PED-ML: Phishing Email Detection Using Classical Machine Learning Techniques

CENSec@Amrita

Anu Vazhayil, Harikrishnan NB, Vinayakumar R, Soman KP
Center for Computational Engineering and Networking(CEN),
Amrita School of Engineering, Coimbatore
Amrita Vishwa Vidyapeetham, India
anuv.1994@gmail.com

Abstract

In the modern era, all services are maintained online and everyone use it to speed up their day to day activities. This include social as well as financial activities which involves usage of sensitive information to carry out the intended task. With the increase in usage of such facilities put forth the importance of securing the data used to perform such actions. Over the last decade phishing has become a serious threat to the society by stealing sensitive information to get hold of these facilities. This is considered to be the most profitable cybercrime and according to IBMs X-Force researchers statistics, the number of people becoming the victim of such activities are increasing tremendously. As the risk of phishing emails are increasing steadily, the need to detect and overcome such situations stands as one of the highest priority task at hand. In the present work, we will use non-sequential representation such as term document matrix approach followed by Singular Value Decomposition (SVD) and Nonnegative Matrix Factorization (NMF) to model phishing email detection as a supervised classification problem to detect phishing emails from legitimate ones.

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

1 Introduction

The growth of internet has revolutionized the digital era. This revolution has changed entirely the way we communicate, carry out business, advertisement etc. In fact, in today's world in order to establish a successful business a web presence is mandatory. And in all cases important communications takes place through email. At the same time there are instances where phishing emails are send to users and the main goal of such emails is to steal sensitive information of the user. Phishing emails does this by sending emails claiming to originate from some trusted sources. And these emails contain links or attachments which tries to get sensitive information from the user. In such a scenario an efficient mechanism that detects and classify phishing emails has to be addressed. The conventional techniques used are blacklisting, greylisting and whitelisting. In the case of blacklisting, IP and email address of those mails which attempts to collect the private information of users are stored in a list and all emails arrived from the email address specified in the list are marked as phishing scams. Whitelist functions exactly opposite to blacklist by allowing emails from trusted users specified in the whitelist. The drawback of these methods is the requirement of human involvement in defining and updating the list and it also fails at detecting the new or the variants of existing phishing email. The other popular method include Bayesian filters, a heuristic approach. Bayesian filters are popularly used detection techniques during 1990s. With the increase in the computational capability, there is a paradigm shift from conventional techniques to data driven techniques. Data driven techniques popularized the impact of machine learning in the area of cyber security [NVK+15] in unfathomable ways.

Table 1: Training email corpus details

Training Dataset	Legitimate	Phish	Total
No header	5088	612	5700
With header	4082	501	4583

Table 2: Testing email corpus details

Testing Dataset	Total
No header	4300
With header	4195

There has been significant amount of research going on in the direction of phishing email classification. Researchers have come up with many mathematical models to detect phishing emails. Some of the commonly used techniques are naive bayes classifier, boosted decision tree [CM01], SVM [DWV99a], LVQ-based neural network [CXMX05] etc. These methods needs a Bayesian prior knowledge about the nature of phishing emails [SVKS15], [BVP].

Recent trends in the field of computer vision and Natural Language Processing (NLP), clearly conveys the potential use of machine learning techniques to tackle many significant problems in these areas. In such a situation our research mainly focus on machine learning based solution to classify emails as either phishing or legitimate. In this paper the authors used Term Document Matrix (TDM) for non sequential representation of the corpus. Feature engineering is an important step in all machine learning tasks. In order to extract the important features SVD and NMF is applied on the data. These are then passed to machine learning algorithms like Decision tree, K-NN, Naive Bayes, Random forest, SVM and logistic regression.

The remaining part of the paper is arranged as follows: Section 2 represents related works, Section 3 discusses the model altogether, covering dataset description, representation of the data and highlights the methodology used, Section 4 and 5 represents results and conclusion respectively followed by acknowledgement.

