Impact Analysis of Training Data Characteristics for Phishing Email Classification

Akash Sundararaj* and Gökhan Kul
University of Massachusetts Dartmouth, Dartmouth, MA 02747 United States
{asundararaj6, gkul}@umassd.edu

Received: January 16, 2021; Accepted: March 31, 2021; Published: June 30, 2021

Abstract

E-mail is the most essential form of formal communication for organizations. However, phishing attacks occurring through e-mail are a prevalent threat, and these attacks are steadily rising even after e-mail filters to prevent these attacks have become ubiquitous. Phishing attacks are often one of the first steps of major hacking attempts such as Advanced Persistent Threat (APT) attacks or ransomware attacks. In this work, we look into the training data that phishing e-mail detectors to identify the ideal dataset parameters to optimize the phishing e-mail classifiers. To perform this assessment, we surveyed through phishing e-mail detection methods in the literature and identified that majority of phishing e-mail detectors either use structural properties or text mining methods. Therefore, we analyze the optimal ratio for phishing and legitimate e-mails in the training data for these approaches. We design an experiment using Enron dataset and a phishing e-mail collection to evaluate the effectiveness of these methods with varying sizes of legitimate and phishing emails to empirically show their strengths and weaknesses for specific data parameters. We display the influence of the balanced and unbalanced dataset of e-mails on the results produced by the machine learning classifiers. Interestingly, unbalanced datasets provide better accuracy while they consistently provide worse precision and recall compared to balanced datasets. The empirical results also suggest that phishing e-mail filters have not been perfected, warranting that there is still room for development in this area. Our findings will help the researchers to avoid the common mistakes native to this type of threat before building machine learning classifiers for this domain.

Keywords: Datasets, E-mail, Machine learning, Phishing detection

1 Introduction

Phishing is the fraudulent attempt to obtain sensitive information such as usernames, passwords and credit card details by disguising as a trustworthy entity in an electronic communication [6]. Users are lured by communications such as e-mails purporting to be from the trusted parties such as social websites, auction sites, banks, online payment processors [5]. Furthermore, phishing e-mails are considered one of the first steps of Advanced Persistent Threat (APT) attacks [7].

Around half of the world is using e-mail as of 2020 which can be clearly seen from the steady growth rate of e-mails transferred per day [16]. The number of e-mails transferred per day will exceed 306.4 billions in 2020 and expected to reach 347 billion by 2023 predicted by Statista [4]. Even though the benefits offered by the e-mail communication are numerous, the number of risks associated with this form of communication are also growing in numbers. Evidently, wide usage of e-mail makes it

Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA), 12(2):85-98, June 2021
DOI:*10.22667/JOWUA.2021.06.30.085*

*Corresponding author: Department of of Computer and Information Science, University of Massachusetts Dartmouth, 285 Old Westport Rd., Dartmouth, MA, 02747 United States

a convenient and inexpensive tool to implement phishing attacks. Combined with social engineering methods, phishing e-mails can be extremely convincing for the end-users [10], therefore, they can cause serious problems if not filtered by a mechanism that can distinguish illegitimate e-mails. Once the e-mail reaches the inbox of the end-user, the unwitting user becomes susceptible to the phishing attack. The victim users, if they fail to identify the phishing attack and give sensitive information to the attackers, they risk both themselves and their organizations. Therefore, even though these users do not aim to harm their organizations, they become an *insider threat* to their organization [11, 12]. Moreover, even capable users are susceptible to phishing attacks due to over-confidence [19].

Majority of the research work on phishing e-mails focuses on the direction of improving the accuracy of the machine learning (ML) classifiers in the detection of phishing e-mails [3, 1, 2, 9, 15, 13, 14, 8]. Although ML is a powerful tool that is efficient in detecting phishing attacks, frequent hacking attempts via phishing show that many phishing detection systems still fail to detect phishing e-mail variants.

One common problem we noticed with most of the existing research works is the nature of dataset which is used for training and testing the ML algorithm used in phishing detection mechanisms. The datasets mostly consist of a constant number of labeled legitimate and phishing e-mails, and the experiments mostly include cross-validation of the full dataset with the proposed ML algorithm without intentionally changing the ratio of legitimate and phishing e-mails in the training and test sets. Ignoring this important factor leads to different interpretations of the work being presented due to the fact that the tuning of the ML classifier performance is dependent on the dataset structure.

Other branches of research work on phishing e-mails focus on psychological factors [5, 19], and usability and user experience factors [18].

In this paper, *our aim is to identify the insights from the existing research works in detection of phishing e-mails with machine learning classifiers by comparing the results obtained from them using a new combined dataset and a set of experiments*. The findings obtained from this work can be considered and used for building effective machine learning classifiers for phishing detection in the future. We focus on the ratio of legitimate and phishing emails used in the training of supervised machine learning classifiers. We analyze the impact of the size of phishing and legitimate e-mails on accuracy, precision and recall scores of specific classification techniques. We would like to emphasize that our aim is not comparing ML algorithms against each other to show their efficacy.

