Detecting Phishing E-mail using Machine learning techniques
CEN-SecureNLP

Nidhin A Unnithan, Harikrishnan NB, Vinayakumar R, Soman KP
Center for Computational Engineering and Networking(CEN),
Amrita School of Engineering, Coimbatore
Amrita Vishwa Vidyapeetham, India
nidhinkittu5470@gmail.com

Sai Sundarakrishna
Caterpillar, Bangalore, India
sai.sundarakrishna@gmail.com

Abstract

The number of unsolicited aka phishing emails are increasing tremendously day by day. This suggests the need to design a reliable framework to filter out phishing emails. In the proposed work, we develop a supervised classifier for distinguishing phishing email from legitimate ones. The term frequency-inverse document frequency (tf-idf) matrix and Doc2Vec are formed for legitimate and phishing emails. This is passed to various traditional machine learning classifiers for classification. The machine learning classifiers with Doc2Vec representation have performed well in comparison to the tf-idf representation. Thus we conclude Doc2Vec representation is more appropriate for detecting and classifying phishing and legitimate emails.

1 Introduction

Electronic-mail (Email) is one of the most effective and easy source for transferring messages. It is considered as the safest message transfer over networks and is an inexpensive method. Even though there are many modes of message transfer, email popularity didn't reduced mostly in business, colleges and other private and government sectors as email is considered the safety transfer of message. Email communication plays an important part in everybody's life. Nowadays email usage gets a tremendous increase compared to olden days. There is a tremendous increase in users compared to 2016 in 2017. Nearly 4.8 billion persons are using email in 2017 and calculations shows that the number will rise to 5.6 billion users by 2021 over other apps [RH11]. But main problem with email has been phishing mails which causes malwares and are used in fraud schemes, advertisements etc. Considering previous years email phishing has increased recently and many security threats evolves and cause serious damages to business, individuals and economics. Especially for business emails extracting and analyzing these communication networks can reveal interesting and complex patterns of processes and decision making within a company. Detecting these fraud/phishing Emails precisely in communication networks is essential.

Phishing mails are type of spam mail which are hazardous to users. A phishing mail can steal our data without our knowledge once its opened. Thus identifying phishing mails from spam mails is very important. One way to protect our data from phishing mail is to add a secondary password to log in credentials. Another way is to alarm the user once a Phishing mail tries to steal our data.

During the infant stages of email communication,]

In: R. Verma, A. Das (eds.): Proceedings of the 1st AntiPhishing Shared Pilot at 4th ACM International Workshop on Security and Privacy Analytics (IWSPA 2018), Tempe, Arizona, USA, 21-03-2018, published at http://ceur-ws.org

clear rules was followed [SHP08], but recently due to the diversity of email programs and formatting standards we have the freedom to edit and change quoted text. Despite with these limitations, Symantec Brightmail Sanz [SHP08] has been showing good performance even now for detection of phishing emails. Moreover, it has the capability to keep track of IP (internet protocol) addresses of that sent phishing mail. The performance was comparable to [MW04]. Email services like Microsoft Outlook, Mozilla Thunderbird, or even online email communication such as Gmail, usually group emails into conversations and attempt to hide quoted parts in order to improve the readability.

In 2011 2.3 billion users were using emails which have increased to about 4.3 billion by 2016 [RH11]. TREC has defined phishing as an unwanted email sent discriminately [C+08]. Thus emails have been used for marketing and advertising purposes [CL98].

Datasets such as the Enron [KY04] or Avocado corpus [OWKG15] provide real world information about business communication and contains a mix of professional emails, personal emails, and phishing. [PS05] published parts of his personal email archive for research. A recent survey shows the diversity of email classification tasks alone [MSR+17]. Similarly another interesting analysis of communication networks based on metadata like sender, recipients, and time extracted from emails are discussed in [BCGJ11]. Models based on the written contents of emails may get confused by automatically inserted text blocks or quoted messages. Thus working with real world data requires normalization of data prior to solving the problem at hand. Rauscher et al. [RMA15] developed an approach to detect zones inside work-related emails where relevant business knowledge may be found. By ending overlapping text passages across the corpus, Jamison et al. managed to resolve email threads of the Enron corpus almost perfectly [JG13]. It has to be noted that the claimed accuracy of almost 100% was only tested on 20 email threats. In order to reassemble email threats, Yeh et al. considered a similar approach with a more elaborate evaluation reaching an accuracy of 98% separating email conversations into parts [YWD05]. To do so they rely on additional meta information in emails sent through Microsoft Outlook (thread index) and rules that match specific client headers. Thus, such an approach will not work on arbitrary emails nor can it handle different localization or edits by the user. Even though there are different ways to detect phishing [DAY+15] gives an overall evaluation of different classifiers used for phishing detection. Recently deep learning methods has also been used extensively for detecting phishing mails as stated in [BMS08] and for detecting malicious URLs and domains as stated in [VSP18b, VSP18a]. Domain Generation Algorithms which can be used by malicious families were also classified using deep learning methods as said in [VSPSK18].

