Classifying Phishing Email Using Machine Learning and Deep Learning

Sikha Bagui
Department of Computer Science
University of West Florida
Pensacola, USA
bagui@uwf.edu

Debarghya Nandi
Division of Epidemiology and Biostatistics
University of Illinois at Chicago
Chicago, USA
dnandi4@uic.edu

Subhash Bagui
Department of Mathematics and Statistics
University of West Florida
Pensacola, USA
sbagui@uwf.edu

Robert Jamie White
AppRiver
Pensacola, USA
jamie@appriver.com

***Abstract*— In this work, we applied deep semantic analysis, and machine learning and deep learning techniques, to capture inherent characteristics of email text, and classify emails as phishing or non -phishing.**

***Keywords*— Deep learning, Machine learning, one-hot encoding, deep semantic analysis, phishing emails.**

I. Introduction

Phishing email attacks are intelligently crafted social engineering email attacks in which victims are conned by email to websites that impersonate legitimate sites. Victims of phishing email attacks perceive these sites to be associated with trusted companies such as Amazon or Google and hence are tricked into logging into such sites and sharing sensitive information. The FBI has suggested that the impact of phishing attacks could be costing US businesses somewhere around $5 billion a year [1]. Although many anti-phishing tools and techniques have been developed, phishing is still difficult to effectively defend, which puts the individual as well as organization at risk [2]. Though many email filters have been developed for spam emails, very few phishing email filters have been developed [8].

II. Our Work

In this work, we applied deep semantic analysis, and machine learning and deep learning techniques, to capture inherent characteristics of the email text, and classify emails as phishing or non-phishing. Until very recently, little work has been devoted to semantic analysis in phishing detection [8], let alone phishing email detection. Representation of text is a significant task in natural language processing and in recent years deep learning has been widely used in various natural language processing tasks like topic classification, sentiment analysis and language translation [8]. In this research, we applied one-hot encoding with and without word-phrasing for deep semantic analysis with machine learning and deep learning techniques, to classify emails. A comparison of the results of various machine learning (ML) classifiers – Naïve Bayes, Support Vector Machines (SVM), Decision Tree, and Deep Learning (DL) classifiers -- Long Short Term Memory (LSTM), Convolutional Neural Networks (CNN), and Word Embedding have been presented.

A. The Phishing Data

An email data set that contains 18366 labelled emails, of which 3416 are phishing emails and 14950 are regular emails, was used for this study. The emails were collected across US industries – insurance, law, medical, hotel, school, banking, and real estate marketing. 70% of the data was used for training and the rest was used for testing. Each of these emails contained a subject as well as body text, and the number of users that this email was sent to. This work focused on analyzing the text content of the emails and classifying them as phishing or not.

B. Classifiers Used

- *Naïve Bayes* is a probabilistic classifier that has been used in many different fields including text classification [3,7] and sentiment analysis [6].
- *Support Vector Machines (SVM)* is a supervised learning model widely used for classification, known to be efficient in high dimensional spaces.
- *Decision Tree* is a popular Machine Learning tool that uses a tree-like structure to represent events and their possible outcomes in different conditions.
- *Long Short Term Memory (LSTM)* is one of the most emerging and scalable models for learning sequential data [4, 9].
- *Convolutional Neural Networks (CNN)* are a class of deep neural networks that have been widely used in Natural Language Processing.
- *Word Embedding* is a technique where individual words are represented as sparse vectors based on the context window and then mapped to a low dimensional vector space.

C. One Hot Encoding

ML or DL algorithms cannot operate directly on textual data. They require numeric input. So, as part of feature engineering, the emails were encoded as one-hot vectors with a vocabulary size of 100. A one-hot vector is a $1 \times N$ matrix (vector) consisting of 0s in all cells of the vector with the exception of a single 1 in a cell used uniquely to identify a word. In this one-hot encoding process, each of the words of the emails were taken and encoded separately and the vectors were then padded at the end to ensure that they were of the same length.

D. Experimentation with Machine Learning and Deep Learning Algorithms

These matrices were fed into the ML and DL algorithms (without word phrasing), where the vectors are mapped to a low dimensional space. DL experiments were performed using the keras python library. Keras is built on the top of Tensorflow designed for fast testing of deep neural networks. Proper estimation of hyperparameters is key to optimizing DL models. As pointed out in [5], selection of hyperparameters in DL, like context window, filter size, number of feature maps, pooling strategies are completely dependent on the nature of the data set. We carried out plenty of trial runs and came up with the optimal parameters suited for our data set, to classify phishing versus non-phishing emails. Our LSTM model consisted of 2 LSTM layers with 4 hidden nodes, followed by a dense layer of one node and a sigmoid activation layer. This created a total of 393 trainable parameters. For CNN, the most optimum filter size (2), pool size (2), strides (1) and the relu activation function was used for the convolution layers. Accuracy gradually improved with the increase in the number of feature maps. The Word Embedding architecture constituted an input layer, an embedding layer with 4 hidden nodes and finally a dense layer with a single hidden node. The sigmoid activation function was used in the final layer. The base model had a total of 721 parameters. The tests were run with Dropout as well as without Dropout, to compare their performance. For the deep learning model, 'Relu' was the most effective activation function because it does not suffer from the diminishing gradient problem. However, we used sigmoid for the end-Dense layer, because it seemed to perform well for our binary class data.

