URL Classification Based on Lexical Features by Machine Learning

1st Cing Gel Vung
University of Computer Studies, Mandalay, Myanmar
cinggelvung@ucsm.edu.mm

2nd Yu Yu Win
University of Computer Studies, Mandalay, Myanmar
yuyuwin2@ucsm.edu.mm

Abstract*—The malicious website becomes the hub sector in the cybercrime component of the internet. Attackers delivered malicious URLs to target users via links, emails, or advertisements. Many of the previous research has analyzed URL phishing detection with several approaches to reduce the risk. In this work, we have investigated the lexical structure of the URL as input for the classification models. The system has employed the Extreme Gradient Boosting (XGBoost), Support Vector Machine (SVM), and Artificial Neural Network (ANN) as evaluators for detecting malicious URLs. The datasets are collected from the Phish Tank website to build the proposed system. The approach has adopted static lexical features with imbalanced dataset for safer and faster extraction. Evaluation of the classifiers achieved the accuracy of 88%, 87%, and 88% respectively. The detection rate is high, a false positive rate is 0.13%, and false negative rate is 0.07% in XGBoost. The results show that the imbalanced nature of phishing URL affects the detection system performance.

Keywords—cybersecurity, feature extraction, machine learning, classification

I. INTRODUCTION

The internet becomes a bridge for people to communicate and share information around the global world. Almost of the important information are stored online using different kind of services such as cloud storage, email. The fraudsters attempt to steal confidential information for illegal economic benefits, such as user-names, passwords, and credit card details using advanced hacking technology. Therefore, phishing is one of the major challenges in cybersecurity since it can cause damage to the organization. The attackers disseminate phishing emails with camouflaged contents and links where text characters look similar to real text. In the last few years, a large number of studies have been dedicated to network security to make sure that transmitted and stored data is safe and secure.

URLs that are used for compromising the security of the system or the organization in cyber-attacks are termed as malicious URLs. Many approaches have developed to tackle the problem of malicious URL detection. These approaches can be categorized into (i) Blacklisting or Heuristics, and (ii) Machine Learning approaches [15]. Feature engineering is a process of machine learning algorithms that select the most relevant and remove redundant features [6].

There is much research aimed to develop an application that can correctly predict malicious URLs based on website detection or e-mails [4, 9, and 10]. The author focused on phishing e-mails using the enhanced techniques XGBoost algorithm with feature extraction method Latent Dirichlet Allocation (LDA) to increase the accuracy and precision of prediction. Vector Space Models (VSM), a type of feature engineering represents each message as symbols in vector space using numerical values. This model solved three problems: the curse of dimensionality, the sparsity, and the context portion represented together in the VSM [12].

Effective systems to detect malicious URLs on time can greatly help to counter a large number of cyber-security threats. The problem of selection feature sets for phishing cases plays an important role in classifications. Some studies have implemented the optimization algorithm such as gravitational search and swarm optimization-based for evaluating feature significance [1, 7, 16, and 17].

In malicious URL detection, machine learning has time-consuming and challenges in feature engineering. However, it is still a hot topic in the research area**.** In this work, we focus on machine learning for the classification of websites using lexical and statistical analysis of URLs. This paper has proposed an Artificial Neural Network (ANN), Support Vector Machine (SVM), and XGBoost to classify the phishing URL. We also manipulated the feature weighting to distinguish significant features. We have established our model on an imbalanced and labeled dataset of legitimate and malicious URLs.

The remainder of this study is organized as follows: Section II represents the related work of the phishing detection using different methods. Section III describes the process of the proposed system. The experimental results are shown in section IV and Section V consists of Conclusion and Future Work.

II. RELATED WORK

This section discusses some of the previous works that are useful for enhancing the detection of phishing URLs based on machine learning, including traditional approaches and deep learning techniques. Resource on the Internet is addressed by the Uniform Resource Locator (URL), which consists of two parts, Hostname and FreeURL. Consider a phishing URL, https://netflix-rebillings.com/finish.php?bank=citi as an example. The structure is as follow:

- Protocol: https
- Hostname: netflix-rebillings.com
- Path (location): finish.php
- Parameter and value: bank, citi

To confuse users, the attackers forge the URL of target website to produce the phishing URL [5].