In this task we propose a machine learning based approach to extract the underlying structure in email text to overcome problems of error-prone rule-based approaches. This will enable the downstream tasks to work with much cleaner data and additional information by focusing on particular parts. Also further we show the performance improvements and flexibility over the previous work on similar tasks.

Table 1: Training Dataset details

Category	Legitimate	Phishing	Total
With header	4082	501	4583
With no header	5088	612	5700

Table 2: Testing Dataset details

Category	Email Samples
With header	4195
With no header	4300

2 Background

This section discusses the mathematical details of various traditional machine learning algorithms and vector space modeling techniques such as tf-idf and Doc2Vec.

2.1 Term frequency-inverse document frequency (tf-idf)

Term frequency-inverse document frequency (tf-idf) can be used in information retrieval. It will reflect how much a word is important in a document or corpus. Tf-idf is also used for text mining and user modeling as a weighting factor. It will give less important to the words which are frequently repeated in a particular document. It is also used to remove stop words from a corpus. Nowadays the importance of tf-idf in search engine is very huge. Tf-idf can be calculated by the following equations

$$tf(t,d) = \frac{f_{t,d}}{\sum\limits_{t' \in d} f_{t',d}} \tag{1}$$

$$idf(t,D) = \log \frac{N}{|\{d \in D : t \in d\}|} \tag{2}$$

where N is the total number of documents in the corpus.

$$tfidf(t,d,D) = tf(t,d) \bullet idf(t,D) \tag{3}$$

Table 3: 10 fold cross validation accuracy of train data without header

Task	Representation	Algorithm	Accuracy
No Header	Doc2Vec	Decision Tree	81.2
No Header	Doc2Vec	Naive Bayes	79.5
No Header	Doc2Vec	Adaboost	83.4
No Header	Doc2Vec	Logistic Regresson	80.1
No Header	Doc2Vec	K-nearest neighbour	76.8
No Header	Doc2Vec	Support vector machine	88.4
No Header	Doc2Vec	Random Forest	87.4
No Header	tf-idf	Decision Tree	74.2
No Header	tf-idf	Naive Bayes	71.4
No Header	tf-idf	Adaboost	75.6
No Header	tf-idf	Logistic Regresson	70.2
No Header	tf-idf	K-nearest neighbour	63.2
No Header	tf-idf	Support vector machine	79.4
No Header	tf-idf	Random Forest	78.1

2.2 Doc2Vec

Doc2Vec is an unsupervised learning algorithm which gives a fixed length vector representation of a variable length text. The text can be a sentence, paragraph or a document. It is an extension of Word2Vec in which given a vector representation of context words as the input it predicts the word which is most likely to accompany the context words. Word2Vec is inspired because it can be used to predict the next word in a sentence given the context word vectors, thus capturing the semantics of the sentence even though the word vectors are randomly initialized. Instead of word vector we use document vector to predict next word given context from a document in Doc2Vec. In document vector every document is represented by a column of unique vector called document matrix and words are represented by unique vectors called word matrix. Next word in a context is predicted by the concatenation or averaging of document and word vectors.

In Doc2Vec the document vector is same for all context generated from same document but differs across documents. However word vector matrix is same for different document, i.e., the vector representation of same word across different document have the same vector representation.

2.3 Machine Learning

2.3.1 Decision Tree

In modern era, the most sensible discrete method plus a supervised algorithm personifying output in graphical format is decision trees. It's an algorithm where each element in the given domain is put to an element of its range which could be either discrete or continuous. It's better for class type variables. In this procedure, each split is chosen in such a way that it reduces the target variable's variance. The Decision tree input is often passed as an object or scenario which imitates some set of properties and output is usually a decision saying either YES or NO.