2 Related Works

Phishing attacks are serious cyber threats for both multinational companies as well as users. These emails seems like they are legitimate but contains malicious contents which can steal important information like bank account number, credit card details etc, and bring huge loss to individuals and organizations. This calls the importance of segregating such emails. Methods like blacklisting requires human intervention to manually select and classify the emails. While on the other hand there are feature engineering techniques which analyses the contents of emails and helps in the classification process. In [SDHH98], the work has conveyed the importance of phishing specific features for classification. In [KMAH04] the classfication error was reduced by utilizing the temporal relation in email sequence and using those as features. Heuristics based feature selection was highlighted in [MW04]. Due to the growth of computing facilities, data driven methods were widely used in email classification. In [Faw03] and [Gee03] data mining techniques were introduced for filtering non-legitimate emails. Also [DWV99b] used PCA as a pre processing technique for extracting features as well as for dimensionality reduction. Authors in [ANNWN07] has used machine learning based models like logistic regression, SVM and random forest for classifying emails as either phishing or legitimate. In this work we make use of the importance of dimensionality reduction and TDM representation of data. For dimensionality reduction we use SVD and NMF. The representation is then followed by application of classical machine learning techniques on the processed data.

3 Proposed Architecture

The proposed architecture for an anti-phishing framework to detect phishing emails from legitimate ones is explained using a flow chart in Figure 1. The same model is used in both the cases where the data contains emails with and without header. Detailed explanation of all the levels are given below.

3.1 Dataset description

As part of the anti-phishing shared task at first security and privacy analytics(IWSPA-AP 2018) two subtasks were held. Task 1 is classifying Email with headers and Task 2 is Email with no headers. The dataset details [EDMB+18], [EDB+18] is provided in Tables 1 and 2 above.

3.2 Dataset representation

Data representation is considered to be the most important part in any machine learning task and need to be chosen properly depending on the nature of the dataset. The corpus received for the shared task contains text and special symbols. So, the first step is to produce meaningful representation of the data. In this work, for all the experiments TDM is used for the

