Machine Learning Based Phishing E-mail detection

Security-CEN@Amrita

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

Abstract

Phishing email detection is a significant threat in today's world. The rate at which phishing are generated are tremendously increasing day by day. It is high time to deploy a self-learning system that gives a time bound detection and prevention of phishing email efficiently. This work proposes a system which uses term document matrix as feature engineering mechanism and classical machine learning techniques for detecting phishing email from legitimate and phishing ones. The system also incorporates the domain knowledge and lexical features as part of feature engineering mechanism. The efficiency of the system is compared using different classical machine learning techniques. Based on the accuracy, we propose the best model that solves the formulated problem efficiently.

1 Introduction

Email plays an important part of everybody's life. It is one of the easiest and effective source for transferring messages and files. Even though there are many modes of communication, the popularity of e-mail did not diminish as it is considered as one of the safest and fastest message transfer over networks and is an inexpensive method of communication.

Nowadays e-mail usage gets a tremendous increase compared to previous decades. In 2017 there were nearly 4.8 billion persons using email and it is estimated that by 2021 there will be an increase to 5.6 billion users as email is considered to be main medium of transfer for messages over other apps. But main problem in email is the presence of phishing mails. These phishing mails are unwanted mails which may carry malwares, fraud schemes, advertisements etc. In comparison to previous years, phishing mails have increased and have caused serious damages to business, corporates, individuals and economics. Detecting the fraud/phishing emails precisely is essential, extracting and analyzing these mails can reveal us complex and interesting patterns and we can make appropriate decisions within a company to block phishing mails. During the early stages of communication via email clear rules were followed. But nowadays due to diversity present in email services, like Microsoft Outlook, Mozilla Thunderbird, Google's Gmail, mails are grouped into conversations and attempts to hide quoted parts in order to improve the readability.

One type of spam mail which is hazardous to users is phishing mails. A phishing mail is the one which covers itself as a legitimate mail but once opened can steal our data without our knowledge. 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.

In [SAZ+15] Sami S et.al proposed a model for detecting phishing emails that rely on a preprocessing technique which extracts different part of email as feature. And this extracted feature is fed into a j48 classification algorithm to perform classification. In [SZL+15], they considered meaningless tokens and new pages as the feature set. Authors in [SZL+15], selected some features that have better predictability from initial feature set. They provide the O(1) complexity as an evaluation method to each feature set to evaluate

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

its predictive ability. In the paper [KK15], sukhjeel kaui et.al used Genetic algorithm for the detection of phishing webpage and for categorizing pages they preferred a filter function. Lu fang et.al in [FBJ+15] proposes some solution to overcome the time lag in detecting phishing websites. Here they provide a solution to detect phishing websites by analyzing the peculiarity in its WHOIS and URL information. In [VSP18b, VSP18a] deep learning methods were employed to detect malicious URL′s and domains. Binay kumar et.al has used html contents for detecting email phishing in [KKMK15]. But Rachna Dhamija et.al in [TC09] mainly concentrated in this topic to know which phishing activity works during the attack and why. For that they used a large given set of data which contains reported phishing activities. Fergus toolan et.al made a different approach. They used only five features for classification. For classification they used a C5.0 algorithm which have more precision compare to other algorithms. Mayank pandey et.al in [PR12] used different types of classification methods such as Multilayer Perceptron (MLP), Decision Trees (DT), Support Vector Machine (SVM), Group Method of Data Handling (GMDH), Probabilistic Neural Net (PNN), Genetic Programming (GP) and Logistic Regression (LR). Lew may form et.al in [FCT+15] proposed a method which uses hybrid features for detecting phishing emails. It is called Hybrid features because it is a combination of URL based, behavior based and contend based features. Here they acquired an overall accuracy of 97.25 % with an error percentage of 2.75 %.

Even though there are different ways to detect phishing, [DAY+15] gives an overall evaluation of different classifiers used for phishing detection. Recently count based representation combined with domain level features integrated with machine learning techniques are used for classifying phishing mails and legitimate mails [EDB+18, BMS08]. The proposed methodology uses feature engineering approach combined with deep learning, which is one the significant direction in which world is moving to because it has performed well in most of the text classification tasks [LBH15] and even in phishing detection [LNRW, EC].

The rest of the sections are organized as follows. Section 2 discusses the background details of email representation and the machine learning algorithms. Section 3 includes the description of data set, experiments and proposed architecture. Section 4 includes results. Conclusion is placed in Section 5.

2 Background

This section discusses the mathematical details of various traditional machine learning algorithms and details of vector space modeling techniques such as TF-IDF and Bag of words.

