Email Phishing Detection using Deep Learning

Aatrayee Bhattacharjee
{a56bhatt}@uwaterloo.ca
University of Waterloo
Waterloo, ON, Canada

Introduction

In today's world, most of us rely on online resources for majority of our activities, such as shopping, watching movies, banking and education. Our online presence makes us susceptible to malicious activities from attackers. Phishing is one such malicious activity where attackers send fraudulent emails or messages to trick people into sending out their sensitive information. One example of a phishing mail is where people are given false warnings about their account credentials expiring if the link in the email is not clicked to enter the person's login details. To look more genuine, these attackers sometimes pose as big companies with subtle change in the email domain or as a friend or family member. They add an emotional incentive to the email such as, limited time to perform the task specified in the email. People who are not aware of phishing emails tend to click on the URL given in these emails and are taken to malicious websites. They are prompted to fill in their login credentials and this information is used by the attackers for malpractices such as identity theft or monetary gains.

Phishing attacks are extremely prominent, especially in the corporate world. Big corporations conduct mandatory annual training to help their employees identify phishing attacks. Some conduct social engineering tests as well, to check if the employees are aware of signs of phishing. Failing the test might limit the employees mailbox features. Some common features in phishing emails are: incorrect spellings, poor grammar, uncommon phrases, suspicious URLs. Though training employees on detecting phishing attacks is important, due to excessive emails received by them all, people may not notice the phishing email and fall prey to it. Attackers are also upgrading the content of phishing emails to make it look more genuine. One solution to this problem is to have automatic phishing email detectors in mailboxes that would prevent such malicious attacks.

According to the Anti-Phishing Working Group's report for 2020, the first quarter had 165,772 phishing sites detected, second quarter had 146,994 phishing sites detected and the third quarter had 571764 phishing sites detected (APWG 2021). This shows that the number of attacks drastically increased during the third quarter compared to quarter one and two. There was an increase in COVID-19 related phishing emails such as health packages. Many of the phishing sites were SSL protected. This makes us aware of the speed with which phishing attacks are increasing and we must come up with an efficient solution to prevent further attacks.

There are many implementations of using machine learning techniques or neutral networks to detect phishing emails. However, the studies take place using different databases and we cannot make a proper evaluation on which technique is more efficient. The research question of this project is: Will neural networks like Recurrent Convolutional Neural Network or LSTM perform better than Support Vector Machines in detecting phishing emails using the same database and similar features?

Emails tend to have numerous features such as sender ID, sender domain, subject, body, etc. In this study, we will compare the performance of Recurrent Convolutional Neural Network and LSTM with the performance of CS-SVM using a combination of important features to efficiently detect phishing emails. The performance metrics used to compare the results are: accuracy, True Positive Rate, False Negative Rate, True Negative Rate and False Negative Rate. The dataset to be used is approximately 3000 phishing URLs from PhishTank (OpenDNS 2016) and Apache SpamAssassin (Mason 2005) for spam and non-spam emails. The study will take place using Python and its existing libraries such as scikit-learn, keras and tensorflow.

Related Work

Various techniques have been used in the past to detect phishing emails. Before moving to approaches that use machine learning to detect phishing emails, let us discuss few other approaches:

White-list: The white-list contains a list of websites that the user regularly visits. The approach works well for zero day phishing attacks and has a low false positive rate. The issue with this approach is, if a user visits a new website that is not regularly used, the white-list considers it as a phishing URL. Due to this, white-lists have a high false negative rate. Another disadvantage of white-list is that it has to be manually updated and maintained. (Li, Helenius, and Berki 2012)

Black-list: The black-list approach is a popular and simple approach to detect phishing websites. Most widely used web browsers such as Internet Edge and Google Chrome use a black-list to prevent users from opening malicious websites. In the black-list approach, a URL is compared with a black-list database that is already predefined. Blacklist tend to have low false positive rates. The disadvantage of black-list is that they do not work well on zero day phishing attacks as the black-list is already predefined and it is not possible for it to contain newly made URLs. (Sheng et al. 2009)

A zero day phishing attack is an attack that harms any component of software the developer is not aware of and must be fixed immediately to prevent further harm.

Network-Based approach interferes TCP or UDP to block malicious IP addresses but this approach is expensive and time consuming. (Gupta et al. 2017)

Content based approaches are most effective as they have a high accuracy rate, but they require large amount of training data to be efficient. One study that uses Cuckoo Search-SVM outperforms regular SVM as Cuckoo Search helps pick optimal parameters for Radial Basis Function instead of the default SVM parameters. This method has a higher True Positive Rate, False Negative Rate, 99.52% accuracy and uses 23 features to train the model on a dataset. (Niu et al. 2017)

A phishing detection system based on LSTM Recurrent Neural Network has an accuracy of 99.17% compared its this baseline CNN with 97% accuracy. This system takes into consideration the URLs in the email instead of considering both the URL and text, which is taken in the previous study. (Chen, Zhang, and Su 2018)

Systems that use dynamic evolving neural networks along on reinforcement learning use 55 features including both the text and URL which is appropriately selected based on the dataset used. Using reinforcement learning (mean square error) the number of nodes of the input, output and hidden layers of the neural network are selected according to the dataset and the neural network is trained efficiently. The system has 98.63%, 99.07%, and 98.19% of accuracy, True Positive Rate and True Negative Rate respectively.(Smadi, Aslam, and Zhang 2018)

In the study by (Moradpoor, Clavie, and Buchanan 2017), a custom neural network model is implemented. Sufficient results are obtained using this model.