Sadique, Farhan, et al [18] used four sets of features: lexical, host, GeoIP, and domain WHOIS collected from Phishtank.com. They formulated an automated framework for real-time phishing using an online learning classifier and analyzed the feature importance. However, the system extracted many URL features so the cost of collecting features decreased the performance.

This paper [13] manipulated only the lexical 18 features from the URL string. This study proved that Random Forest is better than the 1DConv-LSTM approach. The authors adopted under/over-sampling methods for the class imbalance datasets. Yang, Zho [5] applied deep learning for quick classification websites with multidimensional features. The authors proposed a dynamic category decision algorithm (DCDA) to reduce the detection time and cost comparing with the LSTM approach. Although the system performance is improved, the threshold value cannot be useful for real-time datasets.

Classification of malicious URL using the Random Forest model approved a significant increase in detection only with lexical features. The authors combined the features with trigram-based features using the NLP python package to find highly correlated features [14]. Identifying threats and anomaly occurring in smart IoT devices using Artificial Neural Network outperformed Logistic Regression with multi-class [3].

Gravitational Search Algorithm (GSA) optimized the result of Artificial Neural Network to recognize the new attack pattern in this paper [7]. Using multi-class classification with lexical features exploited the feature selection, redundancy, and correlation for giving higher accuracy [19]. These hybrid approaches support the detection system in which features are significant and contribute most to the analysis.

III. SYSTEM ARCHITECTURE

The proposed system consists of three main processes. First, the system extracted a series of the feature from the URLs using the defined rule-based. Static lexical features reduced the amount of time required to extract the useful features for the system. Second, the extracted features are normalized by min-max normalization to feed the features into the model. Finally, the system used the features as the training sample. The system then built the models with three methods that receives a URL as a character sequence, and predicted whether the URL corresponds to the phishing case.

XGBoost feature weighting is utilized to rank the weights value of features. It can also diminish the number of features that are redundancy and irrelevant for the classification. The organization of the system flow is shown in Figure 1.

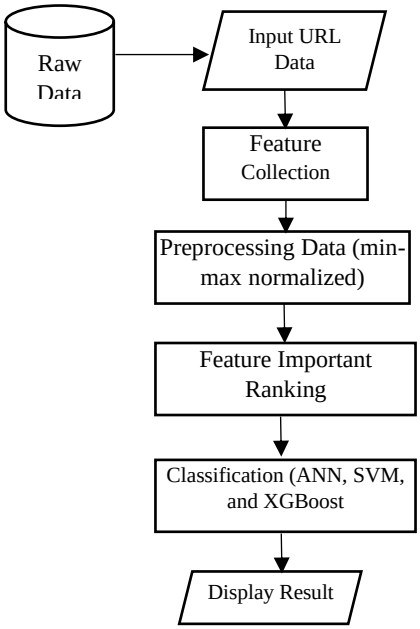

Figure. 1. System Flow Diagram of the System

A. Problem Definition

We can describe the problem of the detection system as a binary classification in which the system is to classify the given URLs either as legitimate or malicious. Let us consider a set of datasets URLs in the forms of $\{(u_1, y_1), (u_2, y_2) \ldots (u_n, y_n)\}$ where:

- u_i for i=1, 2… N denotes given URLs in the dataset

- $y_i \in \{0, 1\}$ for i=1, 2… N indicates the class label of the corresponding URLs where y=0 implies legitimate and y=1 indicates a malicious URL respectively.

B. Feature Representation

The quality of feature representation critically affects the machine learning models to classify correctly the URLs. The feature representation converts the given URLs string into a d-dimensional feature vector that can feed into the machine learning model as an input. There are two steps in the process of feature representation:

- Feature collection: collects the relevant information about the URLs in which features are obtained from the URL string, information about the host, etc.
- Feature preprocessing: The unstructured textual features need to convert appropriately into the numerical feature vector because the machine learning techniques can only accept the numeric data to process.

Based on the above research, this paper has chosen the lexical feature for fast feature extraction. Sixteen kinds of features, extracted based on the structured of the URL. The information entropy refers to the uncertainty of URL characters. The length of the URL is usually longer than 54 in malicious URLs. The phishers intimate the phishing URL by adding special symbols '#', '&', '@'and '_' in the legitimate URL. Phishing URL usually shows higher occurrences of digits than legitimate URL. If the number of dot in the hostname is more than three, the URL tends to

be a malicious one. If the URL is an IP instead of a domain name, it is a feature of the phishing URL. A legitimate URL keeps a balanced ratio between the digits and characters.

