Classification of Phishing Websites Based on Machine Learning Techniques

Safa Orhan
Computer Engineering Department
Istanbul Kultur University
Istanbul, Turkey
1600003764@stu.iku.edu.tr

Eyüp Usta
Computer Engineering Department
Istanbul Kultur University
Istanbul, Turkey
1600003762@stu.iku.edu.tr

***Abstract*— Phishing is attack aimed at stealing people's information. Phishing attacks send a legitimate-looking email to e-mails, and when users click the URL in that mail, their information is stolen. Many models have been developed against phishing attacks. However, in phishing attacks, people are the weak link, attackers exploit vulnerabilities, and attackers keep their attacks up-to-date. These and other problems require these models to be constantly updated and developed. We proposed one of models that emerged using Machine Learning (ML) and Deep Learning (DL) algorithms against phishing attacks. Our aim in this project was to find the model that gives the closest result to 100% in comparison methods. Also, in this study, we prefered our dataset achieved from the UCI Machine Learning Repository site. Preferred dataset consisted of 30 features and 11055 different samples. For this purpose, we first trained our data set with ML ve DL algorithms and tested them using comparison methods. As a result, we have proposed our model, which we think will give the best results against phishing attacks in line with the results we have achieved. In other words, the results show that the model we propose will be successful against phishing attacks.**

Keywords—Machine Learning (ML); Phishing; Deep Learning (DL); Accuracy; Dataset

I. INTRODUCTION

Phishing aims to obtain users' credentials by cheating them via presenting fake web pages which appearing legitimate sites. For example, you receive mail from a bank or a company to your e-mail address. You trust it to be a well-known organization and click the URL in the incoming mail and your information is stolen. Phishing detection mechanisms are grouped into heuristics based, visual similarity based, list based, and Machine Learning (ML) based technical categories. This study is based on ML and Deep Learning (DL) techniques. In this project, a dataset will be used to compare ML and DL methods while detecting phishing websites. Also, precision, accuracy, F1-score, and recall will be evaluated to check DL and ML algorithms.

The purpose of this project is to classify phishing sites using ML and DL techniques, to facilitate the detection of such sites. To develop various models in the classification process. To reach the best result by testing these developed models on the data set. There are many problems in phishing such as people, security vulnerabilities, and attackers constantly updating themselves. As a result of our work, we will propose a model against attackers trying to exploit people's weaknesses using seemingly safe URLs. Thus, it will be more effective against fake e-banking, e-commerce style URLs coming via e-mail.

The objective of the project is to find and develop the model that gives the best result, that is, the closest to 100% value. It is our success criterion that the model we will obtain as a result of the studies has the closest value to 100% in comparison methods. It is impossible to obtain 100% value in such a study.

In the related work part, we will first look at the methods of current work on phishing. Then we will look at the general problems of current studies. Finally, we will compare the methods in existing studies with our own method. The methodology part will follow. In this part, we will first talk about the studies we tested ML and DL techniques on our data set. Also, we will talk about ML and DL techniques we use in this study. We will also explain our dataset. Finally, we will talk about the model we proposed. The experimental result will follow this part. In this part, we will talk about the results of the model or models we have proposed. Finally, the conclusion part will come. In the conclusion part, we will talk about the results we obtained in our study in general.

II. RELATED WORK

We will examine existing systems and their overall problems and compare them with our proposed method in this part. P. Yang and his friends [12] used a network of convolutional and recurrent layers and used character level embeddings. They achieved an accuracy of 98.61%. Also, they used a dataset of about one million samples. Kumar and his friends [10] used three different models which are Logistic Regression, Gaussian Naïve Bayes (GNB), and Random Forest. They achieved accuracy of 97.7% in Logistic Regression, accuracy of 98.03% in the Random Forest, and accuracy of 97.18% in Gaussian Naïve Bayes. They also used a data set that contains one million URLs. S. Yang used a model that contains 5 LSTM layers with 128 nodes each and a regular CNN model in this research [16]. He achieved accuracy of 99.14% in the LSTM model and accuracy of 97.42% in the CNN model. Adebowale and his friends [4] used trained CNN-LSTM network and achieved accuracy of 93.28%. Also, they

used a dataset containing one million URL samples, and a dataset of ten thousand images that they collected themselves from legitimate and phishing sites as the dataset.