(Fang et al. 2019) used recurrent convolutional neural networks with the attention mechanism and had an accuracy of 99.48% and a false positive rate of 0.043%. It used both the email header and the email body at both character and word levels and used an unbalanced dataset to mimic real work conditions.

(Paliath, Qbeitah, and Aldwairi 2020) compared five machine learning techniques (Support Vector Machines, Naive Bayes, Rough Set Theory, Random Forest and Random Tree) and a simple neural network. In this study, the neural network outperformed the other five machine learning algorithms with an accuracy of 99.76%, The performance of SVM was close with an accuracy of 99.75% and the other four machine learning techniques had an approximate accuracy of 98% each.

Clustering is also used to detect phishing emails. The algorithm puts vectors together, if the vectors are less than a certain threshold. Using this method it was discovered that 90% of the sites detected were replicas of older phishing sites. The method has a low False Positive Rate at 0.08%. The author mentioned that though this method works well, since attackers are improving their malicious acts rapidly, this method might not work well when it is finally deployed. (Cui et al. 2017)

(Feng and Yue 2020) created four Recurrent Neural Networks without any manual feature selection, using only lexical features of URLs and attained an accuracy greater than 99%. It also helped create visualization techniques that resulted in the appearance of previously unseen features which can be used with traditional machine learning techniques like Random Forest algorithm to detect phishing.

(Abdelnabi, Krombholz, and Fritz 2020) used triplet Convolutional Neural Network to help facilitate visual phishing detection technique. It used a database of 9363 screenshots of 155 most trusted websites. It had a similarity metric between two websites even though they did not have same content. This approach is different from the other approaches that rely on text and URL data.

References

Abdelnabi, S.; Krombholz, K.; and Fritz, M. 2020. Visualphishnet: Zero-day phishing website detection by visual similarity. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, 1681–1698.

APWG. 2021. Apwg phishing trends reports. *Anti-Phishing Working Group, https://apwg. org/trendsreports*.

Chen, W.; Zhang, W.; and Su, Y. 2018. Phishing detection research based on lstm recurrent neural network. In *International Conference of Pioneering Computer Scientists, Engineers and Educators*, 638–645. Springer.

Cui, Q.; Jourdan, G.-V.; Bochmann, G. V.; Couturier, R.; and Onut, I.-V. 2017. Tracking phishing attacks over time. In *Proceedings of the 26th International Conference on World Wide Web*, 667–676.

Fang, Y.; Zhang, C.; Huang, C.; Liu, L.; and Yang, Y. 2019. Phishing email detection using improved rcnn model with multilevel vectors and attention mechanism. *IEEE Access* 7:56329–56340.

Feng, T., and Yue, C. 2020. Visualizing and interpreting rnn models in url-based phishing detection. In *Proceedings of the 25th ACM Symposium on Access Control Models and Technologies*, 13–24.

Gupta, B. B.; Tewari, A.; Jain, A. K.; and Agrawal, D. P. 2017. Fighting against phishing attacks: state of the art and future challenges. *Neural Computing and Applications* 28(12):3629–3654.

Li, L.; Helenius, M.; and Berki, E. 2012. A usability test of whitelist and blacklist-based anti-phishing application. In *Proceeding of the 16th International Academic MindTrek Conference*, 195–202.

Mason, J. 2005. The apache spamassassin public corpus. *URL: http://spamassassin. apache. org/publiccorpus*.

Moradpoor, N.; Clavie, B.; and Buchanan, B. 2017. Employing machine learning techniques for detection and classification of phishing emails. In *2017 Computing Conference*, 149–156. IEEE.

Niu, W.; Zhang, X.; Yang, G.; Ma, Z.; and Zhuo, Z. 2017. Phishing emails detection using cs-svm. In *2017 IEEE International Symposium on Parallel and Distributed Processing with Applications and 2017 IEEE International Conference on Ubiquitous Computing and Communications (ISPA/IUCC)*, 1054–1059. IEEE.

OpenDNS, L. 2016. Phishtank. *URL: https://www. phishtank. com/index. php*.

Paliath, S.; Qbeitah, M. A.; and Aldwairi, M. 2020. Phishout: Effective phishing detection using selected features. In *2020 27th International Conference on Telecommunications (ICT)*, 1–5. IEEE.

Sheng, S.; Wardman, B.; Warner, G.; Cranor, L.; Hong, J.; and Zhang, C. 2009. An empirical analysis of phishing blacklists.

Smadi, S.; Aslam, N.; and Zhang, L. 2018. Detection of online phishing email using dynamic evolving neural network based on reinforcement learning. *Decision Support Systems* 107:88–102.