Distributed Representation using Target Classes: Bag of Tricks for Security and Privacy Analytics

Amrita-NLP@IWSPA-2018

Barathi Ganesh HB[a,b], Vinayakumar R[a], Anand Kumar M[a], Soman KP[a]

[a]Center for Computational Engineering and Networking(CEN),
Amrita School of Engineering, Coimbatore
Amrita Vishwa Vidyapeetham, India
m_anandkumar@cb.amrita.edu

[b]Arnekt Solutions Pvt. Ltd., Pentagon P-3, Magarpatta City
Pune, Maharashtra, India
barathiganesh.hb@arnekt.com

Abstract

The extensive growth of internet users provides the opportunities for anomalies to intrude our privacy and security. Phishing is one among them that has turned out to be a major issue in the recent times, that directly hit specific targeted group of people asking for their credentials, personal and other sensitive information. This paper elaborates the module submitted to IWSPA-AP Shared Task at IWSPA 2018 that focuses on distinguishing the phishing and legitimate emails. In fundamental it is a text classification problem in which representation serves as the core component and also has a direct relationship to the final performance. This work assess and reports the performance of distributed representation in detection of phishing emails as a text classification problem. The word embedding and neural bag-of-ngrams facilitates to extract syntactic and semantic similarity of emails. The experimented module obtains promising and consistence performance on both the train and test corpus in-terms of time and accuracy. The model obtains 99% and 97% as the f1 score on the unseen test corpus.

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

Phishing is one of the social engineering method which is used to fetch the personal and sensitive information of the internet users by installing malware on their computers thereby exploiting the weaknesses in current web security. At the year end of 2017, the anti-phishing system prevented nearly 60 million attempts to phishing pages[1], which shows the potential use of an anti-phishing system. Every year, the phishing attack on unique users of internet is increasing worldwide. This ensures the need of effective anti phishing methods and induces the research community to integrate the Artificial Intelligence (AI) methods with Cyber security modules [ZZJ+17]. The International Workshop on Security and Privacy Analytics - Anti Phishing (IWSPA-AP[2]) has been hosting a shared task to build a classifier that will be able to distinguishing the phishing and legitimate emails.

Through spam emails people deliver all kinds of malicious attacks. It can be delivered using several ways, by attaching files with malicious content or by sending a link of a compromised website. The frequently used type of malware attack through spam emails are

[1]securelist.com/spam-and-phishing-in-q3-2017/82901/
[2]dasavisha.github.io/IWSPA-sharedtask/#oc

blended attacks. It uses more than one method to deliver malware on an internal network. Blended attacks often starts from illegitimate emails, which may not contain malware but provide links to compromised websites. Usually attackers send emails in such a way that it looks legitimate to a normal user by mixing authentic links and false links that will contain URLs to some fake website. As per the survey produced by IBM's X-Force research team, more than half of the emails produced worldwide are scam. The percentage of spam email amounted to 55.9% in the first quarter of 2017, which shows there are chances of greater possibility for gradual increase of spam emails in the coming years.

The task can be formulated as a text classification problem in which emails are the documents and target classes are phishing and legitimate ones. Any text classification application will contains representation (representing text as a numeric values), feature extraction (getting informative words with respect to the target classes) and a classifier (transforms features to target classes) as its base components [AZ12]. Among them representation is more complex and core part of the module, that represents the context of the text in numbers. Representation defines the effectiveness of the classification models to make final predictions. Hence this experiment focuses much on the text representation.

The content from the phishing sites are highly semantically similar to contents in original sites. Thus this becomes mandatory to represent the context of the text, than representing text as symbols. The classical representation methods Vector Space Models (VSM) failed to do so [BGAKS17]. The Vector Space Models of Semantics (VSMs) or Distributional Representation methods are able to include context only to some extent [GKS16, BGAKS16a]. Unlike image and speech, texts are represented using numerical values by taking terms (words or phrases) as a symbols in classical methods. Both these models requires high computation because of well known problem called "Curse of Dimensionality". Due to this, these methods cannot be run on the huge corpus which is necessary for the effective representation. Finally distributed representation methods are introduced that reflects the context of the text as a low dimensional dense vector and provides the flexibility in choosing the dimension of the vector. By considering these factors, this experimentation is performed using distributed representation methods.