Of course, while developing such systems, they are developed according to some problems. The weakest link is people for phishing sites, because people think that the URL in a mail sent to their email is legitimate and click it. Later they are redirected to a site and thus their information is captured [6]. Attackers are aware that the method they use will be resolved after a while, so they are constantly updating their methods [25]. Attackers try to make the URL look legit because they try to exploit technical vulnerabilities [6]. List-based methods keep the positive rate low but are not comprehensive. Therefore list-based methods cannot detect an attack on day zero [7]. If the feature sets are not updated regularly, it will not be able to detect new attacks after some point [7].

In Table I, you can see the comparison between our models and the models of existing systems, according to the comparison methods. Ref. [10] has achieved the highest precision according to our machine learning model but our machine learning model has achieved the higher values according to Ref. [10] in other comparison methods. Our deep learning model has not achieved the best scores of the metrics according to other proposed approaches. Our combined model and Ref. [4] use the same neural networks, although our model performed better than Ref. [4] in all comparison methods.

TABLE I. COMPARISON OF METHODS

Model	Accuracy	Precision	Recall	F1-Score
Our Random Forest Model	99.09%	0.9895	0.9940	0.9917
Our RNN Model	97.27%	0.9769	0.9694	0.9732
Our Combined Model	96.26%	0.9635	0.9678	0.9656
[12]	98.61%	0.9941	0.9857	0.99
[4]	93.28%	0.9327	0.9330	0.9329
Random Forest [10]	98.03%	1.0	0.96	0.98
Logistic Regression [10]	97.7%	1.0	0.96	0.98
Gaussian Naïve Bayes [10]	97.18%	1.0	0.95	0.97
CNN [16]	97.42%	0.9648	0.9723	
LSTM [16]	99.14%	0.9874	0.9891	

III. METHODOLOGY

We used ML and DL algorithms in this study. In this part, we will talk about this algorithm and our models. Also, we will talk about our dataset.

The basic concept in the Random Forest is strength out of unity. In other words, the community consisting of many decision trees is called the Decision Tree. Each tree gives a prediction and use the class with the most votes as model in the Random Forest. Eq. (1) is the norm formula. Eq. (2) is the Random Forest classification formula. X is the set of all features, and Y is the set of all trees. Additionally, T is the total number of trees.

$$norm\varphi_i = \frac{\varphi_i}{\sum_{j \in X} \varphi_j} \tag{1}$$

$$RF\varphi_i = \frac{\sum_{j \in Y} norm\varphi_{ij}}{T} \tag{2}$$

We achieved the best accuracy value by combining methods such as max features, criterion, class weight, and warm start. We tested the combinations we made on our dataset [32] we got from the UCI machine learning repository site. As the test size, we used the values between 15% and 35% in increments of five. As a result, the combination that works the best is "max features: auto, criterion: gini, class weight: balanced subsample, warm start: true". In addition, the combination that works worst is "max features: none, criterion: entropy, class weight: balanced, warm start: true".

Recurrent Neural Networks (RNN) is an Artificial Neural Networks using sequential data and uses training data to learn. Also, RNN gathers information from previous entries to influence the current entry and exit. These weights are adjusted in back propagation and gradient descent processes to facilitate reinforcement learning. RNN uses the backpropagation through time (BPTT) algorithm to determine gradients. BPTT calculates errors in each layer and allowing us to set and fit the parameters of the model accordingly. Eq. (3) is the activation formula. Eq. (4) is the RNN formula and eq. (5) is the BPTT formula. t is the time step. a^t is activation. y^t is output. In addition, g_1 and g_2 are activation functions.

$$a^t = g_1(W_{aa}a^{t-1} + W_{ax}x^t + b_a) \tag{3}$$

$$y^t = g_2(W_{ya}a^t + b_y) \tag{4}$$

$$\frac{\partial \mathcal{L}^T}{\partial W} = \sum_{t=1}^{T} \frac{\partial \mathcal{L}^T}{\partial W}\Big|_t \tag{5}$$

In Fig. 1, the model is tested with the 320 different combinations of activation functions, optimizers, and loss functions. In the combinations that are using the FTRL as an optimizer or relu, selu and elu as an activation function in the output layer performed accuracy below 55%. In other combinations model performed accuracy between 90% and 98%. The best performed model combination is binary cross entropy as loss function, adamax as optimizer. Of course, we tested the combinations we made on our dataset [32] we got from the UCI machine learning repository site.