Trees are built using leaves. On every node of the tree a test is conducted which looks for the least possible outcome. The leaves subsist of numerical or categorical values, of the respective item, which is the outcome after each test.

2.3.2 Naive Bayes

This uses Bayes theorem. It is the most singular feature with independence i.e. coordinates present for any feature dependability in a class doesn't affect other features. Naive Bayes Classifier model is prone to outperform when the feature dimension is high and is easy to build. Though it outperforms most of the time when the condition of independence is matched, its independence does not overcomes the problems related to dimensionality. It utilizes conditional probability model i.e. when a problem is posed which needs to be classified and imitates a vector $X = (x_1, x_2, ...x_n)$ which epitomizes features yielding probabilities $P(C_k/(x_1, x2, ...xn))$ for k outcomes. Mathematically it can be expressed as

$$P(C_k/x) = \frac{P(C_k P(x|C_k}{P(x)} \tag{4}$$

2.3.3 AdaBoost

It is a continuous learning algorithm whose main purpose lies in stepping up the achievement of the learning algorithm. It is solemnly used for classification. It performs this task by forming a strong classifier which is a sequence of innumerable weak classifiers. When Ada boost is combined with Decision tress it is best-out-of the box classifier. Irrespective of its swiftness

Table 4: 10 fold cross validation accuracy of train data with header

Task	Representation	Algorithm	Accuracy
With Header	Doc2Vec	Decision Tree	73.1
With Header	Doc2Vec	Naive Bayes	70.1
With Header	Doc2Vec	Adaboost	77.4
With Header	Doc2Vec	Logistic Regresson	72.2
With Header	Doc2Vec	K-nearest neighbour	69.1
With Header	Doc2Vec	Support vector machine	75.4
With Header	Doc2Vec	Random Forest	73.4
With Header	tf-idf	Decision Tree	68.2
With Header	tf-idf	Naive Bayes	64.2
With Header	tf-idf	Adaboost	69.4
With Header	tf-idf	Logistic Regresson	66.7
With Header	tf-idf	K-nearest neighbour	62.2
With Header	tf-idf	Support vector machine	72.4
With Header	tf-idf	Random Forest	71.2

Table 5: Test Data result for SVM combined with Doc2Vec

Task	TP	TN	FP	FN
No Header	3825	0	475	0
With Header	3593	7	489	106

in classifying it has been used as a feature learner as well.

2.3.4 Logistic Regression

It is used when target variable is categorized. It hinges on MLE (Maximum Likelihood Estimation) and is a qualitative choice model. It is used to predict whether the risk factor increases the odds of a given outcome by a specific factor. Logistic Regression can be used to model binary classification problems. The mathematical representation is given as

$$F(x) = \frac{1}{1 + exp(-w^T x)} \tag{5}$$

where F can take values in the range 0 to 1.

2.3.5 k-nearest neighbour (KNN)

It is the simplest algorithm of machine learning. It is known as lazy learning because it furnishes only approximate values. It is flubbed by local structure of the data. This procedure validates the local posterior probability of each class existing by the average of class membership over its K-nearest neighbors.

2.3.6 Support Vector Machine (SVM)

Support Vector Machine (SVM) is a linear classifier algorithm based on supervised learning. It helps to create a boundary between the variables to classify them. It creates a hyper plane boundary with maximum margin to separate the variables. This algorithm is robust to outliers. The co-ordinates of individual observations are called as support vectors. SVM creates a hyperplane separating support vectors with the maximum possible margin.

Support Vector Machines is one of the popularly used method in supervised machine learning techniques. Problems like linear regression and classification tasks could be solved easily with it. Here the training set is separated by a hyperplane where the points nearer to the hyperplane are support vectors which aid them in finding the position of hyper-plane. In case training data set couldn't be linearly separated, it is mapped to a high-dimensional space where it is assumed to be linearly separable.

2.3.7 Random Forest

Random Forest is a supervised learning algorithm used in both classification and regression problems. In the random forest classifier, to get high accuracy results we need to create large number of decision trees. The prediction obtained from a Random Forest is prone to be far better than the predictions obtained by an individual decision tree. Random Forest utilizes the concept of bagging for creating several minimal correlated decision trees. Advantages of Random forest is its ability to handle missing values and to avoid overfitting of the model.