This paper is organized as follows. We start by going through the literature related to phishing e-mail in Section 2. Section 3 explains the dataset and experiment design used in this work, and the approaches adopted by many works in the literature followed by the experimental results of these approaches. We finally conclude and briefly explain our future work in Section 4.

2 Related Work

In this section, we first look into works that utilize supervised machine learning classifiers to detect phishing e-mails.

Many research papers in the field of detection of phishing e-mails utilize machine learning classifiers and claim better accuracy, precision or recall, and selecting the best *feature set* possible to achieve the higher accuracy in the classification [3, 15, 13, 8].

The feature set includes different set of common keywords present in the phishing e-mails, and in some research papers, features such as the presence of URL links or number of URL links in the e-mails is included as one of the features for the algorithm. Some of the research works that use a variety of features are listed in Table 1. These papers are selected due to the fact that they represent different methodologies and feature sets, and we do not claim that we present here a comprehensive literature review, since this would be out of the scope of our work. Table 1 provides technical information about

various works, classifiers used for identifying the phishing e-mails, evaluation method for measuring the effectiveness of classifiers and results obtained from the research works.

Instead of building a new machine learning classifier or identifying the right feature set for the classification, *the aim of this research work is to identify the key insights from the already existing research works which will help the future research*. Based on our findings, we are going to describe the common pitfalls which the researchers can avoid in the future.

Research	**Motivation**	**Feature**	**Feature Structure**	**Classification**
Chandrasekaran *et al.* (2006) [3]	Efficient Classification	- Function words - Structural attributes	Vector	Yes
Toolan *et al.* (2010) [15]	Best Feature Selection	- Body - URL - Subject - Script - Sender	Vector	No
Pandey *et al.* (2012) [13]	Statistical Significance	- Keywords	Vector	No
Verma *et al.* (2013) [17]	Semantic Classification	- Structural features - Keywords - Semantic Association	Vector	Yes
Gutierrez *et al.* (2016) [8]	Efficient Classification	- Structural features - Keywords - Tailored words	Vector	Yes

Table 1: Representation of a variety of feature sets in Phishing Detection Literature

Chandrasekaran *et al.* [3] proposed the idea of using the distinct structural properties of the e-mail as the features for the Support Vector Machines classifier to distinguish between the legitimate e-mails and phishing e-mails. The paper explains about the need of identifying the phishing attempts without checking the authenticity of the websites. The existing methods for detecting the phishing attempts will take the users closer to the hackers and they become vulnerable to the attacks. The author of this paper concluded that the effectiveness of the results in the classification of phishing e-mails from this paper is clearly influenced by the features selected for the classifier.

Verma *et al.* [17] performs keyword selections on the e-mails and applies semantic associations on the features, which eliminates the need to re-train the machine learning model due to changing wording of phishing e-mails that are sent out with minor changes.

Gutierrez *et al.* [8] also applies natural language processing (NLP) methods, and proposed the idea of creating the Semi-Automated Feature generation for Phish classification (SAP-PC) to extract higher level features which are meant to defeat the existing phishing detection strategies. It applies the NLP techniques to collect the five different kinds of features to create the feature set - commonly known phishing words from domain knowledge and their synonyms, words associated with tier-one research institution, commonly occurring words from our phishing corpus and synonyms, proper noun organization names and their types, structural features in the e-mail. The Random Under-Sampling Boost algorithm [8] is created to handle the imbalanced nature of the dataset. The SAF-PC can train the new samples without retraining the entire training data. In this paper, it is concluded that the SAF-PC is efficient and able to detect the 70% of the phishing e-mails which eluded the state-of-the-art e-mail filtering tool. Although we also would have liked to evaluate their strategy, we could not access some certain information

regarding the model the authors created.

Toolan *et al.* [15] proposed the idea of collecting the effective features which can be used for the classification of phishing and legitimate e-mails. 40 features are collected based on the usage in the existing research works and they are experimented with three datasets. The first dataset consists of only Ham and Spam e-mails, the second dataset consists of Ham and Phish e-mails, the third dataset consists of three classes Ham, Spam and Phish e-mails. The conclusion from this paper provided top 10 best features to distinguish the phishing and legitimate e-mails based on the information gain value.

Pandey *et al.* [13] proposed the idea of measuring the statistical significance of the classifiers with and without feature selection. It uses the text mining to collect the features from the dataset, then the top 50 percent features are selected using t-statistic method. Several machine learning classifiers are implemented to distinguish the phishing and legitimate e-mails. The conclusion from this paper mentioned that all the classifiers are not statistically significant by higher value with and without feature selection except Probabilistic Neural Net.

We adopt the methods "Phishing E-mail Detection Based on Structural Properties [3]" and "Detecting Phishing e-mails using Text and Data mining [13]" in this work to perform our evaluation. We selected these methods due to their universal nature. Note that other supplementary methods such as fine tuning the learning algorithm for the context or dataset, phishing detection based on identifying the URL part of the e-mails and checking the validity of the websites linked to the e-mails can also be added for increased accuracy. The problem with manually or automatically inspecting links on a phishing e-mail also has the risk of exposing users to the hacking methods.

We will show the problems in the detailed manner in the following sections. The researchers can use this observation to build the machine learning classifiers and to create the dataset effectively.

