Social Engineering Attack Detection Using Machine Learning: Text Phishing Attack

Asma A. Alsufyani

College of Computers and Information Technology
Taif University
Taif, Saudi Arabia
asma.msufyani@gmail.com

Sabah M. Alzahrani

College of Computers and Information Technology
Taif University
Taif, Saudi Arabia
Sa.sabah@tu.edu.sa

Abstract- The expansion of the internet leads to an increase in the number of cyber-attacks over the days. One of the most common cybersecurity attacks is social engineering, which depends on human physiology. The phishing attack is the most popular form of social engineering. The phishing attacks have many forms, but the traditional one from them is the messages. We need techniques to protect us from these attacks. Awareness, usage policies, and other procedures are not enough. Therefore, we proposed to use natural language processing (NLP) along with machine learning techniques for text phishing detection in this paper. We started with 6,224 emails from an existing dataset that contains both phishing and legitimate emails. NLP was used for preparing the data before extracting features from it and using the features for training the classification models by machine learning algorithm and for testing these models. The features were extracted using the Continuous Bag of Words (CBOW) in the Word2Vec algorithm. We are training four models using four different machine learning algorithms which are k-nearest neighbors (KNN), Multinomial Naive Bayes (MNB), Decision Tree and AdaBoost. The developed models had to classify the text messages into two categories, which are phishing and legitimate. While the dataset is unbalanced, we used performance measurements for unbalanced data in the evaluation process. Three of our models, which were trained by KNN, Decision Tree and AdaBoost algorithms, obtained considerable values while the MNB model obtained an insignificant value.

Keywords: AdaBoost; Decision Tree; Artificial Intelligence; Cybersecurity; Detection; K-Nearest Neighbors; Machine Learning; Multinomial Naive Bayes; Natural Language Processing; Phishing; Social Engineering; Word2Vec.

1. Introduction

The quick technical development in communications and networks has made us dependent upon it in most of the activities in our lives, e.g., money management, communication, shopping and education. The key features of the Internet such as availability, anonymity and unmanageable make it an appropriate environment for cybercrime, which threatens people along with networks and devices [Şahingöz, Ö. K., Buber, E., Demir, Ö., and Diri, B. (2017)].

In the cyber world, social engineering (SE) attacks are popular, easy to perform, and considered as a first step for other attacks [Lansley, M., Mouton, F., Kapetanakis, S., and Polatidis, N. (2020)] [Ni, S., Qian, Q., and Zhang, R. (2018).] [Yasin, A., Fatima, R., Liu, L., Yasin, A., and Wang, J.(2019)]. They are critical threats to cyber systems, end users and data [Wang, Z., Sun, L., & Zhu, H. (2020)]. While there is no computer system that does not depend on humans, SE depends on the human factor to gain access to the valuable data and to the systems [Stergiou, D. (2013)]. While people behave based on their feelings and thoughts, social engineers can exploit this knowledge to achieve some goals. Intelligent manipulation of human emotions like curiosity, greed, trust, sympathy and fear can help with this attack [Yasin, A., Fatima, R., Liu, L., Yasin, A., and Wang, J.(2019)]. SE is a set of techniques used to play on human psychology to have some control of what a victim says or does.

There are many SE attacks and many taxonomies for these attacks [Ivaturi, K., and Janczewski, L. (2011)]. SE includes several types of attacks. These attacks differ from each other based on many factors, which classify the attacks into sub-classes. SE has a life cycle that includes four main stages: collecting data, developing a relationship, exploitation, and exit. For a successful SE attack, the attackers must have motivations, techniques that will be used to launch the attack, and chance such as a right time [Ghafir, I., Prenosil, V., Alhejailan, A., and Hammoudeh, M. (2016)].