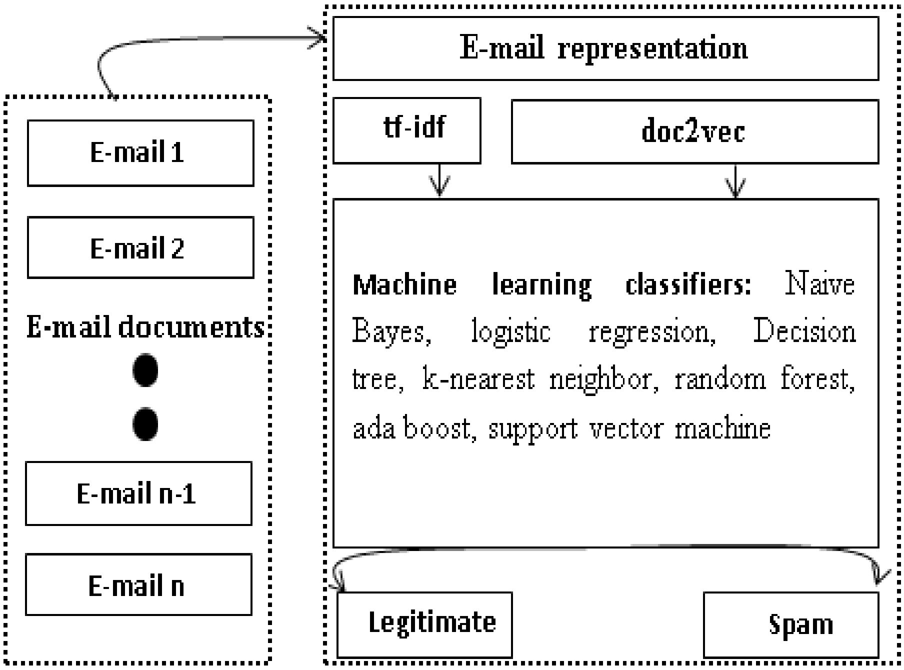
E-mail 1
E-mail 2
E-mail documents
E-mail n-1
E-mail n
E-mail representation
tf-idf
doc2vec
Machine learning classifiers: Naive Bayes, logistic regression, Decision tree, k-nearest neighbor, random forest, ada boost, support vector machine
Legitimate
Spam

Figure 1: Proposed architecture for email phishing detection

3 Experiments

3.1 Description of data set

The anti-phishing shared task is a part of First Security and Privacy Analytics Anti-Phishing Shared Task (IWSPA-AP 2018) at 4th ACM International Workshop on Security and Privacy Analytics [EDMB+18][EDB+18]. Let $E = [e_1, e_2, ...e_n]$ be a set of emails and $C = [c_1, c_2, c_3, ...c_n]$ be a set of email types such as legitimate or phishing. The task is to classify each given email sample into either legitimate or phishing. The detailed summary of training and testing data set is summarized in Table 1 and Table 2.

3.2 Proposed Architecture

In our proposed architecture we used count based and distributed representation for word representation. In count based method we used tf-idf for word representation and for distributed representation we used Doc2Vec using gensim library. Once the word representations were created we used different machine learning techniques to classify the data as legitimate or phishing. The machine learning techniques used are Naive Bayse, Logistic Regression, Decision Tree, K Nearest Neighbour, Random Forest, Adaboost and Support Vector Machine.

4 Results

Our model was trained for seven different machine learning techniques for two different representations, i.e., with and without header data sets. All the results have been consolidated in Table 3 and Table 4. Out of all the different models the one in which SVM combined with Doc2Vec gave the highest accuracy for both the data sets, thus only that model was given for submission even though we trained for seven different techniques. The submitted models were tested using test data and the result for True Positive, True Negative, False Positive, False Negative are consolidated into Table 5.

5 Conclusion

The main objective of this work is to develop a supervised classifier which can detect phishing and legitimate emails. We used count based and distributed representations for our word representation and used different machine learning techniques such as Naive Bayse, Logistic Regression, Decision Tree, K Nearest Neighbour, Random Forest, Adaboost and Support Vector Machine for classification of legitimate and phishing emails. The proposed methodology rely on feature engineering and in future we can apply deep learning on the phishing detection and can be considered as one in the future direction.

Acknowledgements

This research was supported in part by Paramount Computer Systems. We are grateful to NVIDIA India, for the GPU hardware support to the research grant. We are grateful to Computational Engineering and Networking (CEN) department for encouraging the research.

References

[BCGJ11] Francesco Bonchi, Carlos Castillo, Aristides Gionis, and Alejandro Jaimes. Social network analysis and mining for business applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):22, 2011.