3 Methodology

In this section, we present (1) how we created our dataset and how we change its structure, (2) how the dataset characteristics impact methods using structural properties, and (3) how the dataset characteristics impact methods using text mining. We do not aim to compare ML algorithms and their efficacy, therefore, we only show the dataset balance and imbalance using the algorithms presented in the representative papers.

3.1 Creating the experiment datasets

To create the experiment dataset, we used two types of e-mails: (1) Legitimate e-mails, and (2) Phishing e-mails. Note that spam e-mails could also be used, but we would like to focus our attention to phishing since it still is one of the primary methods in the initial phase of a cyber attack. The dataset is prepared based on the collection of e-mails from the two different sources. The total number of e-mails present in the dataset is 525,754. Out of this, the number of phishing e-mails is 8,352 and number of the legitimate e-mails is 517,402.

The legitimate e-mail dataset we used, Enron Corpus [1,2] is a large public database generated by 158 employees of Enron corporation. Phishing e-mails [3] are collected from the monkey.org webpage, and it contains the e-mails collected each year starting from 2015 to 2019. The phishing e-mails available here are not in structured form, so we parsed these e-mails using Python code which we created using *Beautiful Soup* [4] package of Python to remove the html tags and retained all the other fields like From,

[1] https://www.cs.cmu.edu/~enron/
[2] https://www.kaggle.com/wcukierski/enron-email-dataset
[3] https://monkey.org/~jose/phishing/
[4] https://www.crummy.com/software/BeautifulSoup/bs4/doc/

Table 2: E-mail Parsing

Before Parsing	**After Parsing**
To confirm your eBay records click here: <a HREF=3Dhttp://140.130.108.11/account/aw-confirm/ebayDLLupdate/index.html target=3D"_self">http://cgi1.ebay.com/aw-cgi/ebayISAPI.dll?UPdate We appreciate your support and understanding, as we work together to keep eBay a safe place to trade. Thank you for your patience in this matter. 	To confirm your eBay records click here: http://cgi1.ebay.com/aw-cgi/ebayISAPI.dll? UPdate We appreciate your support and understanding, as we work together to keep eBay a safe place to trade. Thank you for your patience in this matter.

To, Date, Subject, Message-ID, X-IMAP, Status, Return-Path, X-Original-To, Delivered-To, Received, Content-Type,Content-Transfer-Encoding,Content-Description, MIME-Version, X-status, X-keywords, X-UID present in those e-mails. For example, we have taken small part of a single e-mail from the dataset and we are going to show how the parsing of the e-mails is performed. Keeping the entire e-mail in this example is difficult, so we have taken the small part from the e-mail for this purpose. In the Table 2, it can be seen that, before parsing the e-mails all the tags are present in the e-mails which is meant for styling the text elements in the e-mail. After parsing the e-mails, all of the opening and closing HTML tags like ¡html¿,¡head¿,¡p¿,¡a¿,¡ul¿,¡img¿,¡div¿,¡span¿,¡ul¿,¡ol¿ will be removed, only the remaining contents will be present. Hence, we can retain all of the required content in the e-mails. Once, the e-mails are parsed, we store those parsed e-mails without any tags in a text file. We can get the values for the features required for the analysis from these parsed e-mails. Based on the requirement of features for our purpose, we have collected the values for the features from the parsed e-mails and prepared the dataset.

The final dataset we used [5] contains these features: Account, Access, Bank, Credit, Click, Identity, Inconvenience, Information, Limited, Minutes, Password, Recently, Risk, Social, Security, Service, Suspended, Total Number of Characters, Vocabulary richness, Unique words and Phishing Status.

We also open-sourced the experiment code we created along with a reproducibility image [6].

3.2 Evaluation of Structural Properties

We first look into evaluating structural properties in e-mails, which is widely adopted, and presented in research by Chandrasekaran *et al.* [3]. It leads to the new idea of validating the e-mails based on the style

[5] https://www.kaggle.com/akashsurya156/phishing-paper1
[6] https://github.com/akashsury/Phishing-paper-1-dataset

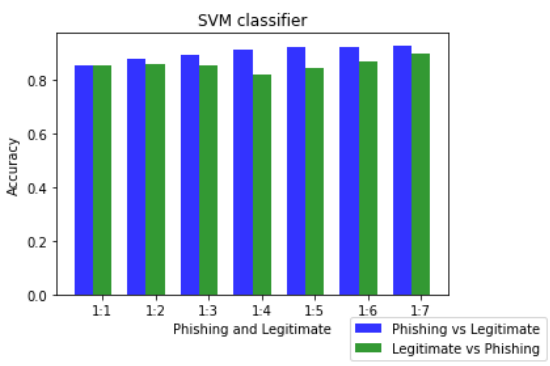
Table 3: Features from the e-mail document

Features
Total Number of characters
Total Number of distinct words
Vocabulary richness
Total number of function words
Structure of E-mail Subject line
Structure of the Greeting in e-mail body
Account
Access
Bank
Credit
Click
Identity
Inconvenience
Information
Limited
Log
Minutes
Password
Recently
Risk
Social
Security
Service
Suspended

