Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 07, No. 01, March 2021.

# PHIBOOST- A NOVEL PHISHING DETECTION MODEL USING ADAPTIVE BOOSTING APPROACH

Ammar Odeh, Ismail Keshta and Eman Abdelfattah

*(Received: 14-Sep.-2020, Revised: 6-Nov.-2020, Accepted: 15-Dec.-2020)*


## *ABSTRACT*

*Every day, cyberattacks increase and use different strategies. One of the most common cyberattacks is Phishing, where the attacker collects sensitive and confidential information by pretending as a trusted party. Different traditional strategies have been introduced for anti-phishing, such as blacklisted, heuristic search and visual similarity. Most of these traditional methods have a high false rate and take a long time to detect the phishing website. New modes have been introduced using machine learning techniques which improve the detection's accuracy. Machine learning techniques require a huge amount of data called features that are collected from different websites. These collected features are classified into four categories. This paper introduces a novel detection model by utilizing features' selection to pick up the highly correlated features with the class label. The phase of features' selection employs independent significance features library from MATLAB and heat-map from Python to find the highly correlated features. Then, the proposed model uses an adaptive boosting approach which consists of multiple classifiers to increase the model's accuracy. The proposed model produces an extremely high predictive accuracy of approximately 99%.*



## *KEYWORDS*

*Adaptive boost, Feature selection, Correlation-based feature, Machine learning.*


## 1. INTRODUCTION

Phishing is against the law. It uses social engineering and technical trick to thieve Internet users' non-public identity facts and financial account credentials. Social engineering schemes prey on unwary sufferers with the means of not only fooling them into believing they're managing a trusted and a legitimate party, but also using misleading electronic mail addresses and electronic mail messages [1]-[2].

Disasters have continually been a good chance for different types of criminals' special cyberattacks. The phishers have created violations to take advantage of hurricanes, recessions and different challenging times, merchandising fake charitable giving possibilities and nonexistent services or products. One of the most recent world catastrophes in 2020 is the COVID-19 pandemic. Anti-Phishing Working Group (APWG) classifies four cybercriminal methods that represent more complicated scenarios to lure their victims [3]-[4].

Several types of research introduced phishing attack problems and their consequences on customer trust in e-commerce and online services [5]. The phishing attackers create a website that pretends as a trusted website to collect valuable and sensitive Internet user information. At the same time, different anti-phishing software models for phishing detections are introduced. The phishing detection strategies are classified into seven categories [6] as follows:

1. User education: this category depends on the educated Internet users to distinguish between a legitimate and a phishing website [7].
2. Create a blacklist: this strategy creates centralized phishing websites and compares an URL with the list to find out if the URL is legitimate or not [8].
3. Heuristic blacklist methods: in this strategy, the system identifies the signature of the phishing URL and blacklists it for the future use of intrusion detection systems [9].
4. Visual similarity: These techniques use URL features to find out the similarity between websites (page source code, images, textual content, text formatting, HTML tags, CSS, website logo).

1. A. Odeh is with Computer Science Department, Princess Sumaya Uni. for Technology, Amman, Jordan. Email: a.odeh@psut.edu.jo
2. I. Keshta is with Comp. Sci. and Information Systems Department, AlMaarefa Uni., Riyadh, KSA. Email: imohamed@mcst.edu.sa
3. E. Abdelfattah is with School of Theoretical & Applied Science, NJ, USA.

After that, the system compares the new website with previously visited ones and distinguishes whether it is a legitimate or a phishing website [10].

5. Search engine-based techniques: in this mode, the system uses the search engine and extracts the website features, then checks the website legitimacy. However, the search engine does not give precise output for the non-English search query [11].
6. Supervised Machine Learning detection system uses supervised machine learning models on phishing datasets with predefined features [12].
7. Deep learning techniques: these techniques include Gated Recurrent Neural Network (GRU) and Convolutional Neural Network (CNN). Based on these techniques, the system automatically extracts the features from generic URL, file directory, ...etc. [13].

Table 1 shows a summary of phishing detection strategies and their main drawbacks.

Table 1. Phishing detection strategies.

| | Phishing detection strategies | Problem |
|---|---|---|
| 1 | User education | • Fail to detect a new phishing attack |
| 2 | Create a blacklist | • Produce high false positive rate |
| 3 | Heuristic blacklist methods | |
| 4 | Visual similarity | • Complicated<br>• Slow in nature |
| 5 | Search engine-based techniques | • Not fit for real-time environment<br>• Language dependence |
| 6 | Supervised machine learning detection | • The achieved performance depends on the features' selection and the classification algorithms |
| 7 | Deep learning techniques | |

The rest of the paper is organized as follows: In Section 2, a review of related work is presented. In Section 3, the proposed methodology is described. In Section 4, the experimental results are reported. The conclusion of the paper is included in Section 5.