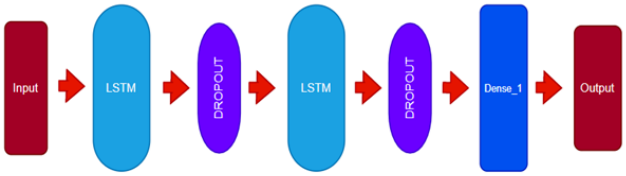

Fig. 1. RNN Model

Convolutional Neural Networks (CNN) is made up of numerous layers of neurons, and artificial neurons compute the weighted sum of multiple inputs. Also, artificial neurons give an activation value. Each of the CNN layers creates several activation maps. Created activation map highlights the relevant property of the data. Each of the neurons multiplies and adds the input by their weight. It then runs them with the help of the activation function. The process of multiplication and addition is called convolution. Last layer output is the next layer input. The classification layer takes the output of the last convolution layer as input. The classification layer determines the result with values between 0-1. Eq. (6) is the dimension formula. Eq. (7) is the convolutional formula. n_h is the size of the height, n_w is the size of the width and n_c is the number of channels.

$$\dim(image) = (n_h, n_w, n_c) \tag{6}$$

$$conv(I,K)_{x,y} = \sum_{i=1}^{n_h} \sum_{j=1}^{n_w} \sum_{k=1}^{n_c} K_{i,j,k} I_{x+i-1,y+j-1,k} \tag{7}$$

In Fig. 2, the Combined Model is tested with "activation function in the output layer: softplus, loss function: binary cross entropy, optimizer: adamax" and epoch number of 150. Of course, we tested the model we made on our dataset [32] we got from the UCI machine learning repository site.

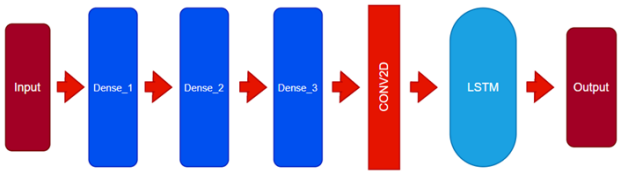

Fig. 2. Combined Model

The models we created using ML and DL gave good results in all comparison methods. While RNN Model and Combined Model gave results above 96% in all comparison methods, Random Forest gave results above 98%. Therefore, the model we propose in this study is our Random Forest model.

One of the most faced difficulties by the research world is the rarity of trustable datasets. Therefore, we used our dataset [32] we got from the UCI machine learning repository site. In the Phishing Websites Data Set, there are a total of 11055 website samples, 6157 of which are legitimate and 4898 of which are phishing. Data set have 30 features and the features are divided into 4 categories. In Table II, you can see the categories and values of the features in detail.

TABLE II. CATEGORIES AND VALUES OF FEATURES

Category	Feature	Values
Address Bar	Using the IP Address	-1, 1
	Long URL to Hide the Suspicious Part	-1, 0, 1
	Links in <Meta>, <Script> and <Link> tags	-1, 1
	URL's having "@" Symbol	-1, 1
	Redirecting using "//"	-1, 1
	Adding Prefix or Suffix Separated by (-) to the Domain	-1, 1
	Sub Domain and Multi Sub Domains	-1, 0, 1
	HTTPS	-1, 0, 1
	Domain Registration Length	-1, 1
	Favicon	-1, 1
	Using Non-Standard Port	-1, 1
	The Existence of "HTTPS" Token in the Domain Part of the URL	-1, 1
Abnormal	Request URL	-1, 0, 1
	URL of Anchor	-1, 0, 1
	Links in <Meta>, <Script> and <Link> tags	-1, 0, 1
	Server Form Handler (SFH)	-1, 0, 1
	Submitting Information to Email	-1, 1
	Abnormal URL	-1, 1
HTML and JavaScript	Website Forwarding	-1, 0, 1
	Status Bar Customization	-1, 1
	Disabling Right Click	-1, 1
	Using Pop-up Window	-1, 1
	IFrame Redirection	-1, 1
Domain	Age of Domain	-1, 1
	DNS Record	-1, 1
	Website Traffic	-1, 0, 1
	PageRank	-1, 1
	Google Index	-1, 1
	Number of Links Pointing to Page	-1, 1
	Statistical-Reports Based Feature	-1, 1

We used comparison methods such as accuracy, precision, recall and F1-score to compare DL and ML algorithms. The values used in the calculation of these methods; True Positive (TP), False Negative (FN), True Negative (TN), and False Positive (FP).