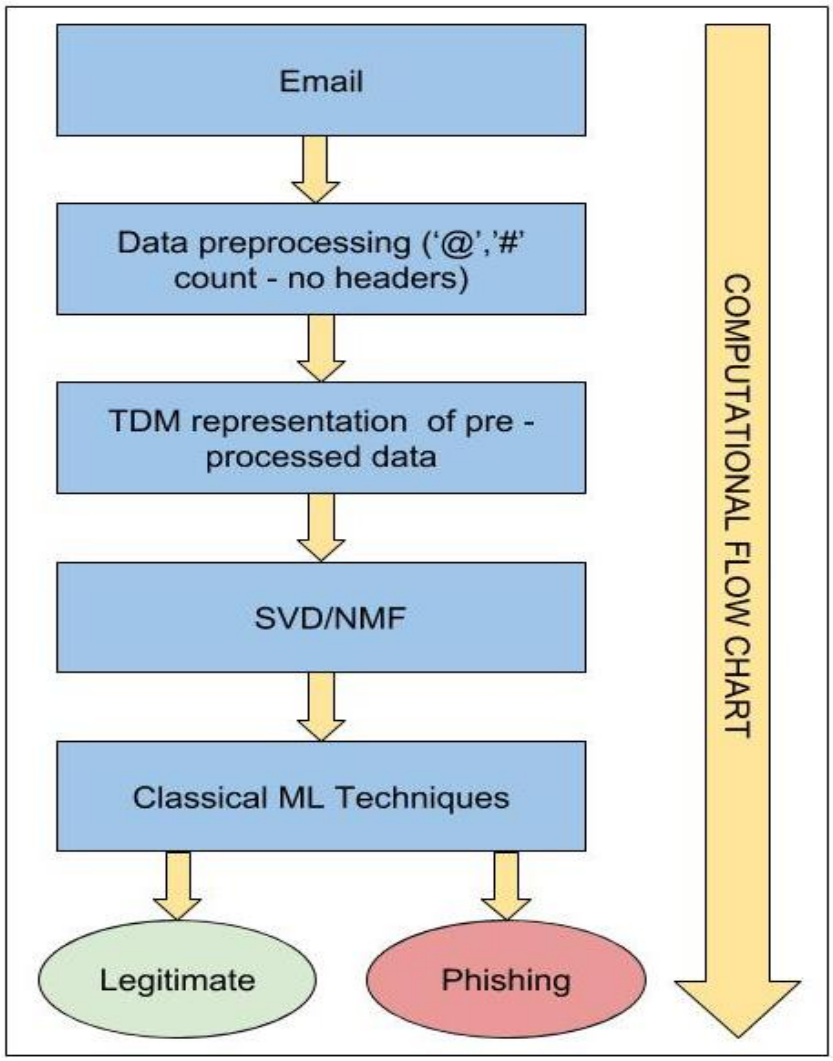

Figure 1: Proposed architecture for phishing email detection

numerical representation of the data for both the subtasks given. After doing the representation the second step involves feature extraction and dimensionality reduction. This is carried out using Singular Value Decomposition (SVD) and Non-negative Matrix factorization (NMF) methods. For this, the TDM is passed to the feature extraction block. In the feature extraction block, the rank is taken as 30 for all the cases which means, the number of columns of the train and test data matrix will be taken as 30 after doing the dimensionality reduction. This numeric representation of the data is then passed to all the different machine learning algorithms for classificaiton. Figure 1 describes the steps involved in the proposed architecture. The proposed architecture consists of 5 blocks. Block 1 represents the raw dataset ie. the set of emails with and without headers. In block 2 the data is preprocessed by removing the special characters and unnecessary details from the raw data. Block 3 represents the process of data representation of the emails. The data representation is followed by dimensionality reduction block where SVD and NMF technniques are applied to the input from block 3. This is passed to block 5 where different classical machine learning algorithms are incorporated. Finally the emails are classified as either legitimate or phishing. The mathematical formulation of the task is as follows:

- Given a set of emails represented as $D = [e_1, e_2, ...e_n]$ and its classes like $C = [c_1, c_2, ...c_n]$. The class values are either 0 or 1. The machine learning models used in the work learn from the training data and label accordingly. After the learning process, the model is used to predict the classes for unseen test data.

Table 3: Results for train set using TDM representation followed by classical ML

TDM	Task	Accuracy (%)	Precision	Recall	F1-Score
Decision Tree	Subtask 1	96.7	0.881	0.791	0.833
KNN	Subtask 1	94.3	0.932	0.490	0.642
Logistic Regression	Subtask 1	89.3	1.00	0.053	0.100
Naive Bayes	Subtask 1	94.8	0.780	0.704	0.740
Random Forest	Subtask 1	97.4	1.0	0.750	0.857
AdaBoost	Subtask 1	98.3	0.966	0.867	0.914
SVM	Subtask 1	98.0	0.916	0.88	0.902
Decision Tree	Subtask 2	99.9	0.994	1.00	0.997
KNN	Subtask 2	99.7	0.983	0.988	0.985
Logistic Regression	Subtask 2	99.9	1.00	0.994	0.997
Naive Bayes	Subtask 2	98.5	1.00	0.865	0.928
Random Forest	Subtask 2	100	1.0	1.0	1.0
AdaBoost	Subtask 2	100	1.0	1.0	1.0
SVM	Subtask 2	99.9	1.0	0.994	0.997

3.2.1 Data representation of samples with headers:

- TDM representation of data is done and the vocabulary is built using train and test data
- SVD or NMF is used for feature extraction and dimensionality reduction
- Step 2 is followed by applying different classical ML techniques like Decision Tree, Random Forest, AdaBoost, KNN and SVM

3.2.2 Data representation of samples with no headers:

- Data Preprocessing- data preprocessing involves counting the number of @, # symbol in each data sample. Then @ and # counts are removed from orginal corpus
- TDM representation of data, followed by appending the @ count and # count
- SVD or NMF is applied for feature extraction and dimensionality reduction
- Step 3 is followed by applying different classical ML techniques like Decision Tree, Random Forest, AdaBoost, KNN and SVM on the numeric representation of the data