## 2. LITERATURE REVIEW

Different research papers have conducted an intensive work on website security, some of which manipulated the routing security [14], while others dealt with intrusion detection, intrusion prevention and smart grid security [15].

Pawan Parakash et al. proposed two methods to identify phishing websites, where first proposed method introduced five heuristics to enumerate the combination of the known phishing websites to find out the new phishing websites. The second method used matching algorithms to find out the new phishing websites [16].

Samuel Marchal et al. analyzed and evaluated the URL of the websites and extracted the features of the URL. Based on the several queries through Google and Yahoo search engines, the authors determined the keywords for each website. Then, the keywords with the extracted features are used in a machine learning classification algorithm to find out the phishing websites from the real dataset [17]. In [18], the authors introduced models using machine learning and data mining algorithms for detecting website phishing.

The authors in [19] used the artificial neural network to spot phishing websites. The proposed work used 17 neurons as input for 17 characteristics and one hidden layer level and two neurons as output to decide whether or not the website is phishing. The dataset was divided into 80 percent as a training set and 20 percent as a test set. The suggested model achieved 92.48 percent accuracy.

Authors in [20] introduced a model relying on a machine learning technique called PLIFER. This model requires an age of the URL domain. Also, ten features are extracted and Random Forest (RF) model is used to identify the phishing website. 96 percent of phishing e-mails were correctly identified by this

model. Classification models are also used to identify phishing utilizing labeled datasets. Different classification methods used features, like URL-based and text-based applications.

A proposed software collection model hybrid set of features (HEFS) to identify phishing websites relying on machine learning algorithms is presented in [21]. A cumulative distribution gradient technique is used to extract the primary feature set. Then, the second set of features is extracted using a method called data perturbation ensemble. Random Forest (RF), an ensemble learner, is subsequently implemented to identify phishing websites. The results indicated that HEFS identified phishing features with a precision of up to 94.6 percent.

In 0, The authors selected the most suitable components to identify website phishing and proposed two new selection methods or detection techniques based on machine learning algorithms. The two methods include the AdaBoost classifier and the LightGBM classifier. When combined, they form a hybrid classifier. These two algorithms have proved to be effective and efficient in improving the accuracy of single classifiers in detecting web phishing attacks.

In 0, The authors investigated agreeing on the final conclusion of the features used to detect phishing on webpages. Using three standard datasets, the authors used the Fuzzy Rough Set (FRS) theory as a tool to select the most significant features to identify intrusion on webpages. The chosen features were then fed into three standard classifiers to detect phishing. When Random Forest classification was used, the maximum accuracy gained by Fuzzy Rough Set (FRS) feature selection was 95%. The Fuzzy Rough Set (FRS) had used three sets of data to come up with nine universal features of detecting phishing. When these versatile features were used to measure the accuracy value, the accuracy was about 93%, which is comparable to the Fuzzy Rough Set performance, with only a slight difference of 2%.

The authors of 0 proposed three ensemble learning models based on Forest Penalizing Attributes (Forest PA) algorithm. The algorithm exploited the prowess of all attributes in a given set of data using a weight increment and weight assignment strategy to build highly resourceful decision trees. The results of the experiment showed highly efficient meta-learners with an accuracy of 96.26%.

## 3. MOTIVATION AND MAIN CONTRIBUTION

All phishing attacks have some salient features; however, these attacks exhibit some similarities and patterns. Thus, using machine learning methods to detect these similar patterns and recognize phishing websites has become possible 00.

In this paper, an inventive detection model is introduced that utilizes feature selection to pick up similar features on phishing websites with the class label. The independent significant features library from MATLAB and heat-map from python are employed in the features' selection to find the associated features on phishing websites. The proposed novel detection model consists of multiple classifiers incorporated in an adaptive boosting technique to increase the model's accuracy.

The adaptive AdaBoost classifier is selected as an efficient technique for detecting website phishing, because it is flexible and straightforward, yet it has a high generalization performance 0-0. The fact that it is based on several weak classifiers makes it flexible and straightforward to implement. Also, it doesn't use large sets of features that may be unnecessary sometimes, but it treats each class's attributes separately 00. Moreover, the AdaBoost classifier achieves much high accuracy, as it regulates the errors of weak classifiers; therefore, it needs much fewer settings as compared to other robust classifiers 0-0.