A phishing attack is one of the well-known attacks in SE [Gupta, S., Singhal, A., and Kapoor, A. (2016)] [Yeboah-Boateng, E. O., and Amanor, P. M. (2014)]. The main goal of this type of attack is to access sensitive data by deceiving people for a malicious goal [Salahdine, F., and Kaabouch, N. (2019)]. Sending fraudulent emails is the traditional way of the phishing attack. There are many other ways to conduct it like calling the target directly, luring the target to use a fake web page, sending spam emails, or using advertisements. The private data may be data we do not care about hiding, like the date of birth, favorite book, favorite place, name of your child or any data that can be used to obtain more important information or to access a system.
In 2019, the greatest number of victims in the United State was for the social engineering victims [Ghafir, I., Prenosil, V., Alhejailan, A., and Hammoudeh, M. (2016)]. We need to protect our systems, data, and selves in different and smart methods. Enforce security policies, awareness and education, authentication measures are some of the existing SE attacks countermeasures, but we still need more.
In our project, we developed four models for SE attacks detection, precisely for text phishing attacks, using machine learning techniques. We used NLP together with four machine learning algorithms, which are KNN, MNB, Decision Tree and AdaBoost, to build the models. When we compared them based on several performance measures, we found that the KNN, Decision Tree and AdaBoost models have good values while the MNB model has low values.
The remainder of this paper is organized as follows. First, we present a literature survey of some related research works in section 2. In section 3, we discuss the research methodology. The results are displayed in section 4. Finally, our concluding remarks are provided in section 5.

2. Related Work

The researchers in [Chandrasekaran, M., Narayanan, K., and Upadhyaya, S. (2006)] proposed partitioning the email message into different parts and after that extract structural features of each of those parts. Using the Support Vector Machine algorithm (SVM) within a dataset of 400 emails, they evaluated the structural features. They got a good result.

The study [Miyamoto, D., Hazeyama, H., & Kadobayashi, Y. (2008)] scrutinized nine machine learning algorithms, which are adaboost, Bagging, Support Vector Machines, Classification and Regression Trees, Logistic Regression, Random Forests, Neural Networks, Naive Bayes, and Bayesian Additive Regression Trees, and compared between them to find which one has a high-quality for detecting phishing websites. The dataset that was used with these algorithms included 3000 samples. The adaboost algorithm has proven effective.

The study [Şahingöz, Ö. K., Buber, E., Demir, Ö., and Diri, B. (2017)]. suggests a real-time phishing detection system depending on Natural Language Processing (NLP) of urls and by using several machine learning classification algorithms, such as Naive Bayes, KNN and Adaboost. After the researchers build a balanced dataset of malicious and benign urls that contains 73,575 urls, they preprocess the data and extract three types of features which are NLP, word based and hybrid features. When evaluated the features by the classifiers, a highest accuracy of 97.98\% was achieved by Random Forest.

The paper [Parekh, S., Parikh, D., Kotak, S., and Sankhe, S. (2018)] aims to detect phishing websites using machine learning techniques. The dataset in this paper was collected from the phishtank site. The urls features were extracted as a first step. Then, the Random Forest algorithm was used for the classification purposes. The built machine learning model obtained a high accuracy value. To achieve better performance, each step applied a different technique for data processing.

The authors in [Baykara, M.and Gürel, Z. (2018)] built a software that analyzes the content of emails to detect phishing and spam mails. They used the Bayesian algorithm with a spam words dataset for spam emails detection. After calculating the weight of spam words in a message and classifying it as a spam message, this message will be forwarded to spam mail.

3. Methodology

In this paper, we use an existing phishing dataset. First, we prepare this data for training with NLP techniques. Next, we extract features from the processed data. After that, we train the classification models using the extracted features with the machine learning algorithm. Figure 1 illustrates our methodology. We use python language and its libraries to implement all tasks because it is a simple and flexible language and also it has a large number of machine learning libraries.

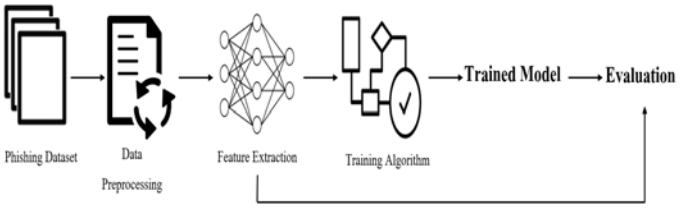

Fig. 1: Proposed Method

3.1. *Data*

The researchers in [El Aassal, A., Moraes, L., Baki, S., Das, A., Verma, R., and Verma, A. D. R. (2018)]. collected real data from as wide a range of sources as possible to create a varied dataset, and also created artificial data. The dataset contains two types of emails, which are full header email messages and no-header email messages. Each of these two types divides into phishing and legitimate messages. They had difficulty collecting phishing emails for many reasons such as companies do not prefer to publish phishing emails that they receive, and users usually remove these types of messages before they move forward. The number of phishing emails is less than the number of legitimate emails in the dataset to reflect the reality of the world, which consists of 9172 legitimate messages and 1132 phishing emails. We use their dataset in our work.