One of the well known distributed representation method is word2vec (word to vector) and the latterly introduced methods like doc2vec (document to vector), Glove (global vectors) and fastText which are the flavours of word2vec with some notable changes to enhance the representation. Given a word to word2vec, it will produce the vector in desired dimension that reflects the context of word [GL14]. When it comes to representing a text with multiple words, either average of those word vectors or the matrix out of concatenating those word vectors will be decomposed to form a single vector [BGAKS16b]. The learning of word2vec is improved by combining word2vec with the co-occurrence matrix by forming the so called Glove. Glove provides the flexibility to train small corpus with promising performance [PSM14]. Both these methods represent poor sequence of words since averaging of word vector does not consider the order of the word. The doc2vec method introduces a way to represent the sequence of words to a vector [LM14]. The architecture is similar to the word2vec, provided one more weight matrix will also be learned along with the weight matrix of word2vec for representing the sequence of words. At-last fastText[3] has been introduced, where it learns vector for a given word from the class it belongs to, rather than the earlier methods where it learns by predicting next word of the given word [JGBM16]. Since the number of classes is always less than number of words, this method is faster than others.

By observing above, this work utilizes fastText for representing texts as vector and softmax for making the final predictions. The given email documents are normalized through a preprocessor to remove uninformative features, then fastText with hyper-parameter tuning used for the document representation and classification. The remaining part of the paper details the related work performed in detection of phishing in Section 2, problem formulation and working principles of fastText are given in Section 3 and the Section 4 details the experiment conducted and discusses about the obtained results.

2 Related Works

Among the traditional methods that we have been following since ages in text classification, artificial intelligence (AI) is another technique which became popular in last few decades. AI uses supervised learning classification algorithms to do binary classification of spam emails.

Presently there are not many methods designed for effective detection of phishing emails (focused on finding phishing URLs) and most of the methods showing good performance on detecting spam mails. Mostly the researchers try to perform the manual feature learning (number of words, number of domains, URLs, number of links, number of dots, message hashes) by analysing content of the email and then applies

[3] fasttext.cc

Table 1: Training Corpus : L - Legitimate, P - Phishing, T - Total

Task	Type	Training		
		L	P	T
1	No Header	5088	612	5700
2	Header	4082	501	4583

Table 2: Testing Corpus

Task	Type	Testing
1	No Header	4300
2	Header	4195

classical machine learning algorithms [MW04, RH04, TK17, SMAC17, Alt17]. Most of the researchers follows Bag Of Word models (Document - Term Matrix and Term Frequency - Inverse Document Frequency Matrix) based on of VSM [AN15, AZZ+15, Alt17].

Email headers play a key role in identifying spam emails. It determines the recipient of a message and also tracks the route of the mail as it passes the mail servers. Email headers provide extremely useful features that could be used for machine learning models to efficiently classify spam emails [LT04, S+09, WC07].

Recently authors started developing anti-phishing models using deep learning algorithms like Deep Belief Network (DBN), Recurrent Neural Network (RNN), Convolutional Neural Network (CNN) and etc,. [ZL17, BBV+17, SAZ18, RJ17, LNRW]. The manual feature engineering has got eliminated and it has been taken care by the intermediate hidden layers of neural networks. We can conclude from the above that, though the technology advances in text classification anti-phishing problem is not addressed properly and requires more research. The semantic representation of text with less computation is suitable for real world data and hence here this work assess the performance of distributed representation of text with respect to target classes in anti-phishing task.

3 Corpus Statistics

Only a few set of benchmark corpus is available for detecting phishing content from the emails. The corpus for this experiment has been provided by the IWSPA-AP shared task organizers [EDMB+18]. There are two set of corpus provided one with header and another is without headers. The detailed statistics about the training corpus has given in the Table 1 and testing corpus in 2.