[BMS08] Ram Basnet, Srinivas Mukkamala, and Andrew H Sung. Detection of phishing attacks: A machine learning approach. In *Soft Computing Applications in Industry*, pages 373–383. Springer, 2008.

[C⁺08] Gordon V Cormack et al. Email spam filtering: A systematic review. *Foundations and Trends® in Information Retrieval*, 1(4):335–455, 2008.

[CL98] Lorrie Faith Cranor and Brian A LaMacchia. Spam! *Communications of the ACM*, 41(8):74–83, 1998.

[DAY⁺15] Ammar Yahya Daeef, R Badlishah Ahmad, Yasmin Yacob, Naimah Yaakob, and Mohd Nazri Bin Mohd Warip. Phishing email classifiers evaluation: Email body and header approach. *Journal of Theoretical and Applied Information Technology*, 80(2):354, 2015.

[EDB⁺18] Ayman Elaassal, Avisha Das, Shahryar Baki, Luis De Moraes, and Rakesh Verma. Iwspa-ap: Anti-phising shared task at acm international workshop on security and privacy analytics. In *Proceedings of the 1st IWSPA Anti-Phishing Shared Task*. CEUR, 2018.

[EDMB⁺18] Ayman Elaassal, Luis De Moraes, Shahryar Baki, Rakesh Verma, and Avisha Das. Iwspa-ap shared task email dataset, 2018.

[JG13] Emily Jamison and Iryna Gurevych. Headerless, quoteless, but not hopeless? using pairwise email classification to disentangle email threads. In *Proceedings of the International Conference Recent Advances in Natural Language Processing RANLP 2013*, pages 327–335, 2013.

[KY04] Bryan Klimt and Yiming Yang. The enron corpus: A new dataset for email classification research. In *European Conference on Machine Learning*, pages 217–226. Springer, 2004.

[MSR⁺17] Ghulam Mujtaba, Liyana Shuib, Ram Gopal Raj, Nahdia Majeed, and Mohammed Ali Al-Garadi. Email classification research trends: Review and open issues. *IEEE Access*, 5:9044–9064, 2017.

[MW04] Tony A Meyer and Brendon Whateley. Spambayes: Effective open-source, bayesian based, email classification system. In *CEAS*. Citeseer, 2004.

[OWKG15] Douglas Oard, William Webber, David Kirsch, and Sergey Golitsynskiy. Avocado research email collection. *Philadelphia: Linguistic Data Consortium*, 2015.

[PS05] Adam Perer and Ben Shneiderman. Beyond threads: Identifying discussions in email archives. Technical report, MARYLAND UNIV COLLEGE PARK HUMAN COMPUTER INTERACTION LAB, 2005.

[RH11] Sara Radicati and Quoc Hoang. Email statistics report, 2011-2015. *Retrieved May*, 25:2011, 2011.

[RMA15] François Rauscher, Nada Matta, and Hassan Atifi. Context aware knowledge zoning: Traceability and business emails. In *IFIP International Workshop on Artificial Intelligence for Knowledge Management*, pages 66–79. Springer, 2015.

[SHP08] Enrique Puertas Sanz, José María Gómez Hidalgo, and José Carlos Cortizo Pérez. Email spam filtering. *Advances in computers*, 74:45–114, 2008.

[VSP18a] R Vinayakumar, KP Soman, and Prabaharan Poornachandran. Detecting malicious domain names using deep learning approaches at scale. *Journal of Intelligent & Fuzzy Systems*, 34(3):1355–1367, 2018.

[VSP18b] R Vinayakumar, KP Soman, and Prabaharan Poornachandran. Evaluating deep learning approaches to characterize and classify malicious urls. *Journal of Intelligent & Fuzzy Systems*, 34(3):1333–1343, 2018.

[VSPSK18] R Vinayakumar, KP Soman, Prabaharan Poornachandran, and S Sachin Kumar. Evaluating deep learning approaches to characterize and classify the dgas at scale. *Journal of Intelligent & Fuzzy Systems*, 34(3):1265–1276, 2018.

[YWD05] Chi-Yuan Yeh, Chili-Hung Wu, and Shine-Hwang Doong. Effective spam classification based on meta-heuristics. In *Systems, Man and Cybernetics, 2005 IEEE International Conference on*, volume 4, pages 3872–3877. IEEE, 2005.