Conference of Smart Systems and Emerging Technologies (SMARTTECH). doi: 10.1109/SMART-TECH49988.2020.00026

[18] A. K. Jain and B. B. Gupta., “Comparative analysis of features based machine learning approaches for phishing detection”, IEEE International Conference on Computing for Global Development, pp 2125-2129, 2016.

[19] Y. Zhang, J. Hong, and L. Cranor, “CANTINA: a content-based approach to detecting phishing web sites” in Proceedings of the 16th International Conference World Wide Web, 2007.

[20] H. Huang, L. Qian, Y. Wang, “A SVM-based technique to detect phishing URLs”, Inf. Technol. J. vol. 11, no. 7, pp. 921-925, 2012.

[21] N. Abdelhamid, A. Ayesh, F. Thabtah, “Phishing Detection based Associative Classification”, Data Mining. Expert Systems with Applications (ESWA), vol. 41, pp 5948-5959, 2014.

[22] W. Hadi, F. Aburrub, and S, Alhawari, “A new fast associative classification algorithm for detecting phishing websites”, Applied Soft Computing, vol. 48, pp 729-734, 2016.

[23] A. D. Kulkarni and L. Brown, “Phishing Websites Detection using Machine Learning”. Journal of Advanced Computer Science and Applications, vol. 10, no. 7, pp. 8-13, 2019.

[24] D. Sahoo, C. Liu, and C. H. Hoi, “Malicious URL detection using machine learning: A Survey”, https://arxiv.org/abs/1701.07179, 2017.

[25] H. Yang, Q. Liu, S. Zhou, and Y. Luo, “A spam filtering method based on multi-modal fusion, Appl. Sci. vol. 9, 2019.
doi:10.3390/app9061152.

[26] R. Chriramdasu, G. Srivastava, S. Bhattacharya, P. K.Reddy, T. R. Gadekallu, “A machine learning driven threat intelligence system for malicious URL detection”, Proceedingd of 16th International Conference on Availability, Reliabilty, and Security, August 2021, Aritical 154, pp 1-7, doi.org/10.1145/3465481.3470029.

[27] UCI Machine Learning Repository: Website Phishing Data Set (Online) https://archive.ics.uci.edu/ml/datasets/Website+Phishing.