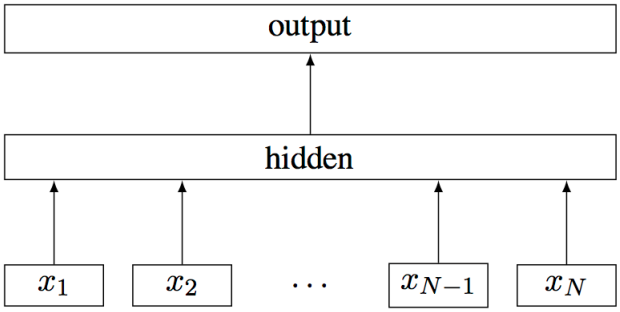

Figure 1: Distributed Representation Architecture - in word2vec output will be a word ; in fastText output will be a target class

4 Problem Definition

The given corpus are the text documents that belongs to the target classes phishing and legitimate.

$$D = \{d_1, d_2, d_3, ..., d_n\} \tag{1}$$

$$D \epsilon \{phishing,\ legitimate\} \tag{2}$$

The objective of this experiment is to map the documents d_i to one of the target class it belongs. n refers to the total number of documents. The first step is to find the distributed representation of d_i that reflects the context of d_i. This is given as,

$$(R_i)_{1\times m} = distributed\ representation\,(d_i) \tag{3}$$

$$(R)_{n\times m} = distributed\ representation\,(D) \tag{4}$$

On successful representation each R_i will be maps to the target class it belongs. In this paper we have experimented distributed representation of text with respect to it target class.

Word2vec is the first distributed representation method developed to represent the context of the given word as a vector [GL14]. In word2vec the representation is learned by feeding a $word_i$ to the architecture, which in turn has to predict the $word_c$ i.e. co-occurring words of the $word_i$. For an example "boy chases the cat". Given the word "boy" the architecture predicts its co-occurring words "chases", "the" and "cat". During the learning phases word2vec learns W_1 inner matrix and W_2 outer matrix that transforms $word_i$ to $word_c$ and vice verse. W_1 gives the representation for words in the vocabulary. This is given as,

$$h = W_1^T X = V_{W_1}^T \tag{5}$$

$$u_{w_2} = W_2^T h \tag{6}$$

$$p\,(word_c \mid word_i) = \frac{exp\,(u_{W_2})}{\sum^V exp\,(u_{W_2})} \tag{7}$$

Table 3: Training Corpus Word Count Statistics : U/D - average number of unique words per document; W/D - average number of words per document

Type	#Total Words	#Unique Words	Avg. W/D	Avg. U /D
Header	2065327	117085	362.3	34.9
No Header	1101893	82075	240.4	17.9
Combined	3167220	199160	602.7	52.8

The softmax is applied after W_2 to make the prediction of $word_c$, that includes high computation. Another constrain is representing sequence of the text through word2vec that is performed by computing aggregated sum of word vectors present in the corresponding word sequence. This is not an effective method and losses the properties of text to be represented. FastText overcomes this scenario by finding word representation and predicting the class it belongs to at the output layer [JGBM16]. In this work, both the sub-tasks are experimented using fastText. This is given as,

$$-\frac{1}{N}\sum_{i=1}^{n} y_i log\left(f\left(W_2 W_1 x_i\right)\right) \tag{8}$$

Where, x_i is the normalized bag of features of the ith document, y_i the target label, W_1 and W_2 are the weight matrices. Since the target classes are in finite count at output layer, the computation required by the softmax also become lower. These words will be represented as a vector with respect to the target class it belongs to. There are a number of parameters available which needs to be tuned with respect to the data and the classification problem. A higher level common architecture for distributed representation is given Figure 1 [JGBM16]. This same architecture is used for both the sub tasks.

5 Experiment and Observations

The task is to classify each given email sample into either legitimate and phishing [EDB+18]. The subtask 1 contains email samples without header and subtask 2 contains email samples with header. The given data-set is unbalanced in the ratio of nearly 1:8 $\approx$ legitimate:phishing. The word count statistics of the corpus is detailed in Table 3.