The ratio of correctly categorized samples' total number to samples' total number is called accuracy [28]. Eq. (8) is an accuracy formula.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{8}$$

The ratio of phishing's number detected by the model to phishing's total number is called precision [28]. Eq. (9) is the precision formula.

$$Precision = \frac{TP}{TP + FP} \tag{9}$$

The ratio of how accurately the model predicts is called recall. [28]. Eq. (10) is the recall formula.

$$Recall = \frac{TP}{TP + FN} \tag{10}$$

The ratio of twice the products of recall and precision to their sum is called F1-score [28]. Eq. (11) is the F1-score formula.

$$F1\ Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{11}$$

IV. Experimental Results

The experiments are done by using Tensorflow, Keras and Sklearn packages with Python programming language. The dataset [32] is used train the models. 70% of the data is randomly selected as train set and 30% of the data is randomly selected as the test set. To compare the models and to evaluate the performance of the models F1 score, recall, precision, accuracy, and confusion matrix techniques are used with the test data. The details about the deep learning models is shown in Table III. Also, the accuracy values of the models we use for Deep Learning, used the features, epoch values and test sizes are shown in Table IV. In addition, you can see confusion matrix of RNN Model in Fig. 3 and can see confusion matrix of Combined Model in Fig. 4.

TABLE III. Parameters, training and test accuracy values, epoch values, and test sizes used in models.

Model	Parameters	Train Accuracy	Test Accuracy	Epoch	Test Size
RNN Model	Sigmoid/Binary Cross Entropy/Adamax	98.31%	97.20%	150	30%
Combined Model	Softplus/Binary Cross Entropy/Adamax	98.54%	96.26%	150	30%

TABLE IV. Model features, epoch values, test sizes and accuracy values

Model	Feature	Accuracy	Test Size	Epoch
RNN Model	Non-categorical	98.31%	30%	150
Combined Model	Non-categorical	96.26%	30%	150