The malicious URL directs to the other domain or links in the URL string, so the length is longer than others are. Fake URLs are no longer live, whereas the legitimate URL have indexed in Google. The attackers used the similar top-level domain name to create the malicious URL. The system also considered the extension name in the path of URL because some hackers add malwares inside the files, pictures and websites. TABLE I describes the information of features of the URLs string used in this system.

TABLE I. URLS FEATURES

URL lexical features description	Feature Names
Information Entropy	Entropy
URL length	urllength
HTTPS protocol	hasHttps
HTTP protocol	hasHttp
Containing the "@" symbol	Having_@_symbol
Containing the "_"	Seperation
Numbers of digits	numDigits
Numbers of top level domain in path	Number_Subdomain
Number of "#"	NumParameter
Number of "&"	numParams
File Extension in the path	FileExtension
The ratio of digits to characters	RatioDigit_Char
Google Index	Google_Index
IP address	IsIP
Top level domain in host name	TopLevelDomain
The URL has redirection ('//')	Redirection

C. Classification Algorithms

In this paper, we have deployed three classification algorithms:

- Extreme Gradient Boosting (XGBoost): it is a supervised learning algorithm and an extension to gradient boosted decision tree (GBM) specially designed to improve speed and performance. "XGBoost" performs well in machine learning classification, regression, and ranking problems. It found the optimal solution using regularization, and stochastic gradient boosting [12], [5].
- Artificial Neural Networks (ANN): a neural network is a bio-inspired machine learning model that behaves like a neuron in the human brain, and connected as a set of artificial neurons. ANN is a fully connected neural network as shown in Figure 2. It consists of an input layer, two hidden layers, and an output layer. ReLU activation function is used in the hidden layers, and the ANN applied sigmoid in the output layers.

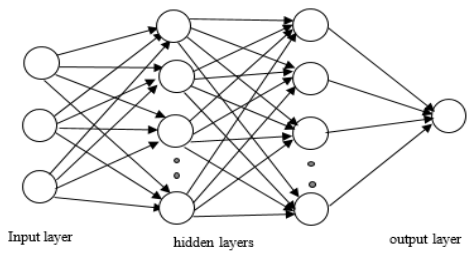

Figure. 2. Artificial Neural Network

- Support Vector Machine (SVM): finds a hyperplane in the feature variables to divide the data points into two classes. The most significant margin that exits in the maximum distance between the data points of two groups is the best hyperplanes. Training points that lie on one of the hyperplanes, called support vectors and are essential to the classification and accuracy [12].

IV. EXPERIMENTAL RESULTS AND DISCUSSION

The system has implemented using the python language. XGBoost and SVM classifier used the Scikit-learn package to get feature importance. We installed TensorFlow for implementing the neural network learning classifier.

A. Dataset and Data source

The phishing datasets are real-life data collected from the Phish Tank website, and the legitimate URLs dataset driven from this paper [19]. The dataset consists of 16,036 records from out of which 11,033 are malicious datasets, and 5,004 are benign respectively. Moreover, it is an imbalanced dataset. Most of the URL detection system had made research more based on legitimate URL. This paper had studied malicious URL more than the legitimate URL to discover the pattern and characteristic of URL representation.

The unstructured dataset (URLs) are crawled URLs the Phish Tank website, and it is shown in TABLE II.

TABLE II. UNSTRUCTURED DATASET

	url	Veri-fied
0	http://cheezburger.com/8491583232/funny-sign-pic-kids-smoking?ref=leftarrow/	no
1	http://codecanyon.net/item/accordion-for-layers-wordpress-theme/screen_preview/	no
2	http://codecanyon.net/item/flowflow-social-streams-for-wordpress/9319434	no
3	http://codecanyon.net/item/hide-my-joomla-hide-your-source-links/8988449	no
4	http://codecanyon.net/item/imgrid-media-grid-responsive-gallery/11227113	no

In the feature collection step, the system extracted features from the given URLs string, and transmitted them to the numeric values through the predefined URL-based. If the features are present, we assigned a binary value 1. Otherwise, we set the binary value 0 shown in TABLE III.