This experiment has been performed on a system with the configuration : 16 GB RAM and i7 processor. The model has been built using Python 2[4], fastText library package[5] and code made publicly available[6].

By considering the computation required for 10-fold 10-cross validation, in this work the given corpus has been shuffled to avoid the localization of model to particular subset and split into 80% for training and 20% for validation. Before splitting the corpus, the corpus is preprocessed to remove the punctuation, special symbols and empty spaces. The hyper-parameters are tuned (Dimension: 100 to 1000, Minimum Word Count: 1 to 5, Epochs: 3 to 10, N-Grams: 2, Loss Function: Softmax and Learning rate: 0.001, 0.01, 0.1) to obtain the maximum f1 score for validation data. The F1 score has been considered to make sure that the system performs well over all the classes, which would inherently mean a better sensitivity to the prediction of phishing mails (lower in quantity), from the legitimate mails (higher in quantity). The vector we get from distributed representation will capture semantic properties which will be very helpful in improving the performance of the natural language processing (NLP) system to get better results than traditional bag-of-words representations. Neural bag-of-ngrams vectors resulting from fastText is a dense, real-valued vector representation and also captures the semantics of the context. It is the combination of bag-of-ngram and neural word embedding which is robust, simple and flexible.

We have submitted two models, where the first model is developed by considering data with header and data without headers independently while the second model built by combining both data to make a single model. The results obtained during the training phase is given in 4. It can be observed that combined

Table 4: Training Performance on 20% Test set

Type	Precision	Recall	F1
Header	0.98	0.98	0.98
No Header	0.99	0.99	0.99
Combined	0.98	0.98	0.98

Table 5: Hyper - Parameters Values

Parameter	Value
Dimension	100
Minimum Word Count	1
Epochs	5
N-grams	2
Loss Function	Softmax
Learning Rate	0.1

[4]www.python.org
[5]pypi.python.org/pypi/fasttext
[6]github.com/BarathiGanesh-HB/IWSPA-AP

Table 6: Model Performance on Test Corpus

Model	TP	TN	FP	FN
Header	3680	480	16	19
No Header	3742	347	128	83
Combined Header	3676	487	9	23
Combined No Header	3724	369	106	101

Table 7: Model Performance on Test Corpus in terms of P - Precision, R - Recall and F1

Model	P	R	F1
Header	**0.99**	**0.99**	**0.99**
No Header	**0.97**	**0.98**	**0.97**
Combined Header	0.99	0.99	0.99
Combined No Header	0.97	0.97	0.97

data model performs 1% lesser than the independent models. The final model has been built using hyper-parameters listed in 5.

The model performance on the test corpus has been measured by the task organizers. The performance reported by the task organizers are shown in detail in Table 6. These reports given by organizers included Precision, Recall and F1 measures as shown in Table 7. From the Table 4 and 7 we can conclude that the model has performed well on the test corpus as on the train corpus.

6 Conclusion

An anti-phishing system has been built successfully using distributed representation method. This attains good performance during the training phase. The combined data model performs 1% lesser than the independent models built, with and without header files which attains near 99% as the f1 score in training period. On test corpus both the models gave similar performance. The semantic representation of text with less computation and reliable performance is suitable for real world data. Hence this experimented model is suitable for real world applications. The performance of the system can be enhanced with more complex deep learning architecture at the classification stage. In future this architecture could be made more effective by training using Graphics Processing Unit(GPU).

References

[Alt17] Altyeb Altaher. Phishing websites classification using hybrid svm and knn approach. *Int J Adv Comput Sc*, 421:8, 2017.

[AN15] J Adamkani and K Nirmala. A content filtering scheme in social sites. *Indian Journal of Science and Technology*, 8(33):1, 2015.

[AZ12] Charu C Aggarwal and ChengXiang Zhai. A survey of text classification algorithms. In *Mining text data*, pages 163–222. Springer, 2012.