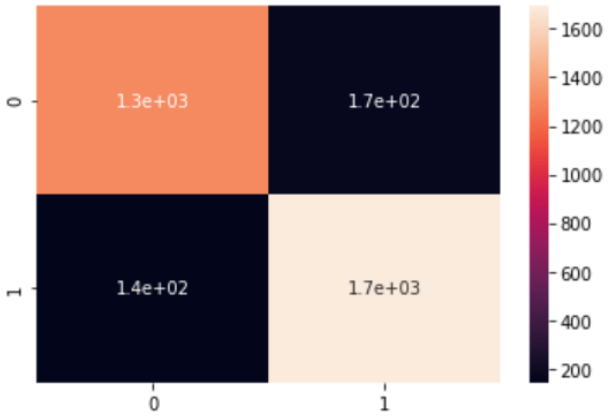

Fig. 3. RNN Model's Confusion Matrix

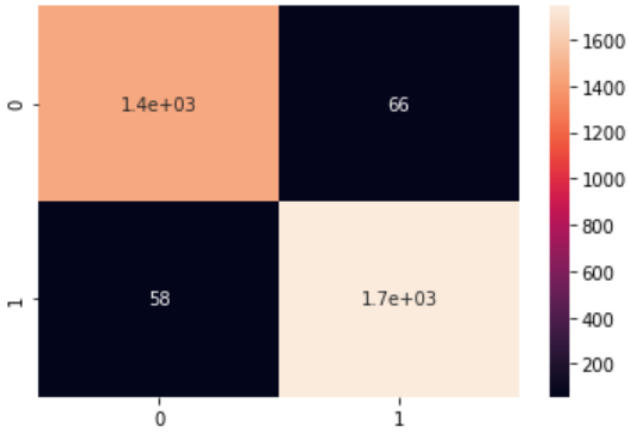

Fig. 4. Combined Model's Confusion Matrix

The test size for which we got the best accuracy value was 25% in the Random Forest. When we examine Fig. 5, the class weight that gives the worst average accuracy value is none, and class weight, which gives the best average accuracy value, is also a balanced subsample. Here we see that the accuracy values are low in combinations made with none. The result is better in combinations with balanced. The combination that works the best is "max features: auto, criterion: gini, class weight: balanced subsample, warm start: true" and this combination has an accuracy of 99.09%. The combination that works worst is "max features: none, criterion: entropy, class weight: balanced, warm start: true" and this combination has an accuracy of 96.68%. We used 240 different model combinations in the random forest and achieved an average accuracy of 97%. The best performed combinations of machine learning model is tested with 10 folds cross validation, Table V, to get more an accurate accuracy on our dataset. You can also see the confusion matrix in Fig. 6.

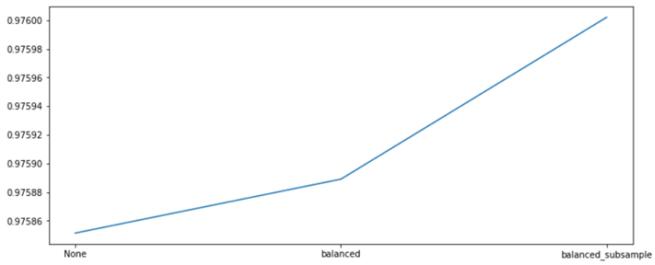

Fig. 5. Average accuracy values according to class weights

TABLE V. COMPARISON OF ACCURACY VALUES AND CROSS-VALIDATION ACCURACY VALUES ACCORDING TO RANDOM FOREST

Model	Accuracy	Cross-Validation Accuracy
Random Forest	99.09%	97.24%

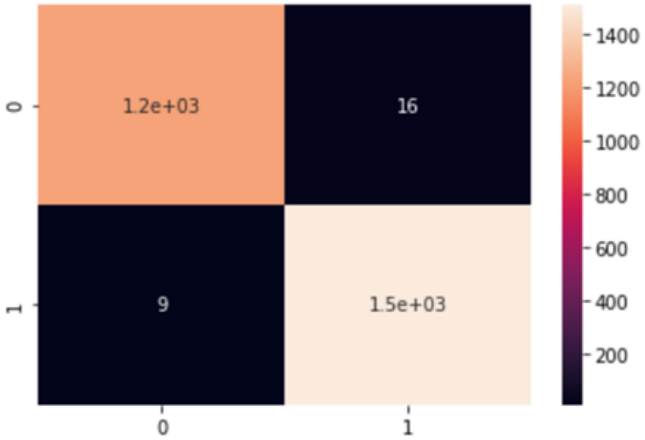

Fig. 6. Random Forest's Confusion Matrix

Finally, you can see results of models according to the comparison methods in Table VI. Additionally, you can see the best accuracy value of our models in Fig. 7, can see the best F1-score value of our models in Fig. 8, can see the best precision value of our models in Fig. 9 and can see the best recall value of our models in Fig. 10.

TABLE VI. F1 SCORE, PRECISION, RECALL VALUES OF ML AND DL TECHNIQUES USED.

Model	F1 Score	Precision	Recall
Random Forest	0.9917	0.9895	0.9940
RNN Model	0.9732	0.9769	0.9694
Combined Model	0.9656	0.9635	0.9678