3.2. *Data Preprocessing*

We use 6,224 emails from 10,306 emails that include in the dataset that was used in the research [El Aassal, A., Moraes, L., Baki, S., Das, A., Verma, R., and Verma, A. D. R. (2018)]. All emails that do not contain headers and only full header phishing emails after removing their headers in this dataset were used. While machine learning results are greatly influenced by data preprocessing, we process our data before use it for training [Zoghi, Z. (2020)].

To achieve that, we work manually and use NLP with the data to prepare it. Deleting messages written in a language other than English, hexadecimal messages, HTML codes that exist in emails, and messages that only contain links were done manually. Where NLP processes like tokenization, lemmatization, removing stop words and punctuation, extracting part of speech, and extracting entities were done using the Python library. The NLP allows computer machines to read and understand natural human language and adds a useful numeric structure to the data [Pinto, A., Gonçalo Oliveira, H., and Oliveira Alves, A. (2016)]. We need NLP because the computer is ideal to use structured data while a lot of data that we need to work with is unstructured. To prepare the data for effective training process, we apply the following NLP concepts:

- Word Tokenization: Splitting each email in the data into its tokens/words.
- Lemmatization: Taking each single word from the emails and finding the lemma or the basic form of it.
- Removing Stop Words: Filler words such as "the", "and", "an", "or", appear more often than other words in the text and that make some noise when working with this text. Filtering the useless words is important to have a high performed text classification model.
- Extracting Part of Speech: Examining each word and trying to guess the part of speech of this word.
- Extracting Named Entity Recognition: Extracting and classifying main data (entities) in the text.

3.3. *Feature Extraction*

Extracting features from the input data is important for getting a learned model with a high performance. This process aims to reduce the number of features and focus on the key features for a fast and effective training. We use CBOW in Word2Vec technique with the processed data for feature extraction [Waykole, R. N., and Thakare, A. D. (2018)].

The word embedding is a natural language processing technique. It follows the idea of distributional hypothesis, in which the semantically similar words tend to appear in the similar linguistic contexts. The real-valued vectors represent the words in a vector space. These vectors seek to capture the features of the neighbors of a word [Waykole, R. N., and Thakare, A. D. (2018)] [Young, T., Hazarika, D., Poria, S., and Cambria, E. (2018)].

Word2Vec is a word embedding method, which is used to train model that recreate the word semantic context. It deals with large datasets that contain text. After entering the data into the created model by word2Vec, the word2Vec will produce a multidimensional vector space. Each word in the dataset will be expressed by a vector,

where each word is assigned an array of numbers. The similar words in context are located nearby in the vector space. The numeric representation of words helps to perform mathematical operations on the words [Waykole, R. N., and Thakare, A. D. (2018)] [Young, T., Hazarika, D., Poria, S., and Cambria, E. (2018)].

Continuous skip gram and continuous bag of words (CBOW) are the two model architectures of word2vec. The expectation of the contextual words of a specific word is the function of continuous skip gram. The continuous bag of words (CBOW) algorithm is for a target word prediction from its context. The CBOW is quicker and has better performances for the most common words [Waykole, R. N., and Thakare, A. D. (2018)] [Young, T., Hazarika, D., Poria, S., and Cambria, E. (2018)].

3.4 Model Training

We split the features that were extracted in the previous phase into two groups which are 80% of them for training and 20% for testing. After that, using 80% of the features we train four models by four classification algorithms which are KNN, MNB, Decision Tree and AdaBoost.

- **K-Nearest Neighbor Algorithm (KNN)**

KNN is a supervised machine learning algorithm used for regression and classification purposes. In classification cases, it places a new data into a predefined class based on the similarity between this new data and the classified data in this class, that can be achieved by calculating the minimum distance between the new data and the classified data. Euclidean distance is a common method that is used to calculate the distance in KNN, which calculates a straight line between two points [Cahyani, D. E., and Nuzry, K. A. P. (2019)]. Authors in a paper [Repalle, S. A., and Kolluru, V. R. (2017)] mentioned that KNN algorithm is the best to use with intrusion detection systems.