TABLE III. FEATURE EXTRACTED DATA

entropy	Separation	TopLevelDomain	urllength
4.3748	1	ph	1
4.2074	0	com	0
4.2713	1	cyou	0
4.2981	0	tk	0
4.3037	1	com	0

The URL string is very unstructured and noisy to build an efficient classification system. The extracted feature is critical for system performance. In the preprocessing step, the system utilized the min_max normalization method to transform some of the features into a standard format (entropy, number of digits, number of parameters and the length of URL) shown in TABLE IV.

TABLE IV. STRUCTURED DATA

entropy	numDigits	numParams	urllength
1	1	0	1
1	0	0	0
1	0	0	0
1	1	0	0

B. *Modeling and Evaluation Phishing URL*

For unbiased classification, we have split the dataset, from which 70% of the records are used for training while the remaining 30% instances are being for testing. The model adopted some of the performance metrics for evaluating the correctness. The brief description is as follows:

- Confusion Matrix: it contains all of the system information about the real and estimated results. The model performance can achieve by calculating the result matrix. We have considered being noted malicious URL as the positive class and benign URL as the negative.
- TP (True Positive) and TN (True Negative) refer to the correctly classified results whereas FP (False Positive) and FN (False Negative) denote the misclassified results.
- Accuracy: it is the ratio of correctly classified observations to the total number of records.

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

- Precision: it shows good performance when the cost of false positive is high.

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

- Recall: it holds a good performance metric when the cost of false negative is high.

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

- F1-score: it is the harmonic mean of precision and recall.

$$F1 - score = \frac{2 * Precision * Recall}{Precision + Recall}$$

The system define the malicious URL as positive and the benign URL as negative. Table V represents the confusion matrix of the three classifiers based on test datasets.

TABLE V. CONFUSION MATRICES

	Predicted		Malicious	Benign
ANN	Actual	Malicious	2821	440
		Benign	126	1425
SVM	Actual	Malicious	2784	477
		Benign	111	1440
XGBoost	Actual	Malicious	2822	439
		Benign	121	1430

Table VI shows the performance comparison of the algorithms. Among the three algorithms, the XGBoost algorithm outperforms the other two methods in Accuracy, Recall, and F1.

TABLE VI. EXPERIMENTAL RESULTS OF MODELS ON TEST DATA

Methods	Accuracy		Precision	Recall	F1
ANN	88.24		95.72	86.51	90.88
SVM	87.781		96.1658	85.373	90.448
XGBoost	88.38		95.65	86.75	90.99

Figure 3 plots the performance measures of the three methods. These measures are achieved high marks in precision (95.65%, 95.72% and 96.17% accordingly) through the evaluation of the algorithms.

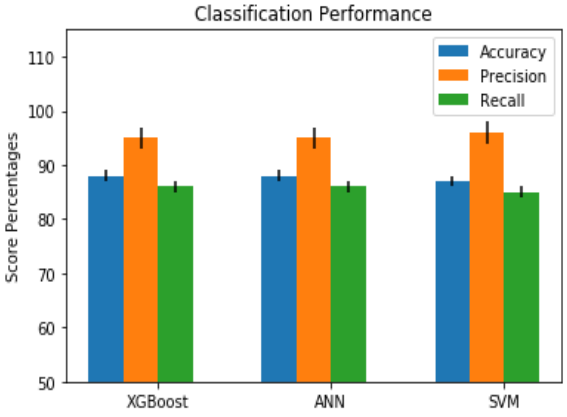

Figure. 3. The performance comparisons of the models

For class-imbalanced data problems, precision and recall are more accurate to analyze the false positive rate and the true-positive rate. Based on the Table V, the false positive rate of ANN and XGBoost is 0.13%, and 0.15% in SVM. The negative rate of ANN is 0.08%, and both of the other two methods are 0.07%. The system also evaluated the AUC of the models; illustrated in Figure 4.