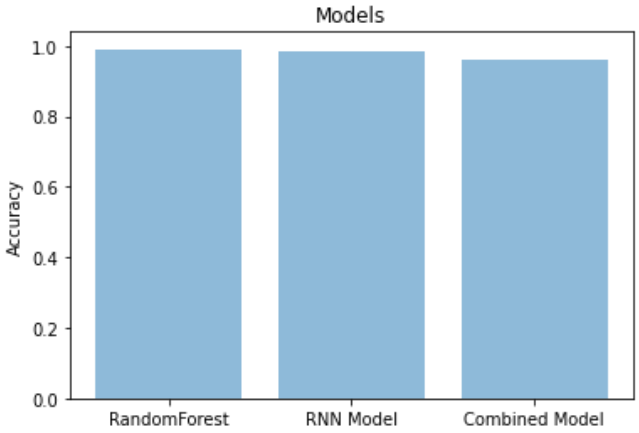

Fig. 7. Models' best accuracy value

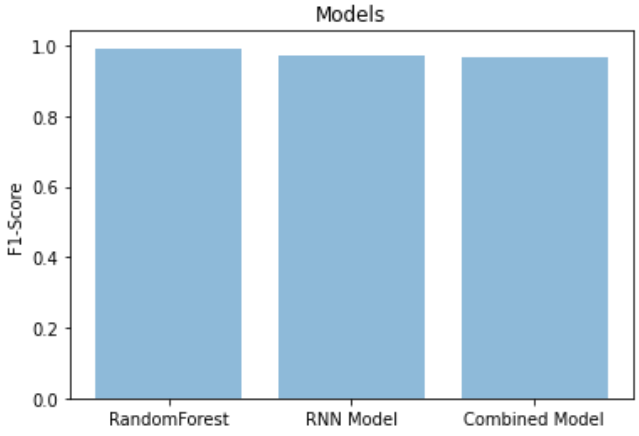

Fig. 8. Models' best F1-score value

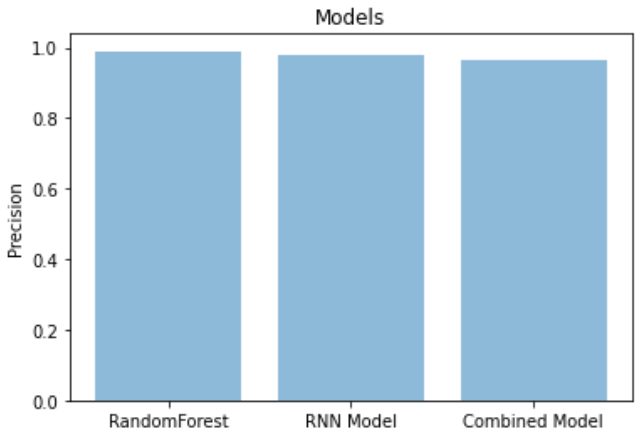

Fig. 9. Models' best precision value

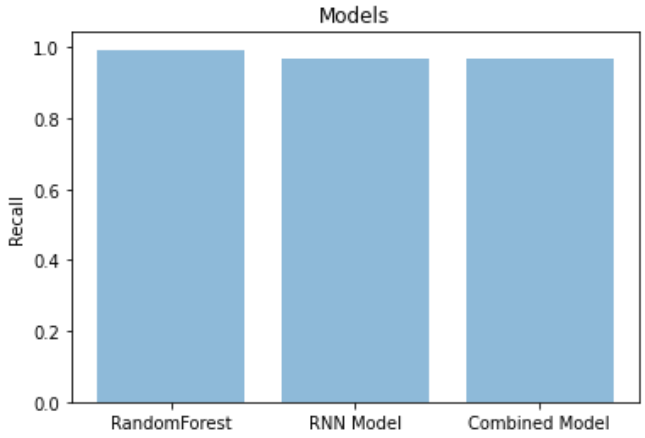

Fig. 10. Models' best recall value

V. Conclusions and Future Work

Phishing aims to steal innocent users' credentials by serving fake web pages that impersonate their legitimate targets. There are many problems in phishing such as people, security vulnerabilities, and attackers constantly updating themselves. We proposed a model that is successful against phishing attacks using ML and DL algorithms. To compare the ML and DL algorithms, we used f-measure, precision, recall, accuracy, and confusion matrix. DL algorithms have good accuracy values, but other metrics such as f-measure, precision and recall did not give good results. For this reason, we did not recommend DL models. Nevertheless, the DL algorithm that worked the best was RNN. Although ML algorithms give good results in all comparison methods, we recommended it because Random Forest gave the best result. Since it is easy to create trees in the Random Forest, we concluded that the models that give the best results are in the Random Forest. Of course, we think that the reason why this is so likely to be due to the structure of our data set. As a result, the model we propose will be successful against phishing attacks. Of course, in the future, considering that the attackers are constantly updating the attacks, we will have to refresh our data set and test it to see if the model still works.

Acknowledgements

We would like to acknowlede the effort of Assis. Prof. Dr. Öznur Şengel for her guidance. Express appreciation to all of those authors whose references we used in this research work. Also, we would like to thank Prof. Dr. Özgür Koray Şahingöz and Assis. Prof. Dr. Bahar İlgen for their contributions throughout our university education life. Finally, we would also like to thank IKU Computer Engineering students Batuhan Üçsu and Taha Anıl Pulat, who have been with us for four years and have always been with us.