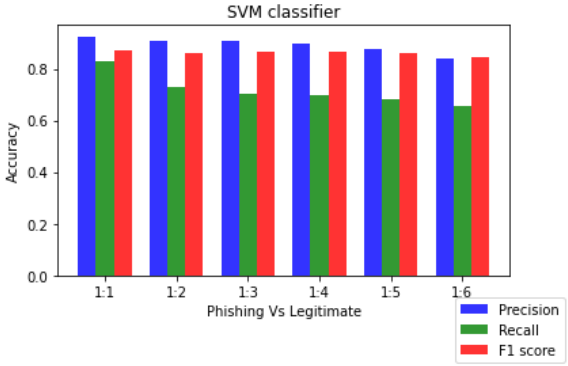
marker and structural attributes of e-mails to classify the phishing e-mails. In this paper, 18 functional words are chosen as features by observing the repository of the phishing e-mails and analyzing the common properties. The words which have length of less than 2 are omitted. The total number of the words W is counted from both header and body of the e-mails and is taken as one of the features. Most of the phishing e-mails exhibit a sense of threat, urgency or concern. The usage of the words in the legitimate e-mails which contain these characteristics are less. Another relevant feature is analyzing the presence of the suspicious pattern in the subject of the e-mails and the presence of the salutation/greeting used in the body of the e-mails.

During the implementation of this approach, Linear SVM classifier is considered to suit well for the tasks of binary classification and for achieving the least false positive rates. In Chandrasekaran *et al.* [3], it is clearly mentioned that the removal of features which measure the structural attributes in the e-mails decreases the accuracy by 20 percent. Therefore, it can be clearly seen that this feature is a good indicator of the classification.

Observations. The Support Vector Machine classifier is cross-validated 5 times with the collection of data. The accuracy scores of the Machine Learning classifier is calculated based on the match between the set of predicted labels and set of true labels. If $\hat{y}_i$ is the predicted value of the i-th sample and y_i is the corresponding true value, then the fraction of the correctly classified e-mails over all the e-mails $n_{samples}$ is defined as



(a) Accuracy comparison



(b) Precision, Recall and F1 score comparison

Figure 1: SVM classifier with varying number of phishing and legitimate e-mails using structural properties

$$accuracy(y, \hat{y}) = 1/n_{\text{samples}} \sum_{i=0}^{n_{\text{samples}}-1} l(\hat{y}_i = y_i)$$

$l(x)$ is the indicator function which compares the predicted and true values are equivalent or not. The features of the SVM classifier is listed in Table 3. SVM is evaluated using two different metrics - Accuracy and Precision, Recall, F1 score in the original research paper [3]. In their paper, the dataset consists of 200 Legitimate e-mails and 200 Phishing e-mails. This is a balanced dataset and it contains almost both of the classes in equal size. Keeping this equal ratio of phishing e-mails and legitimate e-mails for training and testing the machine learning classifier produced good accuracy in the research paper. However, in production systems, this is not the case. Number of phishing e-mails is going to be very small in size compared to that of the entire e-mail workload. Based on this perspective, we can check the efficiency of the same machine learning classifiers by adjusting the size of the dataset.

Our experiment evaluates two approaches. In the first approach, the accuracy of the SVM classifier is calculated by keeping the size of the phishing e-mails constant and increasing the size of the collection of legitimate e-mails. The accuracy scores are steadily rising with increase in size of the legitimate e-mails or phishing e-mails, which can be seen from the resultant scores shown in Figure 1a. The results have shown that the accuracy of the machine learning classifiers is increasing after each iteration with increasing the size of the legitimate e-mails, which may look like a good sign but this is due to the biased nature of the machine learning classifier. The classifier is falsely predicting some of the e-mails as legitimate e-mails. It becomes obvious with the results from the second approach. Using the Precision and Recall metric, the same experiment is repeated.

The Precision, Recall and F1 scores are calculated by increasing the size of legitimate e-mails in the same way of how the accuracy scores are calculated.

Precision = Number of phishing e-mails classified correctly/(Number of phishing e-mails classified correctly + Number of Legitimate e-mails incorrectly identified as Phishing e-mails)

Table 4: SVM classifier with parameters changes - F1 score

Phishing vs Legitimate	SVM rbf	SVM poly	SVM sigmoid	SVM gamma = 'auto'
1:1	0.87869732	0.780707518	0.781035818	0.878577977
1:4	0.865161027	0.827383369	0.738071004	0.863448173
1:5	0.862103683	0.826831592	0.728677188	0.86074171

Recall = Number of phishing e-mails classified correctly/(Number of phishing e-mails classified correctly + Number of Phishing e-mails incorrectly identified as Legitimate e-mails)

$$F1 = 2 * (Precision * Recall)/(Precision + Recall) \quad (1)$$

In the second approach of the experiment, increasing the size of the legitimate e-mails is not predicting with good accuracy. The results of this experiment can be seen in Figure 1b. After each iteration, the precision, recall and F1 score are decreasing which explains that the machine learning classifier does not perform well with the unbalanced nature of the dataset. Only limited number of research works considered this factor in their method.