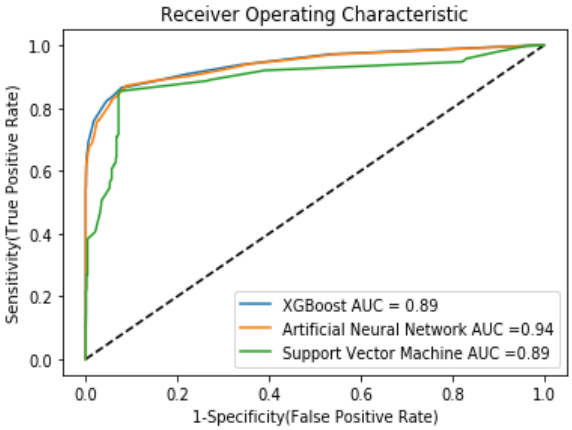

Figure. 4. The ROC curve of XGBoost, ANN, and SVM

Misclassifying a malicious URL as legitimate may bring a security risk to the website. Moreover, a legitimate URL misjudged as a phishing URL may instill inconvenience for the trust issues to the operators of the website. Although the three models achieved the good results, the amount of legitimate URL misclassified as the phishing URL in XGBoost is lower than the amount of misclassified URL in SVM. In addition, the number of malicious URL misclassified as the legitimate URL in XGBoost is less than the numbers in ANN.

The loss and accuracy curves of ANN is shown in Figure 5. ANN model adjust the hyperparameters by training on the train set and select the hyperparameters that estimate the best accuracy on the test set. We have experienced the classifier with epoch value 100, two hidden layers of nodes 50 and 25 accordingly. However, the accuracy is stable after epoch 50. To reduce training time and to improve detection speed, we validated the model with epoch values 50, neuron nodes 50, and 25 for

hidden layers. The system used the Adam optimizer with the learning rate (0.01) to enhance the parameters. Sigmoid activation function is used for binary classification.

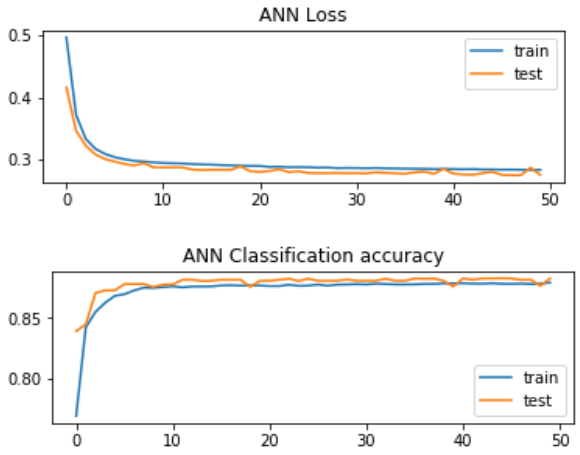

Figure. 5. ANN average loss and accuracy curve

XGBoost classifier practiced the training set with all features, and it attained valuable results. Figure 6 presents the loss and classification error achieved by XGBoost. To get the optimized model, needed to tune the important parameters. "n_estimators" is the number of runs XGBoost will try to learn. "max_depth" parameter represents the depth of each tree, which is the maximum number of different features used in each tree. 'objective= binary: logistic' specify a binary classification task with objective function using probability.

The classification error rate shows a lower error rate around iteration 270. XGBoost log loss stabilize with 300 iterations. We also tested with the number of estimator 1000 and 10000. However, the loss does not decrease, and it is stable at 0.285. The classification error stops in 0.1167 after iterations 300. The model obtained the highest results with the following parameters to save the time for parameter tuning and to avoid fitting (n_estimators = 300, max_depth = 6, objective= binary: logistic).