- **Multinomial Naive Bayes Algorithm (MNB)**

MNB is a parametric model that is normally used for text classification. The main idea of this algorithm is that every feature in the dataset is independent of other features, which means the probability of one feature presence is not affected by the presence of another feature. It is significantly effective in terms of computation and easy to execute [Kibriya, A. M., Frank, E., Pfahringer, B., and Holmes, G. (2004)].

- **Decision Tree Algorithm**

It is a supervised machine learning algorithm. Due to many remarkable characteristics it has, such as the lack of parameters, comprehensibility, and simplicity, it is considered one of the most powerful machine learning algorithms. Furthermore, it has the ability to classify rapidly, learn faster, deal with mixed types of data, be applied serially and concurrently, and be constructed easily [Brijain, M., Patel, R., Kushik, M. R., & Rana, K. (2014)] [Su, J., & Zhang, H. (2006)] [Chauhan, H., & Chauhan, A. (2013)].
It seeks to build a prediction model that uses decision rules to predict values. It works in an iterative manner which begins from above to underneath. The dataset at the root is divided by the best decision rule into separate subsets. This process is repeated with each subset/sub-tree until classifying all instances in the dataset [Su, J., & Zhang, H. (2006)].

- **AdaBoost Algorithm**

AdaBoost is an easy to implement machine learning algorithm that has a solid theoretical foundation [Hu, W., Hu, W., & Maybank, S. (2008)]. It employs an iterative process to improve poor classifiers by learning from their errors. It works sequentially and by taking the advantage of the dependence between models, it gives the samples that the model could not label them higher weights [Unnithan, N. A., Harikrishnan, N. B., Vinayakumar, R., Soman, K. P., and Sundarakrishna, S. (2018)].

4. Experimental Evaluation

We will evaluate the performance of each classification model using the testing set, which represented 20% of the whole features, with three performance measurements for imbalanced data, which are the Confusion Matrix, F-value and ROC-AUC.

4.1. *Confusion Matrix*

Confusion Matrix is a performance measurement table, which contains two types of data: true data and predicted data. Where figure 2 illustrates the traditional confusion matrix, that contains a number of items in each group, of KNN model, figure 3 shows the confusion matrix heat map, in percentage, of it. In the heat map, the low

values take dark colors whereas the high values take light colors. The confusion matrix proves that the KNN model performed well.

```
==================
Confusion Matrix:
==================

    0    1
[[898   36]  0
 [ 47 185]] 1
```

Fig. 2. KNN Model Confusion Matrix

```
==============================
Confusion Matrix with Heat MAP:
==============================

<AxesSubplot:>
```

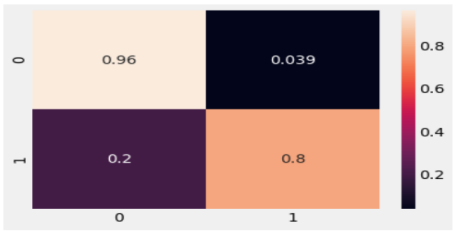

Fig. 3. Confusion Matrix with Heat Map of KNN Model

Confusion matrix in figures 4 and 5 shows that there is a defect in the MNB classification model in which the values of TP and FP are too low while the values of TN and FN are high.

```
==================
Confusion Matrix:
==================

    0    1
[[933   1]  0
 [232   0]] 1
```

Fig. 4. MNB Model Confusion Matrix

```
==============================
Confusion Matrix with Heat MAP:
==============================

<AxesSubplot:>
```

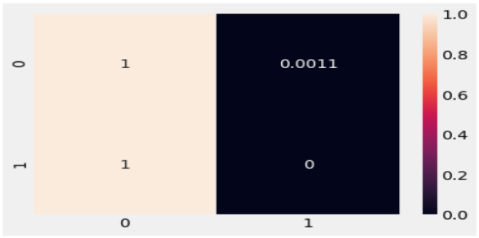

Fig. 5. Confusion Matrix with Heat Map of MNB Model

Figures 6 and 7 present that our Decision Tree model works well while the values of TP and TN indicate that the classifier can predict the two classes.

```
==================
Confusion Matrix:
==================

    0     1
[[874   60]   0
 [ 60 172]]   1
```

Fig. 6. Decision Tree Model Confusion Matrix

```
==============================
Confusion Matrix with Heat MAP:
==============================

<AxesSubplot:>
```