## 4. PRELIMINARIES

This section provides a brief description of the phishing dataset used in the experimental comparison, as well as a background about the dataset, feature selection and the classification model used in this study.

### 4.1 Dataset

The dataset used is collected from the PhishTank archive [22], MillerSmiles archive [23] and Google searching operators. The phishing dataset consists of 30 features, as listed in Table 2. All of these features were classified into four categories: Address Bar Features (1-12), Abnormal Based Features

(13-18), HTML and JavaScript-based Features (19-23) and Domain-based Features (24-30). The last feature is the label column, which represents the class of the website as either phishing or legitimate.

Table 2. Feature classes of the dataset.

| Feature class | Description |
|---|---|
| Address Bar | Feature of Uniform Resource Locator (URL) such as IP address |
| Abnormal Based | Feature of abnormal activities such as URL of tag (Anchor) |
| HTML and JavaScript-based | Feature of HTML and Jscript embedded in the page source code |
| Domain-based | Feature of third party |

For example, the feature number 28 is Google_Index, which examines whether a website is in Google's index or not.

$$\text{Rule: IF}\begin{cases}\text{Webpage Indexed by Google} \rightarrow \text{Legitimate}\\ \text{Otherwise} \rightarrow \text{Phishing}\end{cases}$$

### 4.2 Feature Selection

A subset of features that work well together is selected. The selection process aims to minimize the time needed to build the machine learning model and produce high accuracy. Selection features' process keeps features that have low correlation to each other, but have high correlation to the label feature [28]. The rest of the highly correlated features are dropped.

Table 3. URL features.

| # | Feature name | # | Feature name |
|---|---|---|---|
| 1 | having_IP_Address | 17 | Submit_to_email |
| 2 | URL_Length | 18 | Abnormal_URL |
| 3 | Shortining_Service | 19 | Redirect |
| 4 | having_At_Symbol | 20 | on_mouseover |
| 5 | double_slash | 21 | RightClick |
| 6 | Prefix_Suffix | 22 | popUpWidnow |
| 7 | having_Sub_Domain | 23 | Iframe |
| 8 | SSLfinal_State | 24 | age_of_domain |
| 9 | Domain_registeration | 25 | DNSRecord |
| 10 | Favicon | 26 | web_traffic |
| 11 | port | 27 | Page_Rank |
| 12 | HTTPS_token | 28 | Google_Index |
| 13 | Request_URL | 29 | Links_pointing |
| 14 | URL_of_Anchor | 30 | Statistical_report |
| 15 | Links_in_tags | 31 | Result |
| 16 | SFH | | |

### 4.3 Adaptive Boosting

AdaBoosting is the decision tree on binary classification problems. AdaBoosting is usually used for a discrete dataset, so it's more related to classification than to regression. The AdaBoosting algorithm updates the weight to minimize error, which leads to minimize the misclassification rate. It is necessary to highlight that Freund, Schapire and Abe 0 developed the AdaBoost algorithm to increase the efficiency of binary classifiers. AdaBoost uses an ensemble learning method approach to learn from weak classifiers' mistakes and turn them into strong ones. AdaBoost generates a weak learner through primary training data. The data is then adjusted according to the foreseen performance for the next round

of weak learner training. It is good to note that the training samples with the lowest predicting accuracy in the preceding step are approached with more attention in the step that follows. The weak learners with different weights are finally combined to create a strong learner 0-0.

## 5. PROPOSED MODEL

Figure 1 shows the system's flow diagram to recognize the URL. The proposed system reads the URL from the dataset, then the URL is classified into multidimensional features according to the dataset components. The model's detection accuracy is improved by selecting the most correlated features and eliminating the irrelevant features. The filtered data is split into the training set and testing data. Machine learning model is applied by using an adaptive boost classifier to create the adaptive boost knowledge base. The testing dataset is used as the input for the detection model to evaluate it.

The proposed model uses Weka 3.6, Python and MATLAB. Table 4 shows the experimental parameters, such as the evaluator, the search algorithm and the batch size, the classifier, the number of iterations and the weight threshold.