[AZZ+15] Ahmed Abbasi, Fatemeh Mariam Zahedi, Daniel Zeng, Yan Chen, Hsinchun Chen, and Jay F Nunamaker Jr. Enhancing predictive analytics for anti-phishing by exploiting website genre information. *Journal of Management Information Systems*, 31(4):109–157, 2015.

[BBV+17] Alejandro Correa Bahnsen, Eduardo Contreras Bohorquez, Sergio Villegas, Javier Vargas, and Fabio A González. Classifying phishing urls using recurrent neural networks. In *Electronic Crime Research (eCrime), 2017 APWG Symposium on*, pages 1–8. IEEE, 2017.

[BGAKS16a] HB Barathi Ganesh, M Anand Kumar, and KP Soman. Distributional semantic representation for text classification and information retrieval. *CEUR*, 1737, 2016.

[BGAKS16b] HB Barathi Ganesh, M Anand Kumar, and KP Soman. Semantic relation from word embeddings in higher dimension. *Proceedings of SemEval*, pages 1290–1295, 2016.

[BGAKS17] HB Barathi Ganesh, M Anand Kumar, and KP Soman. Vector space model as cognitive space for text classification. *arXiv preprint arXiv:1708.06068*, 2017.

[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.

[GKS16] HB Barathi Ganesh, M Anand Kumar, and KP Soman. From vector space models to vector space models of semantics. In *Forum for Information Retrieval Evaluation*, pages 50–60. Springer, Cham, 2016.

[GL14] Yoav Goldberg and Omer Levy. word2vec explained: Deriving mikolov et al.'s negative-sampling word-embedding method. *arXiv preprint arXiv:1402.3722*, 2014.

[JGBM16] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*, 2016.

[LM14] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *International Conference on Machine Learning*, pages 1188–1196, 2014.

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

[LT04] Chih-Chin Lai and Ming-Chi Tsai. An empirical performance comparison of machine learning methods for spam e-mail categorization. In *Hybrid Intelligent Systems, 2004. HIS'04. Fourth International Conference on*, pages 44–48. IEEE, 2004.

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

[PSM14] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[RH04] Isidore Rigoutsos and Tien Huynh. Chung-kwei: a pattern-discovery-based system for the automatic identification of unsolicited e-mail messages (spam). In *CEAS*, 2004.

[RJ17] Yafeng Ren and Donghong Ji. Neural networks for deceptive opinion spam detection: An empirical study. *Information Sciences*, 385:213–224, 2017.

[S+09] Jyh-Jian Sheu et al. An efficient two-phase spam filtering method based on e-mails categorization. *IJ Network Security*, 9(1):34–43, 2009.

[SAZ18] Sami Smadi, Nauman Aslam, and Li Zhang. Detection of online phishing email using dynamic evolving neural network based on reinforcement learning. *Decision Support Systems*, 2018.

[SMAC17] Abdulhamit Subasi, Esraa Molah, Fatin Almkallawi, and Touseef J Chaudhery. Intelligent phishing website detection using random forest classifier. In *Electrical and Computing Technologies and Applications (ICECTA), 2017 International Conference on*, pages 1–5. IEEE, 2017.

[TK17] Fadi Thabtah and Firuz Kamalov. Phishing detection: a case analysis on classifiers with rules using machine learning. *Journal of Information & Knowledge Management*, 16(04):1750034, 2017.

[WC07] Chih-Chien Wang and Sheng-Yi Chen. Using header session messages to anti-spamming. *Computers & Security*, 26(5):381–390, 2007.

[ZL17] Jiahua Zhang and Xiaoyong Li. Phishing detection method based on borderline-smote deep belief network. In *International Conference on Security, Privacy and Anonymity in Computation, Communication and Storage*, pages 45–53. Springer, 2017.

[ZZJ+17] Xi Zhang, Yu Zeng, Xiao-Bo Jin, Zhi-Wei Yan, and Guang-Gang Geng. Boosting the phishing detection performance by semantic analysis. In *Big Data (Big Data), 2017 IEEE International Conference on*, pages 1063–1070. IEEE, 2017.