Using Machine Learning Algorithms to detect Phishing scams through Domain Name Detection

Isaac M. Berlin

Machine learning has helped solve many problems in our world today. Phishing scams use a mixture of computer engineering and social engineering to obtain information from unsuspecting users. This project worked to take the fight away from combating social engineering and attempted to see if there is a link between hyperlinks from phishing scams and if machine learning algorithms can detect these links and help protect us from phishing scams. This study shows that an algorithm can learn the correlation between hyperlink detection and detecting phishing scams.

Keywords: *phishing, machine learning, hyperlink*

Intro

Keeping the internet secure is a problem. Ever since the rise of the internet, the problem of cybersecurity has been a madjor concern. From the Morris Worm in 1988 to DarkSide attack against american oil pipelines that caused the shutdown of forty five percent of the east coast's supply of gasoline. (Sanger & Verma, 2021)

Phishing attacks can be especially problematic because of the mixture of computer engineering with human engineering. The name phishing comes from how the criminal is literally fishing for information such as passwords, credit card numbers, or social security numbers. Phishing attacks have only increased and become more dangerous as the internet has progressed. (Chang & Bergholz, 2008). Machine learning is a new way to attempt to solve the problem of phishing.

Machine learning can be used to negate the human engineering used in phishing attacks and turn phishing into purely a computational issue. A type of unsupervised machine learning called deep learning or a deep neural network is used in this project. Deep learning models use neural network architecture to process large amounts of data.

Methods

The two most important pieces of any machine learning algorithm are the dataset and the language/algorithm. In this case, the dataset used for this research contained twelve rows of columns of data. (Nagariya, 2020) The data consisted of the domain of the website, the page ranking, if there is an IP address in the link, if the website is valid according to the url registration, how long the site as been active, the length of the url, if the link has an @ symbol, if the url has a dash, if the url has double dashes possibly indicating a redirect, the length of the domain name, the number of subdomains, and finally whether the domain is a phish or not. These labels help to identify phishing because they include some of the telltale signs that a link is a phish.

To parse all these different columns into information usable in a machine learning algorithm the steps that follow are taken.

1. The first column was deleted to remove text strings from the dataset
2. The data was split into tags (indicators about the data) and labels (if the site was a phish or not)
3. The data processing is implemented to scale the data from zero to one

The first step happens because machine learning doesn't play well with text strings unless built on natural language processing. The second step is implemented so that the machine learning algorithm will not factor in whether the link is a phish when trying to determine if it is a phish. The third step is implemented in Python through a mix of pandas and scikit-learn.

The algorithm is written using pytorch and uses ReLU to calculate the neural network neurons. During this process the dataset is split into two different sets of data. These sets are for training and testing the algorithm respectively. The function to split the data is set to split the data fifty fifty but can be scaled to any different percentage in case of a reduction in size of the dataset. The model uses the training data to learn by adjusting how different tags factor into the confidence of phishing or non phishing. The testing data uses the same algorithm as the training data except it does not factor the results of the test into the training.