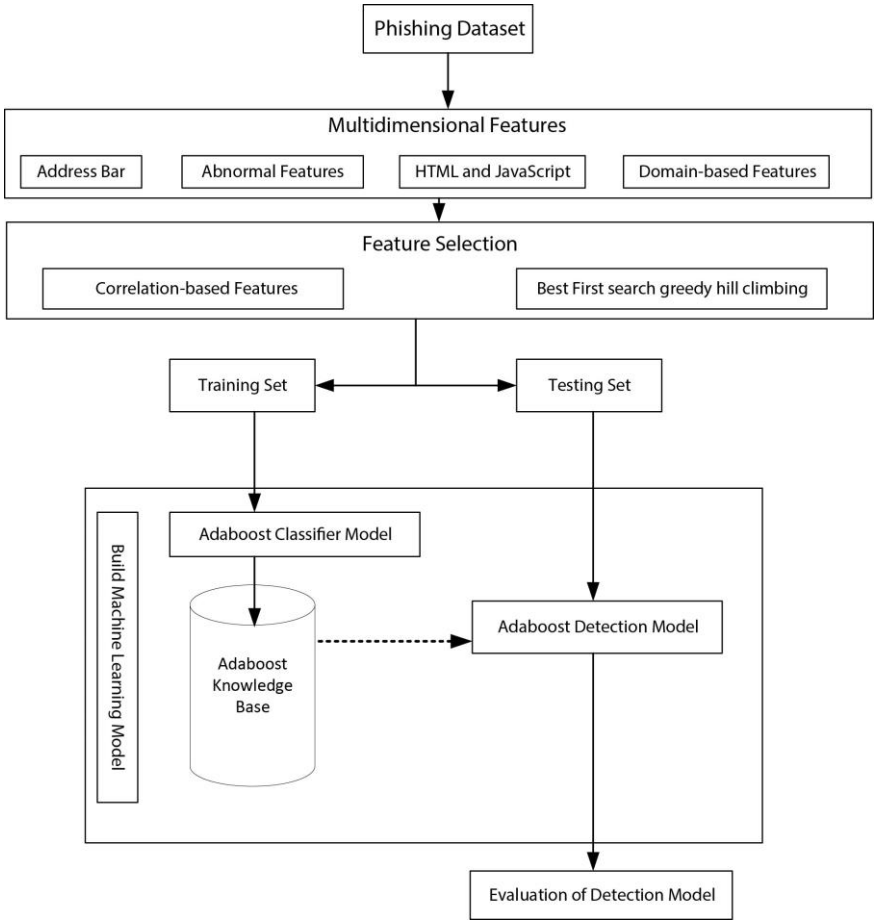


Figure 1. PhiBoost structure.

Table 4. Experimental parameters.

| **Feature Selection** | |
|---|---|
| Parameters | Value |
| Evaluator | Correlation-based Features |
| Search model | Best First search greedy hill-climbing |
| **Adaptive Boost Classifier** | |
| Parameters | Value |
| Batch size | 100 |
| Classifier | Decision Stump |
| Number of iterations | 10 |
| Weight threshold | 100 |

## 6. DISCUSSION OF RESULTS

The proposed model classifies the features into four categories by utilizing the correlation relationship between features and the class label (phishing or legitimate).

The output from the feature selection process is nine features as follows: having_IP_Address, having_Sub_Domain, SSLfinal_State, web_traffic, Google_Index, Request_URL, URL_of_Anchor, Links_in_tags and SFH. In the next feature selection phase, MATLAB built-in procedure called independent significance features test (IndFeat()) is invoked. Figure 2 shows the Python heat map of the output of the independent significance features test.