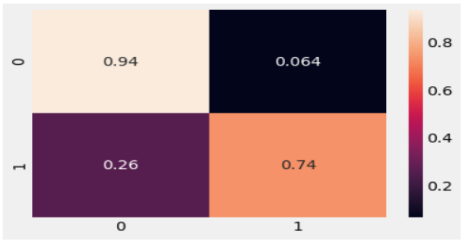

Fig. 7. Confusion Matrix with Heat Map of Decision Tree Model

Confusion matrix in Figures 8 and 9 explains that the AdaBoost trained model can classify correctly in most cases.

```
==================
Confusion Matrix:
==================

    0     1
[[893   41]   0
 [ 58 174]]   1
```

Fig. 8. AdaBoost Model Confusion Matrix

```
==============================
Confusion Matrix with Heat MAP:
==============================

<AxesSubplot:>
```

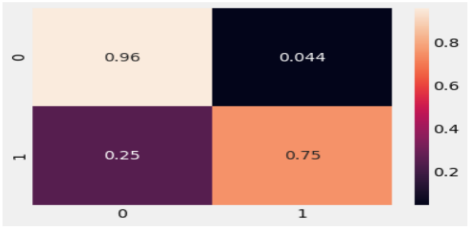

Fig. 9. Confusion Matrix with Heat Map of AdaBoost Model

4.2. *F-value*

F-value is a machine learning performance measurement. It represented by:

$$F - value = (2 * precision * recall) \div (precision + recall)$$

in which:

$$precision = TP \div (TP + FP)$$
$$recall = TP \div TP + FN$$

We use the classification Report to calculate the F-value of each class, which **0** represents the legitimate class whereas **1** represents the phishing class. Figures 10, 11, 12 and 13 show the F-values of KNN model, MNB model, Decision Tree model and AdaBoost Model.

```
======================
Classification Report:
======================
```

	precision	recall	f1-score	support
0	0.95	0.96	0.96	934
1	0.84	0.80	0.82	232
accuracy			0.93	1166
macro avg	0.89	0.88	0.89	1166
weighted avg	0.93	0.93	0.93	1166

Fig. 10. F-value of KNN Model

```
======================
Classification Report:
======================
```

	precision	recall	f1-score	support
0	0.80	1.00	0.89	934
1	0.00	0.00	0.00	232
accuracy			0.80	1166
macro avg	0.40	0.50	0.44	1166
weighted avg	0.64	0.80	0.71	1166

Fig. 11. F-value of MNB Model

```
======================
Classification Report:
======================
```

	precision	recall	f1-score	support
0	0.94	0.94	0.94	934
1	0.74	0.74	0.74	232
accuracy			0.90	1166
macro avg	0.84	0.84	0.84	1166
weighted avg	0.90	0.90	0.90	1166

Fig. 12. F-value of Decision Tree Model

```
======================
Classification Report:
======================
```

	precision	recall	f1-score	support
0	0.94	0.96	0.95	934
1	0.81	0.75	0.78	232
accuracy			0.92	1166
macro avg	0.87	0.85	0.86	1166
weighted avg	0.91	0.92	0.91	1166

Fig. 13. F-value of AdaBoost Model

4.3. *ROC-AUC*

Receiver Operating Characteristic (ROC) is one of the most critical evaluation measures for checking the performance of a classification model. It summarizes the performance of a model at every classification threshold. The Area Under the Curve (AUC) is a ROC curve performance metric. A classification model has a perfect separability measure when this model has a value of AUC close to one. While it has a bad separability measure when it has a value of AUC from 0 to 0.50. The TPR and FPR equations [Chawla, N. V. (2009)]:

Sensitivity= Recall= TPR= TP÷ (TP + FT) %
1 − Specificity = FPR = FP ÷ (FP + TN) %

Figure 14 displays the ROC-AUC values of all trained classifiers. The KNN model, Decision Tree model and AdaBoost model have high values while MNB model has a worthless value. The ROC-AUC value of the MNB model indicates that the model classifies all data in only one class. Table I summarize the performance values of the four classification models.