3.3 Methodology

The paper discusses classical machine learning approaches like Decision Tree, K- Nearest Neighbors, Logistic Regression, Naive Bayes, Random Forest and SVM. The metrics used for analyzing the performance of the model are as follows:

1. Accuracy
2. Precision
3. Recall
4. F1-Score

For numeric representation of data TDM is used. The TDM matrix is passed to SVD and NMF for extracting best features.

- SVD decomposes a matrix as the product of three different matrices. These matrices can be geometrically interpreted as rotation, stretching, rotation. The mathematical representation of SVD is : $A = U\Sigma V^T$ where U represents the orthonormal eigenvectors of AA^T. And V^T represents the orthonormal eigenvectors of A^TA. It is a diagonal matrix and represents the singular values. For extracting features the product of U is sufficient. In all the cases the rank is chosen as 30. So the resultant train and test dataset size will be reduced to, total no of data points x 30.
- The second technique used for feature extraction is NMF. It factorizes a matrix as the product of two matrices i.e, W and H. These matrices does not contain any negative elements. The TDM is passed as the input to NMF. NMF generates a list of topics. These topics acts as a basis for representing the original dataset.

Table 4: Results for train set using TDM with SVD followed by classical ML

TDM with SVD	**Task**	**Accuracy (%)**	**Precision**	**Recall**	**F1-Score**
Decision Tree	Subtask 1	91.5	0.589	0.628	0.607
KNN	Subtask 1	94.0	0.792	0.582	0.671
Logistic Regression	Subtask 1	93.3	0.824	0.454	0.586
Naive Bayes	Subtask 1	29.7	0.125	0.959	0.221
Random Forest	Subtask 1	95.2	0.914	0.597	0.722
AdaBoost	Subtask 1	94.7	0.808	0.643	0.716
SVM	Subtask 1	94.2	0.816	0.566	0.669
Decision Tree	Subtask 2	97.4	0.898	0.871	0.884
KNN	Subtask 2	99.7	0.994	0.977	0.985
Logistic Regression	Subtask 2	99.5	0.971	0.988	0.980
Naive Bayes	Subtask 2	75.1	0.309	0.971	0.469
Random Forest	Subtask 2	99.3	1.00	0.942	0.970
AdaBoost	Subtask 2	99.5	0.988	0.971	0.979
SVM	Subtask 2	99.3	0.960	0.977	0.968

Table 5: Results for train set using TDM with NMF followed by classical ML

TDM with NMF	**Task**	**Accuracy (%)**	**Precision**	**Recall**	**F1-Score**
Decision Tree	Subtask 1	91.4	0.581	0.638	0.608
KNN	Subtask 1	92.7	0.709	0.510	0.593
Logistic Regression	Subtask 1	89.6	0	0	0
Naive Bayes	Subtask 1	34.8	0.133	0.954	0.234
Random Forest	Subtask 1	94.6	0.899	0.546	0.679
AdaBoost	Subtask 1	94.6	0.794	0.648	0.713
SVM	Subtask 1	90.6	0.686	0.179	0.283
Decision Tree	Subtask 2	98.9	0.948	0.953	0.950
KNN	Subtask 2	97.7	0.936	0.854	0.893
Logistic Regression	Subtask 2	89.3	1.00	0.053	0.100
Naive Bayes	Subtask 2	72.4	0.289	0.988	0.447
Random Forest	Subtask 2	99.5	0.982	0.977	0.979
AdaBoost	Subtask 2	99.7	1.0	0.977	0.988
SVM	Subtask 2	97.6	0.979	0.807	0.885

Table 6: Summary of test set results for TDM with SVD representation followed by classical ML