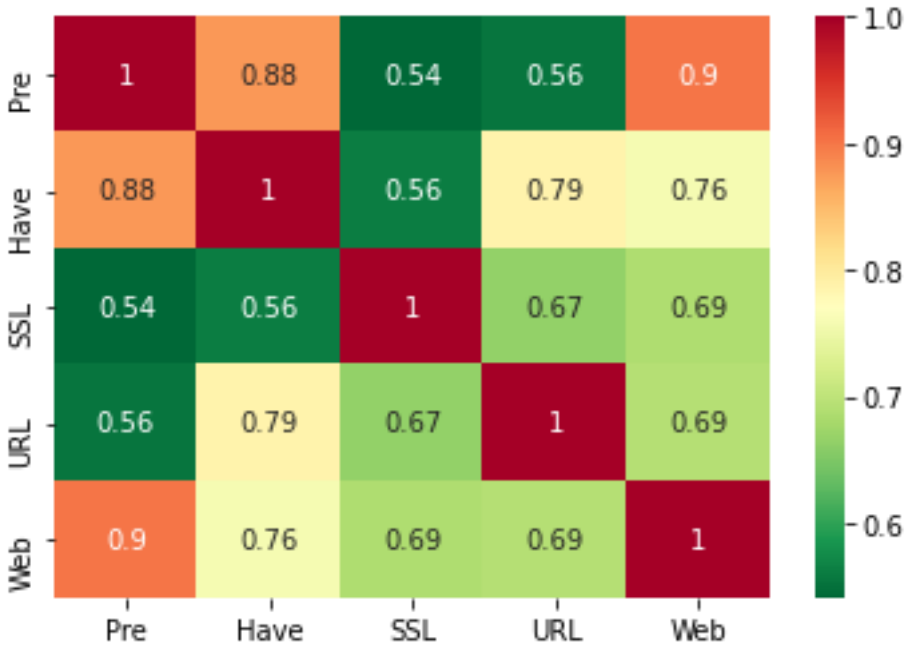


Figure 2. Heat map after applying attribute selector.

Four popular statistical measures were utilized to determine the efficiency of the proposed model. Table 5 lists these performance measures and their effects on the model performance. In our experiments, we evaluate the proposed system by using the accuracy to evaluate the ratio of correctly predicated observations to the total observations of the proposed system. Precision measure enables us to evaluate the ratio of correctly predicated observations to the total of positive observations. The recall measure evaluates the ratio of correctly predicated positive observations to all observations in the actual class. F-measure is a weighted average precision and recall.

Table 5. Popularly statistical measures.

| Statistical measures | Formula |
|---|---|
| Precision | $\frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}$ |
| Recall | $\frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}}$ |
| Accuracy | $\frac{\text{True Positive+True Negative}}{\text{Total Number of Instance}}$ |
| F-measure | $2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$ |

Table 6 shows the experiments conducted on a different percentage split. The minimum accuracy achieved in the proposed model is 97.7% and the F-measure is 97.5% after training the model in 50% of the dataset. The best performance is obtained when the training percentage is 70%, where both accuracy and F-measure are approximately 99%.

Table 6. The performance of the proposed algorithm.

| Experiment # | Training Percentage | Precision | Recall | Accuracy | F-measure |
|---|---|---|---|---|---|
| 1 | 50 % | 97.8 % | 97.1 % | 97.7 % | 97.5 % |
| 2 | 60 % | 98.2 % | 97.6 % | 98.1 % | 97.9 % |
| 3 | 70 % | 99.0 % | 98.6 % | 98.9 % | 98.8 % |
| 4 | 80 % | 98.4 % | 97.8 % | 98.3 % | 98.1 % |
| 5 | 90 % | 98.8 % | 98.2 % | 98.7 % | 98.5 % |

Figure 3 shows the efficiency of the PhiBoost model which explores the precision and accuracy with different percentages of training and testing to avoid any overfitting problem. The minimum accuracy that PhiBoost achieved was when the training test is 50% of the dataset. On the other side, the performance of the PhiBoost model improves if the training set is 70%.

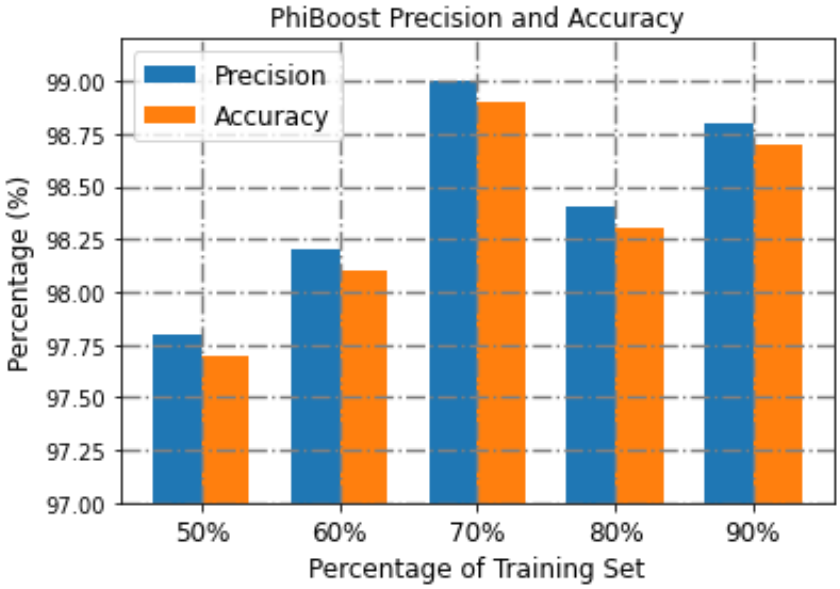


Figure 3. PhiBoost precision and accuracy.