The results presented as for Accuracy, Precision, Recall and F1 are calculated based on the SVM classifier with default values for the parameters. To measure the sensitivity of the SVM classifier, we perform the experiments repeatedly with varying values of the parameters. For this purpose, we have selected two parameters - kernel and gamma. These are hyper parameters which directly affect the accuracy of the SVM. The chosen parameters can be adjusted to check the changes in the results. Specifying the type of kernel will allow the machine learning classifier to produce good accuracy by capturing the different complexity of data like linear and non-linear functions. There are four types of kernels available – linear, rbf, poly and sigmoid. 'linear' type will allow SVM to separate the e-mails by drawing a line. 'poly' type can be used for high dimensions and it can be used when non-linear function is needed to separate the e-mails. 'rbf' type can be used to create the boundary curve around each of the class of the e-mails. This increases the accuracy, but it takes more time when more number of classes have to be separated. 'sigmoid' type of kernel uses logistic functions to separate the e-mails and it can be used to separate the classes when they are non-linear in nature. The parameter 'gamma' will indicate the influence of the single training example reaches with low values means 'far' and high values means 'close'. If the value of 'gamma' parameter is too high, then it will overfit the training data and if the value of 'gamma' parameter is very low the model will not be able to capture the complexity of the data. There are two values allowed for 'gamma' parameter - 'scale' and 'auto'. Gamma parameter with 'auto' as value will include the number of features Gamma('scale')=1/n_features. Gamma parameter with 'scale' as value will also include the number of features and variance of the dataset in the calculation of gamma for SVM. Gamma('scale')=1/n_features*X.var(). In the case where the variance of our dataset is high, then this will make SVM to be regularized. In the Table 4, the results for F1 score of SVM classifier can be seen with the changes applied to each of the parameters at once. The parameter- type of kernels are changed with different values like rbf, poly, sigmoid. The decreasing trend of F1 scores in Table 4 has proven that the changes of the parameters does not affect the sensitivity of the machine learning classifier and it still complies with observation we found before.

Table 5: Features from the e-mails

Features(Keywords from e-mails)
Account
Member
Access
E-mail
Address
Update
Price
Market
Online
Information
Work
Credit
Response
Offer
Transaction
Agreement
Registration
Person
System
Process
Service
Request
Message

3.3 Evaluation of Text Mining

The other approach we look into is evaluation of text mining methods, which was adopted by Pandey *et al.* [13]. In their work, several machine learning classifiers are implemented and t-test is conducted at 1 percent level of significance. In this research, 2500 e-mails are analyzed. Out of this, 1260 are phishing e-mails and 1240 mails are legitimate e-mails. The phishing e-mails are collected from the Phishing corpus and the legitimate e-mails are collected from the Spam Assassin. The unstructured data from the e-mails are collected and converted into appropriate structured format. The keywords which have highest frequency from the e-mails are selected as the features for the machine learning classifier. Rapid miner tool is used to identify the features from the e-mails. The features are represented in the Table 5.

Observations. Decision tree classifier and Logistic Regression classifier are implemented based on the details given in the research paper. Both of these machine learning classifiers are cross-validated 5 times with the collection of the data. In this section, the same set of experiments are going to be performed to find how the unbalanced nature of dataset influences the metrics - accuracy and precision, recall, F1 score.

$$accuracy(y,\hat{y}) = 1/n_{\text{samples}} \sum_{i=0}^{n_{\text{samples}}-1} l(\hat{y}_{\text{i}} = y_{\text{i}})$$

Figure 2: Accuracy score of Decision Tree and Logistic Regression classifiers

Table 6: Decision Tree classifier with parameter changes - F1 score

Phishing vs Legitimate	criterion: 'entropy'	class_weight ="balanced"	max_ features ="auto"	max_ features ="log2"
1:1	0.88498217	0.883621725	0.883128221	0.880783201
1:4	0.864220081	0.847473287	0.858477876	0.852182857
1:5	0.847254738	0.83202417	0.838730782	0.847160266

In the first experiment, we keep the size of the Phishing e-mails constant and size of the Legitimate e-mails is increasing in the each iteration and in total, we have performed 7 iterations. The results from this experiment is displayed in Figure 2 for both Decision Tree classifier and Logistic Regression classifier. The accuracy score of both the Decision Tree classifier and Logistic Regression classifier are steadily

Table 7: Logistic Regression classifier with parameter changes F1 score

class_weight = "balanced"	multi_class ='ovr'	multi_class = 'multinomial'
0.849576982	0.849576982	0.850472729
0.849052716	0.837377001	0.837235914
0.84139033	0.831566264	0.831597904