References

[1] B. Wei, R.A. Hamad, L. Yang, X. He, H. Wang, B. Gao, and W.L. Woo, "A Deep-Learning-Driven Light-Weight Phishing Detection Sensor", Basel, Switzerland: MDPI, pp. 1-13, September 2019.

[2] R.M. Mohammad, F. Thabtah, and L. McCluskey, "An Assessment of Features Related to Phishing Websites using an Automated Technique", in *The 7th International Conference for Internet Technology and Secured Transactions*, London, UK, December 2012: IEEE, pp. 492-497.

[3] G. Vrbančič, I. Jr. Fister, and V. Podgorelec, "Datasets for phishing websites detection", Maribor, Slovenia: Elsevier, pp. 1-7, October 2020.

[4] M.A. Adebowale, K.T. Lwin, and M.A. Hossain, "Deep Learning with Convolutional Neural Network and Long Short-Term Memory for Phishing Detection", in *13th International Conference on SKIMA*, Island of Ukulhas, Maldives, August 2019: IEEE, pp. 1-8.

[5] M. Korkmaz, O.K. Sahingoz, and B. Diri, "Detection of Phishing Websites by Using Machine Learning-Based URL Analysis", in *11th International Conference on Computing, Communication and Networking Technologies*, Kharagpur, India, July 2020: IEEE, pp. 1-7.

[6] A. Razaque, M.B.H. Frej, D. Sabyrov, A. Shaikhyn, F. Amsaad, and A. Oun, "Detection of Phishing Websites using Machine Learning", in *Cloud Summit*, Harrisburg, PA, USA, October 2020: IEEE, pp. 103-107.

[7] C. Opara, B. Wei, and Y. Chen, "HTMLPhish: Enabling Phishing Web Page Detection by Applying Deep Learning Techniques on HTML Analysis", in *International Joint Conference on Neural Networks*, Glasgow, UK, July 2020: IEEE, pp. 1-8.

[8] S.O. Folorunso, F.E. Ayo, K-K.A. Abdullah, and P.I. Ogunyinka, "Hybrid vs Ensemble Classification Models for Phishing Websites", Baghdad, Iraq: Iraqi Journal of Science, pp. 3387-3396, January 2020.

[9] R.M. Mohammad, F. Thabtah, and L. McCluskey, "Intelligent rule-based phishing websites classification", Elsevier, pp. 153-160, 2014.

[10] J. Kumar, A. Santhanavijayan, B. Janet, B. Rajendran, and B.S. Bindhumadhava, "Phishing Website Classification and Detection Using Machine Learning", in *International Conference on Computer Communication and Informatics*, Coimbatore, India, January 2020: IEEE, pp. 1-6.

[11] W. Bai, "Phishing Website Detection Based on Machine Learning Algorithm", in *International Conference on CDS, Stanford*, CA, USA, August 2020: IEEE, pp. 293-298.

[12] P. Yang, G. Zhao, and P. Zeng, "Phishing Website Detection Based on Multidimensional Features Driven by Deep Learning", IEEE, pp. 15196-15209, January 2019.

[13] S. Anupam, and A.K. Kar, "Phishing website detection using support vector machines and nature-inspired optimization algorithms", New Delhi, India: Springer, pp. 17-32, November 2020.

[14] D.R. Ibrahim, and A.H. Hadi, "Phishing Websites Prediction Using Classification Techniques", in *International Conference on New Trends in Computing Sciences*, Amman, Jordan, October 2020: IEEE, pp. 133-137.

[15] N. Abdelhamid, A. Ayesh, and F. Thabtah, "Phishing detection based Associative Classification data mining", Elsevier, pp. 5948-5959, 2014.

[16] S. Yang, "Research on Website Phishing Detection Based on LSTM RNN", in *4th Information Technology, Networking, Electronic and Automation Control Conference*, Chongqing, China, June 2020: IEEE, pp. 284-288.

[17] G. Vrbančič, I. Jr. Fister, and V. Podgorelec, "Swarm Intelligence Approaches for Parameter Setting of Deep Learning Neural Network: Case Study on Phishing Websites Classification", in *International Conference on Web Intelligence, Mining and Semantics*, New York, USA, June 2018: ACM, pp. 1-8.