In Table 7, the proposed model is compared with different detection machine learning models. As demonstrated in the results obtained, the proposed model enhances the accuracy of the detection system. In [27], the authors introduced a phishing detection model by utilizing feature selection and combining as a pre-processing step for the dataset. After that, they employed a multilayer perceptron neural network as a classifier function. In our proposed work, we tried to optimize the accuracy by minimizing the number of selected features and utilizing the adaptive boosting classifier.

Table 7. Comparison with the PhiBoost model.

| Paper | Machine learning algorithm | Accuracy |
|---|---|---|
| [14] | NN | 94.07% |
| [15] | multi-label rule-based | 94.8% |
| [18] | NN | 84% |
| [19] | FFNN | 92.48% |
| [21] | Feed-forward NN | 97.40% |
| [24] | Logistic regression classifier | 98.40% |
| [25] | Naïve Bayesian classifier | 90% |
| [26] | HNB and J48 | 96.25% |
| [27] | Multilayer perceptron neural network | 98.5% |
| PhiBoost model | | 98.9 % |

## 7. CONCLUSION

This paper aims to introduce an outstanding solution to the threat of phishing in our modern community. As a result, this research proposed implementing feature selection and adaptive boosting for an efficient model for detecting phishing websites. The results of this study explored the best splitting rate for the dataset to train the machine learning model, which was 70%. The results achieved a high accuracy and a high F-measure with high predictive capability as well as with low false-positive rates and low false-negative rates. The proposed model minimizes the time to build the training model by picking up the most correlated features and produces an extremely high predictive accuracy of approximately 99%. Conclusively, the application of the implemented methods of this research in a real-time environment remains pivotal in future work. In the future, the system's capability will be investigated by testing it over a real-time environment.

## REFERENCES

[1] G. Varshney, M. Misra and P. K. Atrey, "A Survey and Classification of Web Phishing Detection Schemes," Security and Communication Networks, vol. 9, pp. 6266-6284, 2016.

[2] A. Aleroud and L. Zhou, "Phishing Environments, Techniques and Countermeasures: A Survey," Computers & Security, vol. 68, pp. 160-196, 2017.

[3] C. Singh, "Phishing Website Detection Based on Machine Learning: A Survey," Proc. of the 6th IEEE International Conference on Advanced Computing and Communication Systems (ICACCS), pp. 398-404, Coimbatore, India, 2020.

[4] L. Jelovčan, S. L. Vrhovec and A. Mihelič, "A Literature Survey of Security Indicators in Web Browsers," Elektrotehniski Vestnik, vol. 87, pp. 31-38, 2020.

[5] Y. Al-Hamar, H. Kolivand and A. Al-Hamar, "Phishing Attacks in Qatar: A Literature Review of the Problems and Solutions," Proc. of the 12th IEEE International Conference on Developments in eSystems Engineering (DeSE), 2019, pp. 837-842, Kazan, Russia, 2019.

[6] M. Sánchez-Paniagua, E. Fidalgo, V. González-Castro and E. Alegre, "Impact of Current Phishing Strategies in Machine Learning Models for Phishing Detection," Proc. of the 13th International Conference on Computational Intelligence in Security for Information Systems (CISIS 2020), Part of the Advances in Intelligent Systems and Computing Book Series (AISC), vol. 1267, pp. 87-96, 2020.

[7] A. S. Onashoga, O. E. Ojo and O. O. Soyombo, "Securix: A 3D Game-based Learning Approach for Phishing Attack Awareness," Journal of Cyber Security Technology, vol. 3, pp. 108-124, 2019.

[8] K. Hynek, T. Čejka, M. Žádník and H. Kubátová, "Evaluating Bad Hosts Using Adaptive Blacklist Filter," Proc. of the 9th IEEE Mediterranean Conf. on Emb. Comp. (MECO), pp. 1-5, Budva, Montenegro, 2020.

[9] S. Sarika, "A Heuristic Model to Detect Malicious URLs Using Case-based Reasoning," Journal of Information and Computational Science, vol. 9, no. 11, pp. 1066–1079, 2019.