Figure A

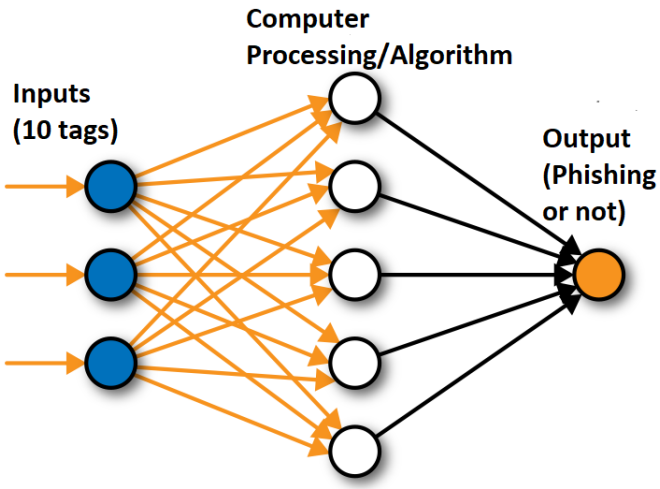


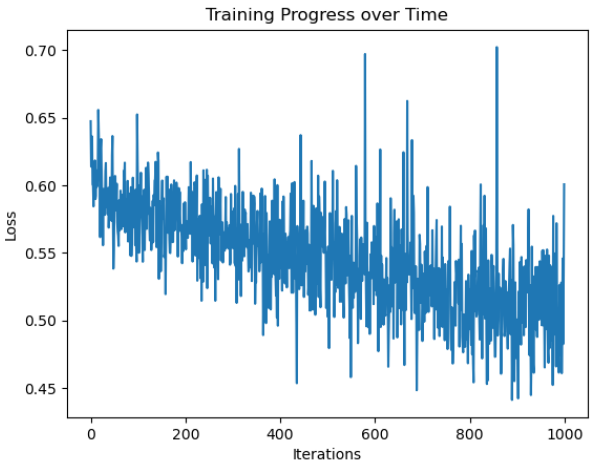
Figure A represents how a single row of data is run through either the training or testing phase to predict phishing or non phishing. This figure shows how the algorithm takes the ten value inputs, runs these values against the past training data values, and makes a prediction. The prediction is not a one or a zero but instead a number that represents the confidence of the algorithm in a gradient from zero to one. (Castrounis, n.d.) This is how a loss function works. The loss function calculates the error of the model based on how far the predicted outcome is from the actual outcome. For example, if the predicted outcome is 0.75 and the actual outcome is 1.00 then the loss would be 0.25. Loss also helps determine if and when the algorithm is improving. If the loss is steadily decreasing then the algorithm must be learning at a steady rate.

Data visualization is a powerful tool that the algorithm uses to make sense of the hundreds of loss numbers. The API used to do this is matplotlib. All the graphs used in this paper are created through matplotlib.

Results

Figure B

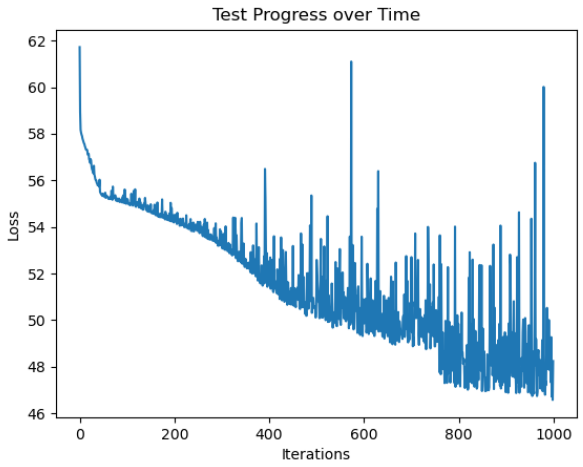
The graph above (figure B) shows the loss of



the training over one thousand iterations of the algorithm. This graph shows a steady decrease of the loss starting at around 0.63 and ending at around 0.54.

Figure C

Figure C is a graph of the loss of the testing



over a period of one thousand iterations. This graph has a jump in learning at around one hundred iterations and another jump in learning at around seven hundred and fifty iterations. This graph, similar to figure b, also shows the loss trending downward.

Discussion and Conclusion

The result of this study looks promising. The overall shrinking of the loss in both the testing and training phase signifies that the algorithm is learning and getting better at predicting phishing.

While the algorithm is learning fine as is, there are a few improvements that can be made to better the algorithm. The dataset is the main limiting factor of the algorithm. Even with an impressively large dataset there is still room for improvement. The dataset could have been improved with added columns of data such as the country of origin of the hyperlink or date the email was sent. The dataset could also expand to encompass more than just the hyperlink. For example, this new dataset could contain information such as number of spelling errors or amount of grammatical errors. It could contain something like business sent from (many phishing scams attempt to impersonate big businesses like wells fargo bank or verizon phone services). In a perfect world there would be enough time to go through hundreds of thousands of phishing emails and create a new dataset that would run through the algorithm but unfortunately that is not the case.

References

Castrounis, Alex. (n.d.)AI, *Deep Learning, and Neural Networks Explained*. Innoarchitrch.

Chang, Jeong Ho; Bergholz, André . 2008. *Improved Phishing Detection using Model-Based Features*. The Fifth Conference on Email and Anti-Spam

Nagariya, Aman. 2020. *Phishing websites Data Classifying Phishing websites from Legitimate ones*. Kaggle

Sahingoz, Ozgur Koray; Buber, Ebubekir; Demir, Onder; Banu. 2019. Diri *Machine learning based phishing detection from URLs. Expert Systems with Applications* Volume 117, Pages 345-357

Sanger, David E.; Verma, Pranshu. 2021. *The F.B.I. confirms that DarkSide, a ransomware group, was behind the hack of a major U.S. pipeline.* New York Times. https://www.nytimes.com/2021/05/10/us/politics/dark-side-hack.html