2.1 Logistic Regression

This is a classification algorithm which is used to separate the data into different classes. This can be normal, ordinary and multinominal. In binary Logistic Regression the outcome or the classification can be done into 0 and 1 whereas in multinominal the outcome or classification will be in multiple ways. The activation function used for performing this is sigmoid function. The mathematical representation of sigmoid activation function is as follows:

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

2.2 Naive Bayes

Naive Bayes is a set of supervised learning algorithm which works on the principle of Bayes theorem. This theorem works on conditional probability by which probability of the events is calculated. Binary and multiple classification are done by using different types of algorithms like GaussianNB, MultinomialNB, BernoulliNB [MN+98]. Here for this problem we used MultinominalNB from scikit-learn as our algorithm.

2.3 Support Vector Machine

SVM is a supervised classification algorithm which builds the model by classifying the data into two classes. Based on the number of classes we will be defining the SVM. It is of two types linear SVM and non-linear SVM. The decision boundary for linear SVM is formulated as a hyperplane in feature space, i.e. a linear function of the features. Non-linear SVMs result in non-linear decision boundaries in the original feature space. From different types of kernals available we used radial basis function (RBF) for our SVM model.

2.4 TF-IDF

TF-IDF stands for term frequency-inverse document frequency and its weight can be considered as a statistical measure which evaluates how important a word is to a document which can in turn be used for information retrieval and text mining. Term Frequency gives us an idea about how frequently a term occurs in a document. This can be mathematically defined as equation given below

$$tf(t,d) = \frac{f_{t,d}}{\sum\limits_{t' \in d} f_{t',d}} \tag{2}$$

Inverse Document Frequency gives us an idea about how important a term is. When we compute term frequency all the terms are given equal importance whether it is a stop word or a terminology word. Thus we need to weigh up terminology word which is less frequent than the stop word in a document by computing inverse document frequency given by mathematical equation

$$idf(t, D) = \log \frac{N}{|\{d \in D : t \in d\}|} \quad (3)$$

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

Now TF-IDF can be calculated as

$$tfidf(t, d, D) = tf(t, d) \bullet idf(t, D) \quad (4)$$

Additionally the domain level features are added. This includes a list of most commonly appeared words and a list of special characters.

3 Experiments

3.1 Dataset details

The email phishing detection is a task in shared task on anti-phishing shared task at 4th ACM International Workshop on Security and Privacy Analytics [EDMB+18]. Let E = [e_1,e_2,...,e_n] and C = [c_1,c_2,...,c_n] be sets of email types such as legitimate or phishing, the task was to classify each given email samples into either legitimate or phishing. Two sets of data sets were used one with header and one without header. Data set statistics are integrated together in Table 1 for training and Table 2 for testing.

Table 1: Training Dataset details

Training Dataset	**Legitimate**	**Phishing**	**Total**
With header	4082	501	4583
Without header	5088	612	5700

Table 2: Testing Dataset details

Testing Dataset	**Data Samples**
With header	4195
Without header	4300

3.2 Proposed Architecture

We used count based representation to create our model. A diagrammatic representation of our architecture is shown in Figure 1. The email samples from data set is first passed through count based representation, here TF-IDF, for word representation. It is then combined with domain level features to get our input word representation for machine learning algorithms. The domain level features include most commonly appeared words (40 features), for example password, fraudulent, business, and special characters like $, #, !, (, [, &, etc. and all the stop words were removed. These are then passed through Logistic Regression, Naive Bayes and Support Vector Machine to do the classification of phishing and legitimate mails.

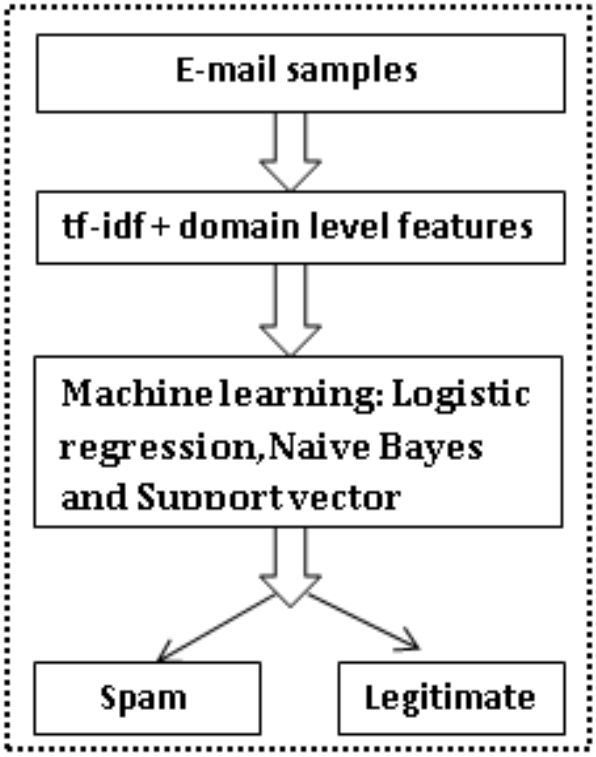

Figure 1: Proposed Architecture

Table 3: Statistics of 10-fold cross validation

Method	**Task**	**Accuracy**
Logistic Regression	Without Header	92.2
Naive Bayes	Without Header	93.4
Support Vector Machine	Without Header	94.3
Logistic Regression	With Header	91.2
Naive Bayes	With Header	92.2
Support Vector Machine	With Header	93.3