[10] S. Abdelnabi, K. Krombholz and M. Fritz, "VisualPhishNet: Zero-Day Phishing Website Detection by Visual Similarity," Proc. of the ACM SIGSAC Conference on Computer and Communications Security (CCS '20), pp. 1681–1698, [Online], Available: https://doi.org/10.1145/3372297.3417233, Oct. 2020.

[11] B. B. Gupta and A. K. Jain, "Phishing Attack Detection Using a Search Engine and Heuristics-based Technique," Journal of Information Technology Research (JITR), vol. 13, pp. 94-109, 2020.

[12] E. Anthi, L. Williams, M. Słowińska, G. Theodorakopoulos and P. Burnap, "A Supervised Intrusion Detection System for Smart Home IoT Devices," IEEE Internet of Things J., vol. 6, pp. 9042-9053, 2019.

[13] 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," Sensors, vol. 19, p. 4258, 2019.

[14] P. Yi, Y. Guan, F. Zou, Y. Yao, W. Wang and T. Zhu, "Web Phishing Detection Using a Deep Learning Framework," Wireless Communications and Mobile Computing, vol. 2018, [Online], available: https://doi.org/10.1155/2018/4678746, 2018.

[15] T. Alves, R. Das and T. Morris, "Embedding Encryption and Machine Learning Intrusion Prevention Systems on Programmable Logic Controllers," IEEE Embedded Sys. Letters, vol. 10, pp. 99-102, 2018.

[16] P. Prakash, M. Kumar, R. R. Kompella and M. Gupta, "PhishNet: Predictive Blacklisting to Detect Phishing Attacks," Proceedings of IEEE INFOCOM, pp. 1-5, San Diego, USA, 2010.

[17] S. Marchal, J. François, R. State and T. Engel, "PhishStorm: Detecting Phishing with Streaming Analytics," IEEE Transactions on Network and Service Management, vol. 11, pp. 458-471, 2014.

[18] A. Subasi, E. Molah, F. Almkallawi and T. J. Chaudhery, "Intelligent Phishing Website Detection Using Random Forest Classifier," Proc. of the IEEE International Conference on Electrical and Computing Technologies and Applications (ICECTA), pp. 1-5, Ras Al Khaimah, United Arab Emirates, 2017.

[19] S. Smadi, N. Aslam and L. Zhang, "Detection of Online Phishing Email Using Dynamic Evolving Neural Network Based on Reinforcement Learning," Decision Support Systems, vol. 107, pp. 88-102, 2018.

[20] N. Abdelhamid, F. Thabtah and H. Abdel-jaber, "Phishing Detection: A Recent Intelligent Machine Learning Comparison Based on Models Content and Features," Proc. of the IEEE International Conference on Intelligence and Security Informatics (ISI), pp. 72-77, Beijing, China, 2017.

[21] K. L. Chiew, C. L. Tan, K. Wong, K. S. Yong and W. K. Tiong, "A New Hybrid Ensemble Feature Selection Framework for Machine Learning-based Phishing Detection System," Information Sciences, vol. 484, pp. 153-166, 2019.

[22] P. PhishTank, "Join the Fight against Phishing," [Online], Available: http://phishtank.org, 2016.

[23] R. K. V. Penmatsa and P. Kakarlapudi, "Web Phishing Detection: Feature Selection Using Rough Sets and Ant Colony Optimization," International Journal of Intelligent Systems Design and Computing, vol. 2, pp. 102-113, 2018.

[24] O. S. Qasim and Z. Y. Algamal, "Feature Selection Using Particle Swarm Optimization-based Logistic Regression Model," Chemometrics and Intelligent Laboratory Systems, vol. 182, pp. 41-46, 2018.

[25] N. A. Azeez and A. Oluwatosin, "CyberProtector: Identifying Compromised URLs in Electronic Mails with Bayesian Classification," Proc. of the IEEE International Conference on Computational Science and Computational Intelligence (CSCI), pp. 959-965, Las Vegas, USA, 2016.

[26] S. Zaman, S. M. U. Deep, Z. Kawsar, M. Ashaduzzaman and A. I. Pritom, "Phishing Website Detection Using Effective Classifiers and Feature Selection Techniques," Proc. of International Conf. on Innovation in Engineering and Technology (ICIET), vol. 23, p. 24, DOI: 10.13140/RG.2.2.24043.08483, 2019.

