URL Phishing Prediction

Preliminary Results

Jordan Waldroop
Leeds School of Business
Univerisity of Colorado – Boulder
Boulder, Colorado
jordan.waldroop@colorado.edu

Jack Beck
Leeds School of Business
University of Colorado – Boulder
Boulder, Colorado
jack.beck@colorado.edu

***Abstract*—The goal is to work on improving a current URL phishing detection algorithm, which has the machine learning goal of creating a model that can predict whether or not a given URL will lead to a phishing website, based on the characters within the URL itself.**

Keywords*— *URL, phishing, detection, malicious, features, classifier

I. Introduction

This project aims to expand upon the work of a previous research study from the scientific journal "Data in Brief", with the goal of creating classifier models to predict phishing URLs. The initial study provides a dataset that was created using an algorithm in the form of a python script which is designed to extract features from website URLs. The algorithm organizes the features extracted from the URL into four categories depending on which part of the URL is being looked at, with the categories being: domain, directory, file, and parameters. With these categories combined, each URL has been broken apart into 112 features. This algorithm was applied to legitimate website URLs as well as known phishing URLs which were obtained from www.phishtank.com. The extracted features from the legitimate and malicious URLs were combined to make a comprehensive dataset. Using this dataset, we have trained and tested two classifier models, with the goal being to create a model using the extracted features that can predict whether a given URL is a phishing site. The two classifiers are a random forest classifier and a neural network. The preliminary results for both classifiers are promising and are described below.

II. Models and Evaulation

A. Random Forest Model

The random forest classifier created for preliminary testing used the full dataset provided by the previously mentioned journal. This iteration of the model uses all 111 independent features provided by the dataset to predict the dependent binary target feature, with a 1 in the target meaning that the URL is a phishing site and a 0 meaning the URL was legitimate. Before training the classifier, the dataset was separated into training data and testing data such that 75% of the dataset was used as training data, while the remaining 25% was used as testing data. The specifications used for the random forest were an estimator count of 200, with each tree having a max depth of 10 to avoid overfitting. Using these specifications, the model was trained and then validated, which produced promising results. The models mean accuracy on the testing data was calculated to be 0.953. The model has an average precision of 0.981 and an F1-score of 0.933. From here, an ROC curve was calculated to further evaluate the model. Like the previous metrics, the results from the ROC curve were extremely promising, with AUC of the ROC being 0.990 (see Figure 1). Based on this metric, it can be assumed that the classifier is correctly predicting both positive and negative classes for the majority of the observations in the data. Following this, a precision-recall curve for the model was calculated. The AUC of the precision-recall curve was calculated to be 0.981 (see Figure 2). Based on these evaluation metrics, the model's performance is very promising, however, there are some concerns and potential limitations that will need to be considered going forward.

B. Neural Network Model

The initial neural network model that has been built for preliminary testing is a Tensorflow Keras Sequential neural network model. The initial model uses all 111 features to make probabilistic predictions on if a URL can be classified as a phishing URL. The neural network includes the input and output layers, along with three hidden layers (see Figure 3 for layer summary). Between each layer there is a 20% dropout rate to allow for better model generalization and to avoid overfitting. The input layer and each hidden layer have a relu activation, with the output layer having a sigmoid activation. The model uses the Adam optimizer with binary cross entropy as the loss measure, with having binary accuracy and AUC as additional measured metrics. These model specifications have been chosen because the target variable is a binary classification. In addition, the model sets aside 15% of the training set for validation during the model fit, while the validation X and y sets that were originally created are not used until scoring. The model fit was run with 15 epochs, with the model seemingly learning more significantly during epochs 8 and 9. When using the evaluate function to test the model on the validation holdout observations, the model returned with a binary cross entropy of 22.861%, binary accuracy of 90.547%, and an AUC of 96.429%. In plots of both the AUC and binary accuracy, the sigmoid activation of the output layer is clearly working correctly, though there is obviously room for improvement (see Figures 4 & 5 for AUC plot and binary accuracy plots). The model returned promising predictive results; using 0.5 (from a range of 0-1) as the threshold, there were 14,154 of the 22,162 validation URLs that the model predicted to be legitimate and 8,008 predicted to be phishing.