Since context might play an important role in phishing detection, rather than only using individual words for the one-hot encoding, in the next step we tokenized the email text into multiple *n*-gram features without removing any of the stop-words. These *n*-gram features were then encoded using one-hot encoding and mapped to a vector space model and fed into the ML and DL algorithms (with word phrasing).

III. Results and Conclusion

In this work we presented how one-hot encoding can be used for phishing vs non-phishing email classification with and without word-phrasing. We compared the accuracy of the different ML as well as DL models with and without word phrasing. On the average, as can be seen from Table 1 and Figure 1, the DL models performed a little better than the ML models. And, except for SVM, the accuracy was slight better with word phrasing then without word phrasing. This means that the context of the email language is important in determining whether an email is a phishing email or not.

	Accuracy with Word Phrasing	Accuracy without Word Planning
Naïve Bayes	93.27%	75.72%
SVM	82.35%	93.88%
Decision Tree	97.50%	94.26%
LSTM	96.64%	95.91%
CNN	97.20%	95.97%
Word Embedding	98.89%	96.34%

Table 1: Accuracy with and without Word Phrasing for different algorithms

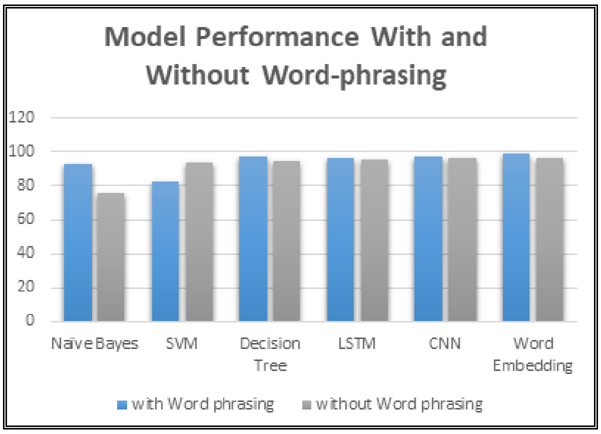

Figure 1: Model Performance with and without Word phrasing

Acknowledgment

The work has been partially supported by the Askew Institute at the University of West Florida.

References

[1] E. Fette, N. Sadeh, and A. Tomasic. "Learning to detect phishing email," in *Proceedings of the 16th International World Wide Web Conference (WWW '07),* Alberta, Canada, 2007, pp. 649-656.

[2] P. Danny, "What is Phishing? Everything you need to know to protect yourself from scam emails and more," Sept 6, 2017: https://www.zdnet.com/article/what-is-phishing-how-to-protect-yourself-from-scam-emails-and-more

[3] V. Gupta and G. Lehal, "A Survey of Text Mining Techniques and Applications," Journal of Emerging Technologies in Web Intelligence, vol. 1(1), 2009, pp. 60-76.

[4] Palangi, H., Deng. L., She, Y., Gao, J., He, X., Chen, J., Song, X., and Ward, R. (2016). Deep Sentence Embedding Using Long Short-Term Memory Networks: Analysis and Application of Information Retrieval, IEEE/ACM Transactions on Audio, Speech, and Language Processing, 24(4), 694-707.

[5] S. Sarju, R. Thomas, and E. Shyni, "Spam Email Detection and Using Structural Features," International Journal of Computer Applications, vol. 89(3), 2014, pp. 38-41.

[6] M. Vadivukarassi, A. P. Puviarasan, "Sentimental Analysis of Tweets Using Naïve Bayes Algorithm," *World Applied Science Journal*, vol. 35(1), 2017, pp. 54-59.

[7] W. Zhang, and F. Gao, "An Improvement to Naïve Bayes for Text Classification," Procedia Engineering, vol, 15, 2011, pp. 2160-2164.

[8] X. Zhang, Y. Zeng, X-B. Jin, Z-W. Yan, and G-G. Geng, Boosting the Phishing Detection Performance by Semantic Analysis, in IEEE International Conference on Big Data (BIGDATA), 2014, pp. 1063-1070.

[9] https://www.analyticsvidhya.com/blog/2017/12/fundamentals-of-deep-learning-introduction-to-lstm/