[27] A. Odeh, I. Keshta and E. Abdelfattah. "Efficient Detection of Phishing Websites Using Multilayer Perceptron," International J. of Interactive Mobile Technologies (iJIM), vol. 14, no. 11, pp. 22- 31, 2020.

[28] Y. Freund, R. Schapire and N. Abe, "A Short Introduction to Boosting," Journal-Japanese Society for Artificial Intelligence, vol. 14, no. 5, pp. 771-780, 1999.

[29] D. C. Feng, Z. T. Liu, X. D. Wang, Y. Chen, J. Q. Chang, D. F. Wei and Z. M. Jiang, "Machine Learning-based Compressive Strength Prediction for Concrete: An Adaptive Boosting Approach," Construction and Building Materials, vol. 230, ID no. 117000, [Online], Available: https://doi.org/10.1016/j.conbuildmat.2019.117000, 2020.

[30] S. Abdulhamit and E. Kremic. "Comparison of Adaboost with MultiBoosting for Phishing Website Detection," Procedia-Computer Science, vol. 168, pp. 272-278, 2020.

[31] V. Shahrivari, M. M. Darabi and M. Izadi, "Phishing Detection Using Machine Learning Techniques," arXiv preprint arXiv:2009.11116, [Online], Available: https://arxiv.org/pdf/2009.11116.pdf, Sep. 2020.

[32] V. Ramanathan and H. Wechsler, "Phishing Website Detection Using Latent Dirichlet Allocation and AdaBoost," Proc. of the IEEE International Conference on Intelligence and Security Informatics, pp. 102-107, Arlington, USA, 2012.

[33] B. Alotaibi and M. Alotaibi, "Consensus and Majority Vote Feature Selection Methods and A Detection Technique for Web Phishing," Journal of Ambient Intelligence and Humanized Computing, [Online], Available: https://doi.org/10.1007/s12652-020-02054-3, 2020.

[34] M. Zabihimayvan and D. Doran, "Fuzzy Rough Set Feature Selection to Enhance Phishing Attack Detection," Proc. of the IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), pp. 1-6, DOI: 10.1109/FUZZ-IEEE.2019.8858884, June 2019.

[35] Y. A. Alsariera, A. V. Elijah and A. O. Balogun, "Phishing Website Detection: Forest by Penalizing Attributes Algorithm and Its Enhanced Variations," Arabian Journal for Science and Engineering, vol. 45, pp. 10459–10470, 2020.

**ملخص البحث:**

تزداد كل يومٍ الهجمات السيبرانية التي تستخدم استراتيجياتٍ مختلفة. ومن أكثر الهجمات السيبرانية شيوعاً ما يعرف بـ "التلصُّص" على البيانات؛ إذ يقوم المهاجم بجمع معلوماتٍ حساسةٍ وسريةٍ بينما يحاول إظهار نفسه كطرفٍ موثوقٍ به. وقد ابتكرت استراتيجياتٌ مختلفة لمقاومة هذه الظاهرة، مثل: الوضع على القائمة السوداء، والبحث الموجه لكشف الهجمات، والتشابه المرئي. ولكن هذه الطرق التقليدية لها في الغالب معدلات خطأ عالية وتستغرق الكثير من الوقت لكشف الموقع الإلكتروني المهاجم. وقد استخدمت نماذج جديدة تستفيد من تقنيات تعلم الآلة التي من شأنها أن تحسن من دقة الكشف.

وتحتاج تقنيات تعلم الآلة الى كمياتٍ ضخمةٍ من البيانات تسمى "السِّمات"، ويتم جمعها من مواقع الكترونية مختلفة. وتصنف هذه السِّمات التي يجري جمعها ضمن أربعة أصناف او فئات.

تقدم هذه الورقة نموذج كشفٍ مبتكراً يستند على الاستفادة من انتقاء السِّمات لالتقاط السِّمات ذات الارتباط العالي بعلامة الصنف. وتوظف مرحلة انتقاء السِّمات مكتبة السِّمات ذات الأهمية المستقلة من ماتلاب (MATLAB) والخريطة الحرارية من بايثون (Python) لإيجاد السِّمات ذات الارتباط العالي. بعدئذٍ، يستخدم النموذج المقترح طريقة تعزيزٍ تكيّفيةً تشتمل على عدّة مُصنِّفات لزيادة دقة النموذج. ويحقق النموذج المقترح دقةً تنبُّؤيّةً عاليةً جداً تصل الى ما يقرب من 99%.