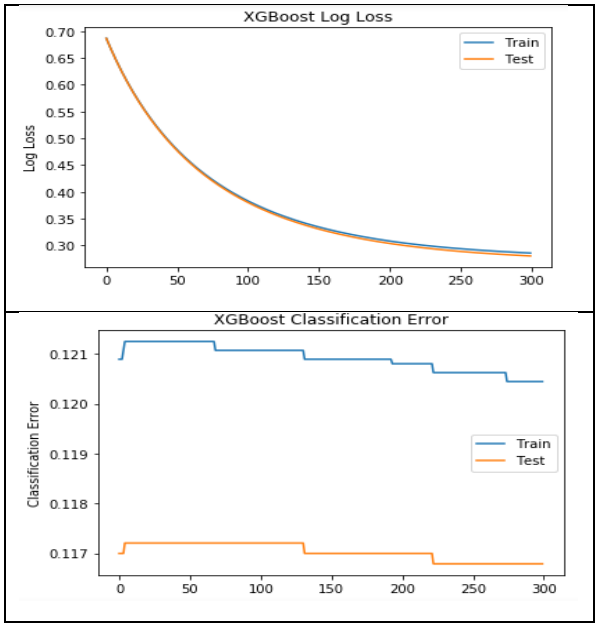

Figure. 6. XGBoost loss and error curve

In SVM model, to find the maximum-margin hyperplane, the parameter C is 100. The value C means the number of misclassified points. The kernel is linear which compute the similarity in the input space.

The training time of ANN has taken more time than XGBoost and SVM. The performance of XGBoost had improved significantly rather than the other two models in accuracy and false positive rate. This model can perform better than [2] that deployed the hybrid algorithm in the classification of URL phishing or not. The accuracy of this approach is higher with the percentages of 14 and 12 than the mark hit in [11] used significance feature selection respectively.

C. Feature Importance

Feature importance assigns scores to input features that indicate how of each feature was in the construction of the boosted decision trees within the model [8]. By calculating the feature importance, we can better understand the data and model [13]. The following steps are included for computing importance of each feature:

- An initial prediction of 1 for each sample data.
- Pass the initial forest for scoring the probability value (P) using the sigmoid function.
- Compute the negative gradient with the equation $G= P-Y_i$
- Calculate the Hessian Value, $H= P(1-P)$
- Construct the tree using the greedy method and get the weight value for each leaf node. The weight is the number of times a feature appears in a tree
- The process repeated until there is no node to split.

Figure 7 illustrated the ranking of lexical features that are important for classification.

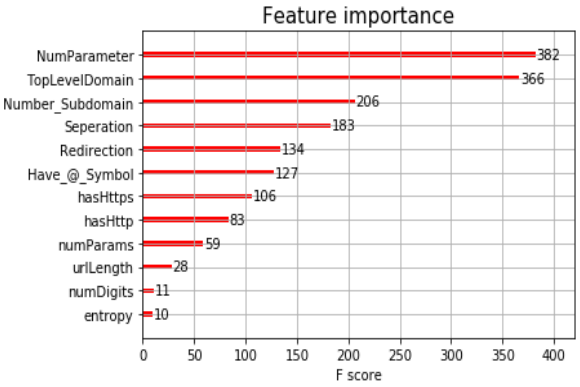

Figure. 7. The ranking of weighted feature importance

The lexical features (ratio of the digit to char, Is IP address, file extension, Google Index) have zero weighted values. The number of parameters ('#') and top-level domain kept the highest in phishing URL detection. It indicates that the attackers try to deceive users by using the trusted domain name. The presence of the '&' parameter and the number of the dash in the path are two critical features in identifying malicious URLs as well.

V. CONCLUSION AND FUTURE WORK

Given this scenario, this paper calculated the weighted feature that is important for the classification decision. In

this paper, we have applied the three-classifier model to classify whether the URL is benign or malicious. The accuracy of XGBoost was higher the ANN as 0.14% as and more 0.6% than SVM. XGBoost is the best model to achieve optimized precision, recall, accuracy, and F1 score. The results display that the statistic lexical feature can classify efficiently to generate fast detection system for URL or website on the Internet. Feature engineering is essential in machine learning. More features can enhance system performance in XGBoost.

We can focus on the deep learning approach to improve the accuracy in future works. The memory requirement and the models' training time can be compared by adding other web content features, host features, and whois (domain) information with more datasets as the challenges to reduce the false-positive rate.

REFERENCES

[1] Bardamova, Marina, Ilya Hodashinsky, Anton Konev, and Alexander Shelupanov. "Application of the Gravitational Search Algorithm for Constructing Fuzzy Classifiers of Imbalanced Data." *Symmetry* 11, no. 12 (2019): 1458.

[2] Ms. Sophiya Shikalgar , Dr. S. D. Sawarkar , Mrs. Swati Narwane, Detection of URL based Phishing Attacks using Machine Learning, International Journal Of Engineering Research & Technology (Ijert) Volume 08, Issue 11 (November 2019).