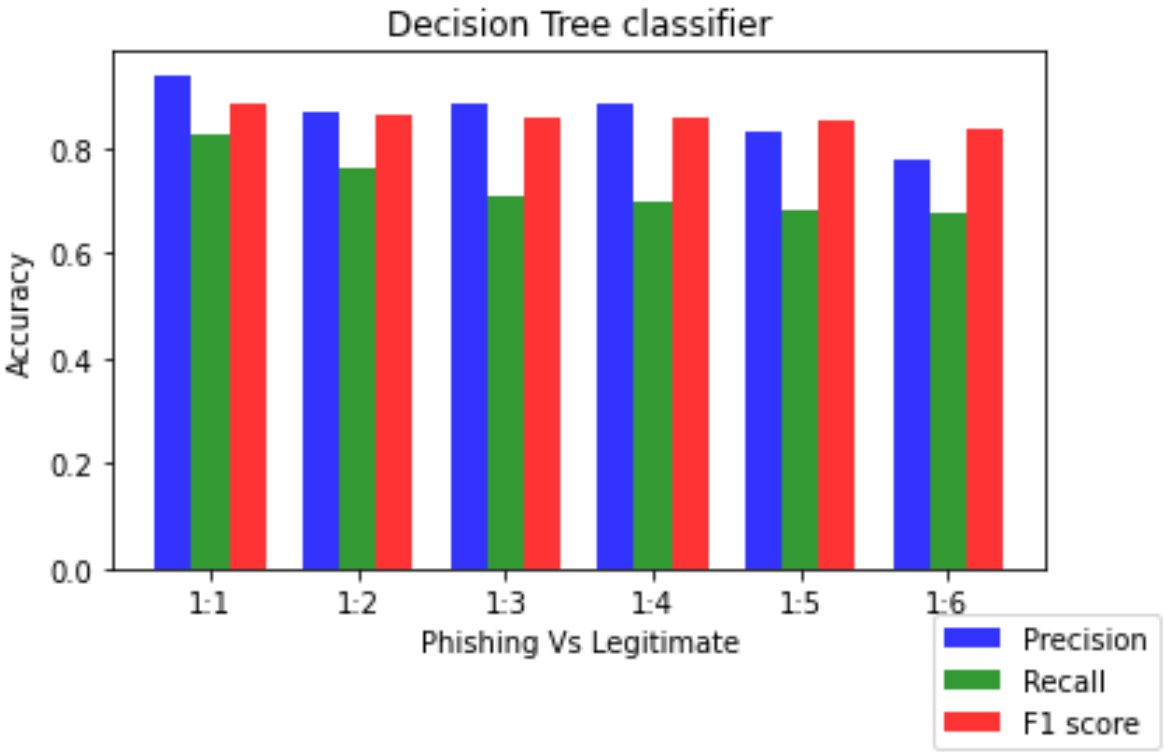


Figure 3: Decision Tree classifier Precision, Recall and F1 score

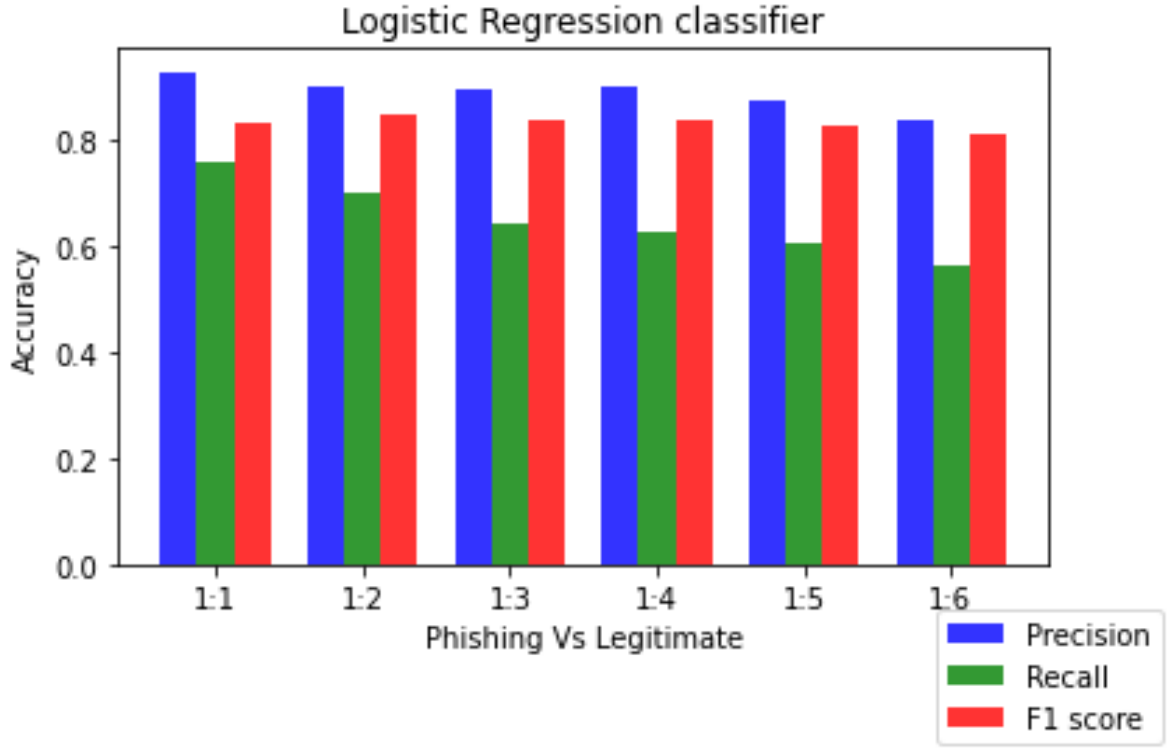


Figure 4: Logistic Regression classifier Precision, Recall and F1 score

increasing with the increase in size of the legitimate e-mails. The similar accuracy pattern which is observed in SVM classifier can be observed for that of Decision Tree classifier and Logistic Regression classifier. In the second experiment, metric - precision, recall and F1 score are calculated for each of the iteration. The results from these experiment is displayed in Figures 3 and 4 for Decision Tree classifier and Logistic Regression classifier. The scores of the Precision, Recall and F1 score are decreasing in each iteration and again it proves the behavioural changes of the machine learning classifier when dealing with unbalanced dataset. To measure the sensitivity of the different machine learning classifier, values of the parameters can be changed to see the impact on the trend of the metric - Precision, Recall and F1 score of the classifiers. The parameters like criterion, class_weight,max features are selected because these parameters influence the accuracy of decision tree classifier. We have selected the parameters criterion and max_features which are specific to the Decision Tree classifier. In general, the class weight parameter will adjust the weights between the classes based on the y values. The criterion 'entropy' is chosen, because it measures the level of impurity in a group of examples. This is used to determine which attribute in a given set of training feature vectors useful for discriminating between the phishing e-mails and legitimate e-mails. The class_ weight 'balanced' uses the value of y automatically adjust weights inversely proportional to class frequencies in the input data. The max_features is the number of features to consider when looking for the best split if "sqrt" max_features = sqrt(n_features) and if "log_2" max_features = log2(n_features). We have selected parameters like class_weight, multi_class. The class_ weight 'balanced' will adjust the weights between the classes based on the y values. The multi_class 'ovr' is used when the binary problem is fit for each label. The multi_class 'multinomial' can be also used for binary problem and the loss minimized is the multinomial loss fit across the entire probability distribution. By changing the values of the parameters, the experiment is repeated for three iterations. The results are displayed in the Tables 6 and 7. The F1 score of the Decision Tree classifier and Logistic Regression classifier are decreasing in each iteration. The changes in the parameters does not affect the pattern observed before. The same decreasing trend can also be seen with the Decision Tree and Logistic Regression classifiers again.

3.4 Supplementary methods