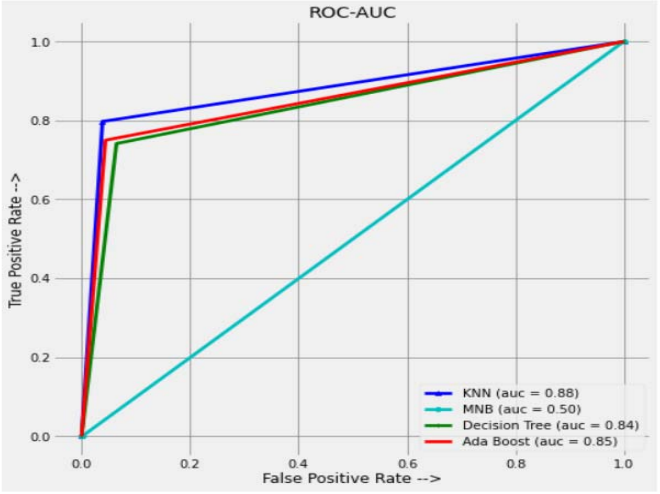

Fig. 14: ROC-AUC of all Models

performance measurement/ classification model	KNN	MNB	Decision Tree	AdaBoost
Accuracy	0.93	0.8	0.9	0.92
Confusion Matrix	TP= 0.80 TN= 0.96 FP= 0.039 FN= 0.2	TP= 0.0 TN= 1 FP= 0.0011 FN= 1	TP= 0.74 TN= 0.94 FP= 0.064 FN= 0.26	TP= 0.75 TN= 0.96 FP= 0.044 FN= 0.25
F-value	Phish= 0.82 Legit= 0.96	Phish= 0.0 Legit= 0.89	Phish= 0.74 Legit= 0.94	Phish= 0.78 Legit= 0.95
AUC	0.88	0.5	0.84	0.85

Table I. Performance Values

5. Conclusion

As a result of the growing use of the Internet in our everyday lives, attackers in cyberspace target users of this platform. Social engineering attacks are one of the most common cyber-attacks. Phishing attack is a type of social engineering attack, which can be performed over several ways. Text phishing attack is a prevalent kind of phishing attack. In our project, we addressed the text phishing attack using machine learning and natural language processing. We trained four classification models to distinguish phishing text from normal text. KNN, MNB, Decision Tree and AdaBoost algorithms were used with features of preprocessed data for training the classifiers. We found that KNN, Decision Tree and AdaBoost models are effective against text phishing attack, whereas the MNB model did not give the expected outcome.