[3] Sahu, Nilesh Kumar, and Indrajit Mukherjee. "Machine Learning based anomaly detection for IoT Network: (Anomaly detection in IoT Network)." In 2020 4th International Conference on Trends in Electronics and Informatics (ICOEI) (48184), pp. 787-794. IEEE, 2020.

[4] Le Page, Sophie, and Guy-Vincent Jourdan. "Victim or Attacker? A Multi-dataset Domain Classification of Phishing Attacks." In *2019 17th International Conference on Privacy, Security and Trust (PST)*, pp. 1-10. IEEE, 2019.

[5] Yang, Peng, Guangzhen Zhao, and Peng Zeng. "Phishing website detection based on multidimensional features driven by deep learning." *IEEE Access* 7 (2019): 15196-15209.

[6] Li, Tie, Gang Kou, and Yi Peng. "Improving malicious URLs detection via feature engineering: Linear and nonlinear space transformation methods." Information Systems 91 (2020): 101494.

[7] Dastanpour, Amin, Suhaimi Ibrahim, Reza Mashinchi, and Ali Selamat. "Using gravitational search algorithm to support artificial neural network in intrusion detection system." *SmartCR* 4, no. 6 (2014): 426-434.

[8] Khan, Hafiz Mohammd Junaid, Quamar Niyaz, Vijay K. Devabhaktuni, Site Guo, and Umair Shaikh. "Identifying Generic Features for Malicious URL Detection System." In *2019 IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, pp. 0347-0352.

[9] Patil, D., and J. Patil. "Feature-based malicious URL and attack type detection using multi-class classification." *ISeCure-The ISC International Journal of Information Security* 10, no. 2 (2018): 141-162.

[10] Chatterjee, Moitrayee, and Akbar-Siami Namin. "Detecting phishing websites through deep reinforcement learning." In *2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC)*, vol. 2, pp. 227-232. IEEE, 2019.

[11] Goswami, D. N., Manali Shukla, and Anshu Chaturvedi. "Phishing Detection Using Significant Feature Selection." In *2020 IEEE 9th International Conference on Communication Systems and Network Technologies (CSNT)*, pp. 302-306. IEEE, 2020.

[12] Gualberto, Eder S., Rafael T. De Sousa, P. De B. Thiago, João Paulo CL Da Costa, and Cláudio G. Duque. "From Feature Engineering and Topics Models to Enhanced Prediction Rates in Phishing Detection." *IEEE Access* 8 (2020): 76368-76385.

[13] Hong, Jiwon, Taeri Kim, Jing Liu, Noseong Park, and Sang-Wook Kim. "Phishing url detection with lexical features and blacklisted domains." In *Adaptive Autonomous Secure Cyber Systems*, pp. 253-267. Springer, Cham, 2020.

[14] Joshi, Apoorva, Levi Lloyd, Paul Westin, and Srini Seethapathy. "Using Lexical Features for Malicious URL Detection--A Machine Learning Approach." *arXiv preprint arXiv: 1910.06277* (2019).

[15] Sahoo, Doyen, Chenghao Liu, and Steven CH Hoi. "Malicious URL detection using machine learning: A survey." *arXiv preprint arXiv: 1701.07179* (2017).

[16] Ali, Waleed, and Sharaf Malebary. "Particle swarm optimization-based feature weighting for improving intelligent phishing website detection." *IEEE Access* 8 (2020): 116766-116780.

[17] Wang, Jie-Sheng, and Jiang-Di Song. "Function optimization and parameter performance analysis based on gravitation search algorithm." *Algorithms* 9, no. 1 (2016).

[18] Sadique, Farhan, Raghav Kaul, Shahriar Badsha, and Shamik Sengupta. "An Automated Framework for Real-time Phishing URL Detection." In *2020 10th Annual Computing and Communication Workshop and Conference (CCWC)*, pp. 0335-0341. IEEE, 2020.

[19] Mamun, Mohammad Saiful Islam, Mohammad Ahmad Rathore, Arash Habibi Lashkari, Natalia Stakhanova, and Ali A. Ghorbani. "Detecting malicious urls using lexical analysis." In *International Conference on Network and System Security*, pp. 467-482. Springer, Cham, 2016.