TDM with SVD	Task	Accuracy (%)	Precision	Recall	f1-score	TP	TN	FP	FN
Decision Tree	Sub task 1	73.32	0.875	0.815	0.844	3121	32	443	704
KNN	Sub task 1	87.79	0.889	0.985	0.9349	3770	5	470	55
Logistic Regression	Sub task 1	88.744	0.889	0.997	0.940	3816	0	475	9
Naive Bayes	Sub task 1	72.97	0.874	0.813	0.842	3110	28	447	715
Random Forest	Sub task 1	88.906	0.889	0.999	0.941	3823	0	475	2
Ada Boost	Sub task 1	88.581	0.889	0.995	0.9394	3809	0	475	16
SVM	Sub task 1	88.302	0.888	0.992	0.93786	3796	1	474	29
Decision Tree	Sub task 2	82.86	0.886	0.923	0.904	3417	59	437	282
KNN	Sub task 2	87.55	0.8812	0.992	0.933	3672	1	495	27
Logistic Regression	Sub task 2	67.29	0.854	0.758	0.803	2804	19	477	895
Naive Bayes	Sub task 2	84.41	0.882	0.949	0.914	3511	30	466	188
Random Forest	Sub task 2	88.17	0.881	1	0.937	3699	0	496	0
Ada Boost	Sub task 2	88.17	0.881	1	0.937	3699	0	496	0
SVM	Sub task 2	54.11	0.8290	0.604	0.6989	2235	35	461	1464

Table 7: Summary of test set results for TDM with NMF representation followed by classical ML

TDM with NMF	Task	Accuracy (%)	Precision	Recall	f1-score	TP	TN	FP	FN
Decision Tree	Sub task 1	86.69	0.916	0.935	0.926	3580	148	327	245
KNN	Sub task 1	90.34	0.9338	0.959	0.946	3670	215	260	155
Logistic Regression	Sub task 1	89.06	0.890	1	0.9421	3825	5	470	0
Naive Bayes	Sub task 1	82.09	0.8831	0.920	0.901	3521	9	466	304
Random Forest	Sub task 1	89.06	0.891	0.998	0.942	3820	9	465	5
Ada Boost	Sub task 1	89.18	0.895	0.994	0.942	3803	32	443	22
SVM	Sub task 1	91.41	0.913	0.9976	0.953	3816	115	360	5
Decision Tree	Sub task 2	65.74	0.857	0.733	0.790	2715	43	453	984
KNN	Sub task 2	75.733	0.867	0.855	0.861	3163	14	482	536
Logistic Regression	Sub task 2	88.15	0.881	0.999	0.937	3698	0	496	1
Naive Bayes	Sub task 2	54.42	0.844	0.592	0.696	2191	92	404	1508
Random Forest	Sub task 2	71.13	0.913	0.742	0.819	2747	237	259	952
Ada Boost	Sub task 2	71.99	0.910	0.7566	0.826	2799	221	275	900
SVM	Sub task 2	70.05	0.903	0.739	0.813	2737	202	294	962

4 Results

The datasets provided are highly imbalanced, and still gives considerably high classification accuracy. The following tables lists the performance of each classical machine learning techniques applied for the formulated binary classification problem to detect whether an email is phishing or legitimate. In the Tables 3, 4 and 5 the results obtained are for predicting the labels for the training data by using sklearn train-test split where 33% of the training data is used for validating the result and the rest for training the model. From the results obtained, Random Forest has outperformed all other techniques for the training data set. Test data results are provided in Table 6 and 7. Table 6 describe the results for classification using TDM with SVD for both subtasks. Table 7 represent the results for classification using TDM with NMF for both subtasks. The shared task organizers had given the true positive (TP), true negative (TN), false positive (FP) and false negative (FN) values for test dataset which are listed in Table 6 and 7 along with accuracy, precision, recall and F1-score, which are estimated taking TP, TN, FP and FN values and using it in the following equations:

$$accuracy = \frac{(tp + tn)}{(tp + fp + tn + fn)} \tag{1}$$

$$precision = \frac{tp}{(tp + fp)} \tag{2}$$

$$recall = \frac{tp}{(tp + fn)} \tag{3}$$

$$f1 - score = \frac{(2 * tp)}{(2 * tp + fp + fn)} \tag{4}$$

5 Conclusion

The paper focuses on phishing email detection which is a major threat in the present scenario. For both the subtasks numeric representation of data is done using the methodology, TDM with SVD and TDM with NMF. These representations are followed by applying classical machine learning techniques to the data inorder to classify an email as phishing or legitimate. One of the drawback with the current model is that the proposed mechanism relies on feature selection, which requires domain knowledge. To overcome this issue deep learning models can be incorporated, which can learn more complex patterns from the raw data and use it as features that produce more efficacy and this can be considered as a possible future work. In addition to that both the subtasks belongs to unconstrained category, allowing external datasets to be used for the training purpose. The datasets provided in the subtasks are highly imbalanced. With highly imbalanced datasets, we are able to achieve considerably high phishing email detection rate in both the subtasks. The tasks are unconstrained but we have not used datasets from any other external sources. Thus, the phishing email detection rate of the proposed architecture can be easily enhanced by adding additional data from external sources with the data provided in the shared task. This will be considered as one of the significant direction towards the future work.

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

[ANNWN07] Saeed Abu-Nimeh, Dario Nappa, Xinlei Wang, and Suku Nair. A comparison of machine learning techniques for phishing detection. In *Proceedings of the anti-phishing working groups 2nd annual eCrime researchers summit*, pages 60–69. ACM, 2007.

[BVP] Barathi Ganesh Hullathy Balakrishnan, Anand Kumar Madasamy Vinayakumar, and Soman Kotti Padannayil. Nlp cen amrita@ smm4h: Health care text classification through class embeddings.

[CM01] Xavier Carreras and Lluis Marquez. Boosting trees for anti-spam email filtering. *arXiv preprint cs/0109015*, 2001.

[CXMX05] Zhan Chuan, Lu Xianliang, Hou Mengshu, and Zhou Xu. A lvq-based neural network anti-spam email approach. *ACM SIGOPS Operating Systems Review*, 39(1):34–39, 2005.

[DWV99a] Harris Drucker, Donghui Wu, and Vladimir N Vapnik. Support vector machines for spam categorization. *IEEE Transactions on Neural networks*, 10(5):1048–1054, 1999.

[DWV99b] Harris Drucker, Donghui Wu, and Vladimir N Vapnik. Support vector machines for spam categorization. *IEEE Transactions on Neural networks*, 10(5):1048–1054, 1999.

[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.

[Faw03] Tom Fawcett. In vivo spam filtering: a challenge problem for kdd. *ACM SIGKDD Explorations Newsletter*, 5(2):140–148, 2003.

[Gee03] Kevin R Gee. Using latent semantic indexing to filter spam. In *Proceedings of the 2003 ACM symposium on Applied computing*, pages 460–464. ACM, 2003.

[KMAH04] Svetlana Kiritchenko, Stan Matwin, and Suhayya Abu-Hakima. Email classification with temporal features. In *Intelligent Information Processing and Web Mining*, pages 523–533. Springer, 2004.

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

[NVK+15] Maryam M Najafabadi, Flavio Villanustre, Taghi M Khoshgoftaar, Naeem Seliya, Randall Wald, and Edin Muharemagic. Deep learning applications and challenges in big data analytics. *Journal of Big Data*, 2(1):1, 2015.

[SDHH98] Mehran Sahami, Susan Dumais, David Heckerman, and Eric Horvitz. A bayesian approach to filtering junk e-mail. In *Learning for Text Categorization: Papers from the 1998 workshop*, volume 62, pages 98–105, 1998.

[SVKS15] Shriya Se, R Vinayakumar, M Anand Kumar, and KP Soman. Amrita-cen@ sail2015: sentiment analysis in indian languages. In *International Conference on Mining Intelligence and Knowledge Exploration*, pages 703–710. Springer, 2015.