III. Future Development Goals

The academic study that we have based this project off of did not implement models in their research, but was primarily focused on the gathering of datasets to be used in the feature extraction process and the implementation of that feature extraction process. This feature extraction process used is based off of another academic study: An assessment of features related to phishing websites using an automated technique by Mohammad, et al. After reviewing the basis of Mohammad et al.'s research publication, we agree that the theory and logic behind the process is sound. We would like to implement our own feature testing using the same logic that Mohammad et al. used to develop their feature extraction process. This will consist of taking the URL attributes and using models to test on each categorical subset to determine the most predictive attribute groups. These groups consist of: domain properties; URL directory properties; URL file properties; URL parameter properties; and URL resolving data and external metrics presented. Where the initial random forest and neural network models used all of the features, further testing on the subsets of the parameters will hopefully give more insight into individual/group feature importance in predicting phishing URLs.

A. *Random Forest Classifier*

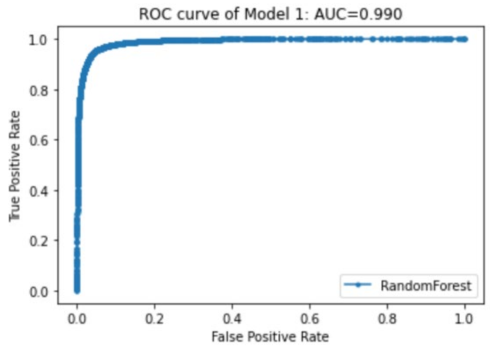
As noted above, while initial results of the models were promising, there are further refinements that could be made to help alleviate potential concerns or problem points within the models. The scoring metrics produced by the random forest model were very good, with several of the metrics being near perfect. This could indicate that the model is suffering from overfitting, which is a common issue for random forest models. Initially, the model's max depth was set to 10 to avoid overfitting, but this value could be lowered further. Going forward, more robust cross-validation metrics will be calculated to evaluate the overfitting in the model. Another thing to consider is that the model could potentially be suffering from target leak, which can be hard to identify given the number of independent features being input into the model. To try to solve this issue, future models will try to incorporate less features. Features will be selected based on their relative importance in the model using sklearn's feature selection function. Furthermore, the features can be subset into the four URL categories described above and in the original data journal. Units

B. *Neural Network Classifier*

The neural network classifier is predicting seemingly well, though there is plenty of optimizing and improvement that can be done. One of the ways that the neural network classifier can be used for feature selection improvement is mentioned above, where a separate neural network model will be created to attempt prediction using feature groups, rather than the entirety of the features in order to find which groups of attributes will be most informative. Other considerations to improve the model is to incorporate batch sizes with the epoch rounds, though the current model is running exceptionally fast, so that may not need to be an aspect of the model. There are many other options to improve/refine the neural network that will simply require testing and comparison. These options could include kernel initializers, changing the validation holdout amounts in both the train-test split and within the model validation holdout. Some other options may include removing or adding additional hidden layers, changing the layer activations or input size, changing the model optimizer (i.e., SGD or Nadam, rather than Adam), or altering the dropout rate between each layer. There is quite a lot of flexibility in the neural network parameters compared to other models/classifiers, which allows for many options for classifier optimization.

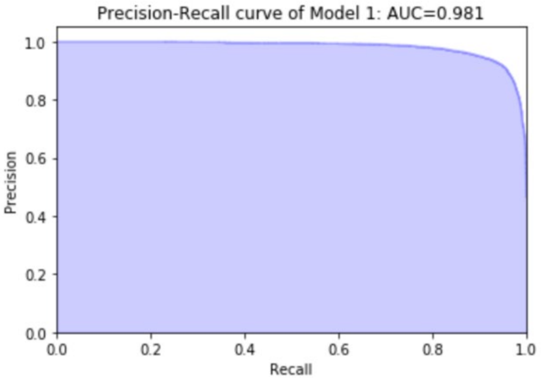
IV. Figures

A. *Figure 1*



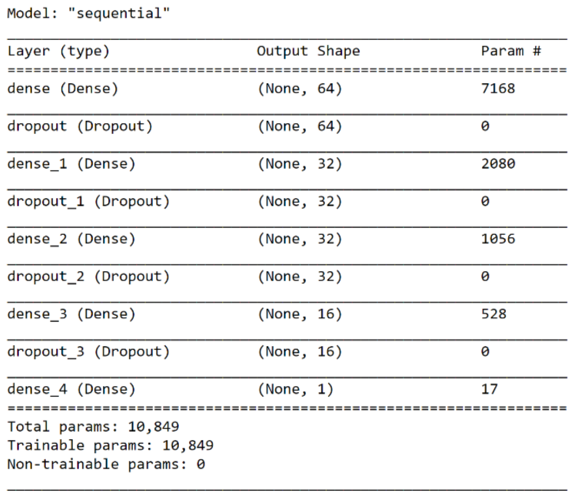
ROC curve of initial Random Forest model/classifier