[18] K.M.Z. Hasan, Md. Z. Hasan, and N. Zahan, "Automated Prediction of Phishing Websites Using Deep Convolutional Neural Network", in *International Conference on Computer, Communication, Chemical, Materials and Electronic Engineering*, Rajshahi, Bangladesh, July 2019: IEEE, pp. 1-4.

[19] P. Vaitkevicius, and V. Marcinkevicius, "Comparison of Classification Algorithms for Detection of Phishing Websites", Vilnius, Lithuania: Vilnius University, pp. 143-160, 2020.

[20] S. Zaman, S.MU. Deep, Z. Kawsar, Md. Ashaduzzaman, and A.I. Pritom, "Phishing Website Detection Using Effective Classifiers and Feature Selection Techniques", in *2nd International Conference on Innovation in Engineering and Technology*, Dhaka, Bangladesh, December 2019: IEEE, pp. 1-6.

[21] R.M. Mohammad, and F. Thabtah, “Predicting phishing websites based on self-structuring neural network”, London, UK: Springer, pp. 443-458, November 2013.

[22] C.L. Tan, “Phishing Legitimate Full”, URL: https://data.mendeley.com/, March 2018.

[23] N. Abdelhamid, F. Thabtah, and H. Abdel-jaber, “Phishing detection: A recent intelligent machine learning comparison based on models content and features”, in *International Conference on ISI*, Beijing, China, July 2017: IEEE, pp. 72-77.

[24] A. Subasi, E. Molah, F. Almkallawi, and T.J. Chaudhery, “Intelligent phishing website detection using random forest classifier”, in *International Conference on Electrical and Computing Technologies and Applications*, Ras Al Khaimah, United Arab Emirates, November 2017: IEEE, pp. 1-5.

[25] A. Basit, M. Zafar, X. Liu, A.R. Javed, Z. Jalil, and K. Kifayat, “A comprehensive survey of AI-enabled phishing attacks detection techniques”, Springer, pp. 139-154, 2021.

[26] T. Rasymas, and L. Dovydaitis, “Detection of Phishing URLs by Using Deep Learning Approach and Multiple Features Combinations”, Baltic Journal of Modern Computing, pp. 471-483, September 2020.

[27] A.C. Bahnsen, E.C. Bohorquez, S. Villegas, J. Vargas, and F.A. Gonzalez, “Classifying Phishing URLs Using Recurrent Neural Networks”, in *APWG Symposium on Electronic Crime Research*, Scottsdale, AZ, USA, April 2017: IEEE, pp. 1-8.

[28] W. Wang, F. Zhang, X. Luo, and S. Zhang, “PDRCNN: Precise Phishing Detection with Recurrent Convolutional Neural Networks”, Wiley-Hindawi, pp. 1-16, October 2019.

[29] J. Saxe, and K. Berlin, “eXpose: A Character-Level Convolutional Neural Network with Embeddings For Detecting Malicious URLs, File Paths and Registry Keys”, New York, USA: arXiv, pp. 1-18, February 2017.

[30] H. Le, Q. Pham, D. Sahoo, and S.C.H. Hoi, “URLNet: Learning a URL Representation with Deep Learning for Malicious URL Detection”, New York, USA: arXiv, pp. 1-13, March 2018.

[31] O.K Sahıngoz, S.I. Baykal, and D. Bulut, “Phishing Detection from URLs by Using Neural Network”, İstanbul, Turkey: IKU, pp. 41-54, 2018.

[32] R.M. Mohammad, L. McCluskey, and F. Thabtah, “Training Dataset”, URL: https://archive.ics.uci.edu/ml/index.php, March 2015.

[33] N. Abdelhamid, “Phishing Data”, URL: https://archive.ics.uci.edu/ml/index.php, November 2016.

[34] S. Banik, “Dataset”, URL: https://www.kaggle.com/, January 2021.

[35] A. Hannousse, and S. Yahiouche, “dataset B 05 2020”, URL: https://data.mendeley.com/, September 2020.

[36] M. Adebowale, “Text Frame Image Features”, URL: https://data.mendeley.com/, December 2019.

[37] M.S.I. Mamun, M.A. Rathore, A.H. Lashkari, N. Stakhanova, and A.A. Ghorbani, “Phishing”, URL: https://www.unb.ca/, 2016.