As mentioned, any machine learning based method can be supported by alternative methods that check the authenticity of the websites(URL links in the mails point to these websites) [3]. To check whether the e-mails lead to the malicious websites, one method is to perform reverse DNS lookup on the website to find the actual IP address of the website. Another method is to collect the blacklisted websites and maintain them as a common source of the reference, in the form of client-server architecture. The collection of the blacklisted websites in the server is the result of the addition of the malicious websites by the large group of clients. The problem inherent in this mechanism is that some hackers disguised as users are using this process of collecting the websites as a loophole to blacklist the valid websites. Another threat is that the websites created by the hackers will be identified as valid websites for the users who completely depend on this defense mechanism in case if they are not added to the blacklisted websites by any users. The mitigation mechanisms for these problems is out of the scope for this work.

4 Conclusion and Future Work

We evaluated two different approaches on phishing detection with a large dataset which we created out of two different e-mail corpuses. To do so, we synthesized e-mails from Enron dataset and a collection of phishing e-mails in `monkey.org` website. Through several experiments, we presented the results from the machine learning classifiers by training and testing them with the dataset we created. In the

set of experiments, we increased the size of the legitimate e-mails with keeping the size of phishing e-mails constant and measured different metrics like Accuracy, Precision, Recall and F1 score. We then performed the same experiment by keeping the legitimate e-mail size constant and increasing the number of phishing e-mails to investigate its effect on the accuracy.

Our experiments showed that increasing the size of legitimate e-mails in training sets while creating the model increases the accuracy but cannot sustain the precision, recall and F1 rates consistently in all ML algorithms we tested no matter if we use the structural properties or text mining methods. We also found that the ML classifiers trained with the unbalanced training sets as usual do not produce the sustained performance of the balanced datasets.

In our future work, we would like to expand our work to implement a combination of feature extraction methods while also considering black lists. We would also like to inspect the black list manipulation methods employed by possible actors for phishing attacks. Another direction is to investigate how phishing e-mails evolve over time and how the attackers take advantage of the important events and emergency situations going on in the world such as COVID-19 pandemic.

Acknowledgments

This work was funded by the College of Engineering and the Cybersecurity Center of the University of Massachusetts Dartmouth. Usual disclaimers apply. We would also like to thank Gandharva Dhanekula for her efforts on testing the dataset.

References

[1] S. Abu-Nimeh, D. Nappa, X. Wang, and S. Nair. A comparison of machine learning techniques for phishing detection. In *Proc. of the 2nd anti-phishing working groups annual eCrime researchers summit (eCrime'07), Pittsburgh, Pennsylvania, USA*, pages 60–69. ACM, October 2007.

[2] R. Basnet, S. Mukkamala, and A. H. Sung. *Detection of phishing attacks: A machine learning approach*, pages 373–383. Springer Berlin Heidelberg, 2008.