References

[1] Baykara, M., & Gürel, Z. Z. (2018, March). Detection of phishing attacks. In *2018 6th International Symposium on Digital Forensic and Security (ISDFS)* (pp. 1-5). IEEE.
[2] Brijain, M., Patel, R., Kushik, M. R., & Rana, K. (2014). A survey on decision tree algorithm for classification.
[3] Cahyani, D. E., & Nuzry, K. A. P. (2019, November). Trending Topic Classification for Single-Label Using Multinomial Naive Bayes (MNB) and Multi-Label Using K-Nearest Neighbors (KNN). In *2019 4th International Conference on Information Technology, Information Systems and Electrical Engineering (ICITISEE)* (pp. 547-552). IEEE
[4] Chawla, N. V. (2009). Data mining for imbalanced datasets: An overview. *Data mining and knowledge discovery handbook*, 875-886.
[5] Chauhan, H., & Chauhan, A. (2013). Implementation of decision tree algorithm c4. 5. *International Journal of Scientific and Research Publications*, *3*(10), 1-3.
[6] Chandrasekaran, M., Narayanan, K., & Upadhyaya, S. (2006, June). Phishing email detection based on structural properties. In *NYS cyber security conference* (Vol. 3).
[7] El Aassal, A., Moraes, L., Baki, S., Das, A., Verma, R., & Verma, A. D. R. (2018). Anti-phishing pilot at ACM IWSPA 2018. In *Proc. 1st AntiPhishing Shared Pilot 4th ACM Int. Workshop Secur. Privacy Anal.(IWSPA)* (pp. 1-9). Tempe, AZ, USA.
[8] Ghafir, I., Prenosil, V., Alhejailan, A., & Hammoudeh, M. (2016, August). Social engineering attack strategies and defence approaches. In *2016 IEEE 4th international conference on future internet of things and cloud (FiCloud)* (pp. 145-149). IEEE.
[9] Gupta, S., Singhal, A., & Kapoor, A. (2016, April). A literature survey on social engineering attacks: Phishing attack. In *2016 international conference on computing, communication and automation (ICCCA)* (pp. 537-540). IEEE.
[10] Hu, W., Hu, W., & Maybank, S. (2008). Adaboost-based algorithm for network intrusion detection. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, *38*(2), 577-583
[11] Ivaturi, K., & Janczewski, L. (2011, June). A taxonomy for social engineering attacks. In *International Conference on Information Resources Management* (pp. 1-12). Centre for Information Technology, Organizations, and People.
[12] Kibriya, A. M., Frank, E., Pfahringer, B., & Holmes, G. (2004, December). Multinomial naive bayes for text categorization revisited. In *Australasian Joint Conference on Artificial Intelligence* (pp. 488-499). Springer, Berlin, Heidelberg
[13] Lansley, M., Mouton, F., Kapetanakis, S., & Polatidis, N. (2020). SEADer++: social engineering attack detection in online environments using machine learning. *Journal of Information and Telecommunication*, *4*(3), 346-362.
[14] Miyamoto, D., Hazeyama, H., & Kadobayashi, Y. (2008, November). An evaluation of machine learning-based methods for detection of phishing sites. In *International Conference on Neural Information Processing* (pp. 539-546). Springer, Berlin, Heidelberg.
[15] Ni, S., Qian, Q., & Zhang, R. (2018). Malware identification using visualization images and deep learning. *Computers & Security*, *77*, 871-885.
[16] Parekh, S., Parikh, D., Kotak, S., & Sankhe, S. (2018, April). A new method for detection of phishing websites: URL detection. In *2018 Second international conference on inventive communication and computational technologies (ICICCT)* (pp. 949-952). IEEE.
[17] Pinto, A., Gonçalo Oliveira, H., & Oliveira Alves, A. (2016). Comparing the performance of different NLP toolkits in formal and social media text. In *5th Symposium on Languages, Applications and Technologies (SLATE'16)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.
[18] Repalle, S. A., & Kolluru, V. R. (2017). Intrusion detection system using ai and machine learning algorithm. *International Research Journal of Engineering and Technology (IRJET)*, *4*(12), 1709-1715.
[19] Şahingöz, Ö. K., Buber, E., Demir, Ö., & Diri, B. (2017). Machine Learning Based Phishing Detection from URIs.
[20] Salahdine, F., & Kaabouch, N. (2019). Social engineering attacks: a survey. *Future Internet*, *11*(4), 89.
[21] Stergiou, D. (2013). Social Engineering and Influence: A Study that Examines Kevin Mitnick's Attacks through Robert Cialdini's Influence Principles
[22] Su, J., & Zhang, H. (2006, July). A fast decision tree learning algorithm. In *AAAI* (Vol. 6, pp. 500-505).
[23] Wang, Z., Sun, L., & Zhu, H. (2020). Defining Social Engineering in Cybersecurity. *IEEE Access*, *8*, 85094-85115.
[24] Waykole, R. N., & Thakare, A. D. (2018). A Review of feature extraction methods for text classification. *Int. J. Adv. Eng. Res. Dev*, *5*(04).
[25] Unnithan, N. A., Harikrishnan, N. B., Vinayakumar, R., Soman, K. P., & Sundarakrishna, S. (2018). Detecting phishing E-mail using machine learning techniques. In *Proc. 1st Anti-Phishing Shared Task Pilot 4th ACM IWSPA Co-Located 8th ACM Conf. Data Appl. Secur. Privacy (CODASPY)* (pp. 51-54).
[26] Yasin, A., Fatima, R., Liu, L., Yasin, A., & Wang, J. (2019). Contemplating social engineering studies and attack scenarios: A review study. *Security and Privacy*, *2*(4), e73.
[27] Yeboah-Boateng, E. O., & Amanor, P. M. (2014). Phishing, SMiShing & Vishing: an assessment of threats against mobile devices. *Journal of Emerging Trends in Computing and Information Sciences*, *5*(4), 297-307.
[28] Young, T., Hazarika, D., Poria, S., & Cambria, E. (2018). Recent trends in deep learning based natural language processing. *ieee Computational intelligenCe magazine*, *13*(3), 55-75.
[29] Zoghi, Z. (2020). Ensemble Classifier Design and Performance Evaluation for Intrusion Detection Using UNSW-NB15 Dataset (Doctoral dissertation, University of Toledo).

Authors Profile

Asma A. Alsufyani, Master Cybersecurity**,** College of Computers and Information Technology, Taif University, Taif, Saudi Arabia
Sabah M. Alzahrani, Assistance professor of Computer and Information systems Engineering. Vice Chair of Computer Engineering Department, College of Computers and Information Technology, Taif University, Taif, Saudi Arabia