B. *Figure 2*



Precision-Recall curve of Random Forest model/classifier

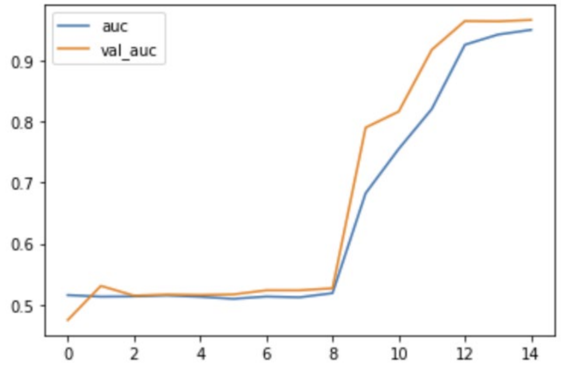
C. *Figure 3*



```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense (Dense)                (None, 64)                7168
_________________________________________________________________
dropout (Dropout)            (None, 64)                0
_________________________________________________________________
dense_1 (Dense)              (None, 32)                2080
_________________________________________________________________
dropout_1 (Dropout)          (None, 32)                0
_________________________________________________________________
dense_2 (Dense)              (None, 32)                1056
_________________________________________________________________
dropout_2 (Dropout)          (None, 32)                0
_________________________________________________________________
dense_3 (Dense)              (None, 16)                528
_________________________________________________________________
dropout_3 (Dropout)          (None, 16)                0
_________________________________________________________________
dense_4 (Dense)              (None, 1)                 17
=================================================================
Total params: 10,849
Trainable params: 10,849
Non-trainable params: 0
_________________________________________________________________
```

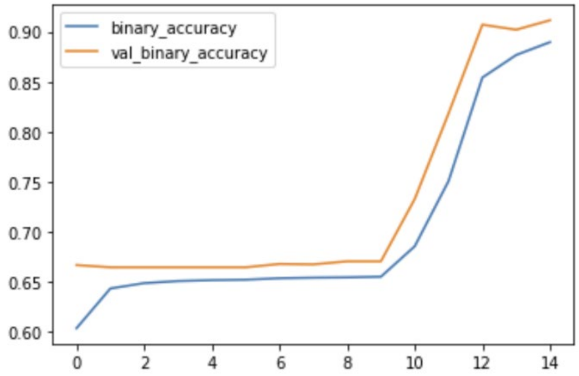
Model summary of initial neural network classifier

D. *Figure 4*



Training AUC vs. validation AUC of neural network.

E. *Figure 5*



Training binary accuracy vs. validation binary accuracy of initial neural network classifier

GITHUB REPOSITORY

https://github.com/jwaldroop/phishing-url-project.git

REFERENCES/RELATED WORK

[1] Grega Vrbančič, Iztok Fister, Vili Podgorelec, "Datasets for phishing websites detection," *Data in Brief*, vol. 33, 2020, accessed at: https://doi.org/10.1016/j.dib.2020.106438.

[2] R.M. Mohammad, F. Thabtah, L. McCluskey, "An assessment of features related to phishing websites using an automated technique", Internet Technology And Secured Transactions, 2012 International Conference for, IEEE (2012), pp. 492-497, accessed at: http://eprints.hud.ac.uk/id/eprint/16229/1/The_7th_ICITST_2012_Conference_-An_Assessment_of_Features_Related_to_Phishing_Websites_using_an_Automated_Technique.pdf.

[3] G. Vrbancic, I.J. Fister, V. Podgorelec, "Parameter setting for deep neural networks using swarm intelligence on phishing websites classification," Int. J. Artif. Intell. Tools, vol. 28, no. 6, 2019, accessed at: https://www.worldscientific.com/doi/abs/10.1142/S021821301960008X.