[3] M. Chandrasekaran, K. Narayanan, and S. Upadhyaya. Phishing e-mail detection based on structural properties. In *Proc. of 9th Annual NYS Cyber Security Conference, Albany, New York, USA*, volume 3, pages 2–8, June 2006.

[4] J. Clement. Number of e-mails per day worldwide 2017-2023. `https://web.archive.org/web/20200620201815/https://www.statista.com/statistics/456500/daily-number-of-e-mails-worldwide/` [Online; accessed on June 22, 2021], August 2019.

[5] R. Dhamija, J. D. Tygar, and M. Hearst. Why phishing works. In *Proc. of the 2006 SIGCHI Conference on Human Factors in Computing Systems (CHI'06), Montréal, Québec, Canada*, pages 581–590, April 2006.

[6] J. S. Downs, M. B. Holbrook, and L. F. Cranor. Decision strategies and susceptibility to phishing. In *Proc. of the 2nd Symposium on Usable Privacy and Security (SOUPS'06), Pittsburgh, Pennsylvania, USA*, pages 79—90. ACM, July 2006.

[7] I. Ghafir and V. Prenosil. Advanced persistent threat and spear phishing emails. In *Proc. of the 4th International Conference on Distance Learning, Simulation and Communication (DLSC'15), Brno, Czech Republic*, pages 34–41. IDET, May 2015.

[8] C. N. Gutierrez, T. Kim, R. Della Corte, J. Avery, D. Goldwasser, M. Cinque, and S. Bagchi. Learning from the ones that got away: Detecting new forms of phishing attacks. *IEEE Transactions on Dependable and Secure Computing*, 15(6):988–1001, August 2018.

[9] R. Islam and J. Abawajy. A multi-tier phishing detection and filtering approach. *Journal of Network and Computer Applications*, 36(1):324–335, January 2013.

[10] A. Karakasiliotis, S. Furnell, and M. Papadaki. Assessing end-user awareness of social engineering and phishing. In *Proc. of the 7th Australian Information Warfare and Security Conference (ISW'06), Perth, West-*

ern Australia, Australia, pages 1–11. School of Computer and Information Science, Edith Cowan University, Perth, 2006.

[11] G. Kul and S. Upadhyaya. A preliminary cyber ontology for insider threats in the financial sector. In *Proc. of the 7th ACM CCS International Workshop on Managing Insider Security Threats (CCS'15), Denver, Colrado, USA*, pages 75–78. ACM, October 2015.

[12] G. Kul, S. Upadhyaya, and A. Hughes. An analysis of complexity of insider attacks to databases. *ACM Transactions on Management Information Systems*, 12(1):1–18, December 2020.

[13] M. Pandey and V. Ravi. Detecting phishing e-mails using text and data mining. In *Proc. of the IEEE International Conference on Computational Intelligence and Computing Research (ICCIC'12), Coimbatore, India*, pages 1–6. IEEE, December 2012.

[14] S. Smadi, N. Aslam, and L. Zhang. Detection of online phishing email using dynamic evolving neural network based on reinforcement learning. *Decision Support Systems*, 107:88–102, March 2018.

[15] F. Toolan and J. Carthy. Feature selection for spam and phishing detection. In *Proc. of the 2010 eCrime Researchers Summit (eCrime'10), Dallas, Texas, USA*, pages 1–12. IEEE, October 2010.

[16] J. van Rijn. Email is not dead. but email is changing. `https://web.archive.org/web/20200512012109/https://www.emailisnotdead.com/` [Online; accessed on June 22, 2021], May 2020.

[17] R. Verma and N. Hossain. Semantic feature selection for text with application to phishing email detection. In *Proc. of the 16th International Conference on Information Security and Cryptology (ICISC'13), Seoul, Korea*, volume 8565 of *Lecture Notes in Computer Science*, pages 455–468. Springer, Cham, October 2014.

[18] M. Volkamer, K. Renaud, B. Reinheimer, and A. Kunz. User experiences of torpedo: Tooltip-powered phishing email detection. *Elsevier Computers & Security*, 71:100–113, 2017.

[19] J. Wang, Y. Li, and H. R. Rao. Overconfidence in phishing email detection. *Journal of the Association for Information Systems*, 17(11):759–783, November 2016.

Author Biography

Akash Sundararaj has received the B.E. degree from Velammal Engineering College in India and the M.S. degree in Computer and Information Science from the University of Massachusetts Dartmouth. Currently, he works as a Full Stack Software Developer in Ash Brokerage Corporation. He is interested in cybersecurity, algorithms, building machine learning models and creating web applications.

Gökhan Kul is an assistant professor at the Department of Computer and Information Science and the associate director of the Cybersecurity Center of the University of Massachusetts Dartmouth. He received his B.S. and M.S. degrees in Computer Engineering from TOBB University of Economics and Technology in 2010 and Middle East Technical University in 2012 in Turkey, respectively. He received his Ph.D. degree in 2018 at the University at Buffalo, SUNY. His research interests include cybersecurity, database systems, data engineering and security, and software engineering. He is a member of IEEE and ACM.