4 Results

Our model build using above architecture was trained for data sets with headers and without headers for classification of phishing and legitimate mails. We trained a total of six models, one each for Logistic Regression, Naive Bayes, Support Vector Machine for mails with

Table 4: Statistics of Test Result

Method	Task	TP	TN	FP	FN	Accuracy	Precision	Recall	F1 score
Logistic Regression	Without Header	3784	325	150	41	0.95	0.96	0.98	0.97
Naive Bayes	Without Header	3807	258	217	18	0.94	0.94	0.99	0.97
Support Vector Machine	Without Header	3671	337	138	154	0.93	0.96	0.95	0.96
Logistic Regression	With Header	3612	490	6	87	0.97	0.99	0.97	0.98
Naive Bayes	With Header	3572	489	7	127	0.96	0.99	0.96	0.98
Support Vector Machine	With Header	3561	458	38	138	0.95	0.98	0.96	0.97

header and without header. We used 10 fold cross validation for our training data and the results obtained by our model has been consolidated in Table 3. For data set without headers SVM gave the highest accuracy with 94.3% and for data set with headers SVM gave the highest accuracy with 93.3%. We didn't extract any features from header data set but extracting features from headers may increase the accuracy. Our model was tested using test data by IWSPA-AP Shared Task committee and the corresponding results for True Positive, True Negative, False Positive, False Negative, Accuracy, Precision, Recall, F1 score for our six models are summarized in Table 4.

5 Conclusion

This paper evaluated the performance of machine learning based classifier for distinguishing phishing emails from legitimate ones. We created a model using count based representation combined with domain level features as word representation and passed to various machine learning techniques such as Logistic Regression, Naive Bayes and Support Vector Machine to classify whether it is phishing or legitimate. Both the sub tasks belong to unconstrained category, i.e., any data sets can be used during training and data sets for both the tasks where highly imbalanced. Even then we have not used any other external data set sources and still were able to achieve good detection rate for phishing email in both sub tasks. By adding some additional data sources we can considerable increase the detection rate of phishing emails for the proposed methodology.

5.0.1 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

[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.

[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.

[EC] Louis Eugene and Isaac Caswell. Making a manageable email experience with deep learning.

[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.

[FBJ+15] Lv Fang, Wang Bailing, Huang Junheng, Sun Yushan, and Wei Yuliang. A proactive discovery and filtering solution on phishing websites. In *Big Data (Big Data), 2015 IEEE International Conference on*, pages 2348–2355. IEEE, 2015.

[FCT+15] Lew May Form, Kang Leng Chiew, Wei King Tiong, et al. Phishing email detection technique by using hybrid features. In *IT in Asia (CITA), 2015 9th International Conference on*, pages 1–5. IEEE, 2015.

[KK15] Sukhjeel Kaui and Amrit Kaur. Detection of phishing webpages using weights computed through genetic algorithm. In *MOOCs, Innovation and Technology in Education (MITE), 2015 IEEE 3rd International Conference on*, pages 331–336. IEEE, 2015.

[KKMK15] Binay Kumar, Pankaj Kumar, Ankit Mundra, and Shikha Kabra. Dc scanner: Detecting phishing attack. In *Image Information Processing (ICIIP), 2015 Third International Conference on*, pages 271–276. IEEE, 2015.

[LBH15] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436, 2015.

[LNRW] Christopher Lennan, Bastian Naber, Jan Reher, and Leon Weber. End-to-end spam classification with neural networks.

[MN+98] Andrew McCallum, Kamal Nigam, et al. A comparison of event models for naive bayes text classification. In *AAAI-98 workshop on learning for text categorization*, volume 752, pages 41–48. Citeseer, 1998.

[PR12] Mayank Pandey and Vadlamani Ravi. Detecting phishing e-mails using text and data mining. In *Computational Intelligence & Computing Research (ICCIC), 2012 IEEE International Conference on*, pages 1–6. IEEE, 2012.

[SAZ+15] Sami Smadi, Nauman Aslam, Li Zhang, Rafe Alasem, and MA Hossain. Detection of phishing emails using data mining algorithms. In *Software, Knowledge, Information Management and Applications (SKIMA), 2015 9th International Conference on*, pages 1–8. IEEE, 2015.

[SZL+15] Hongzhou Sha, Zhou Zhou, Qingyun Liu, Tingwen Liu, and Chao Zheng. Limited dictionary builder: An approach to select representative tokens for malicious urls detection. In *Communications (ICC), 2015 IEEE International Conference on*, pages 7077–7082. IEEE, 2015.

[TC09] Fergus Toolan and Joe Carthy. Phishing detection using classifier ensembles. In *eCrime Researchers Summit, 2009. eCRIME'09.*, pages 1–9. IEEE, 2009.

[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.