Phishing Websites Detection using machine learning

Muhammad Tamjid Rahman

1. Introduction

Phishing costs Internet users billions of dollars per year. It refers to luring techniques used by identity thieves to fish for personal information in a pond of unsuspecting Internet users. A lot of techniques are used by phishers like phishing software, spoofed e-mails to steal financial account details and personal information. Phishers sometimes creates a web page similar to another which can attract a lot of customers' attention where they might provide their personal information and get caught by the scheme.

Phishing continues to be one of the rapidly growing classes of identity theft scams on the internet that is causing both short term and long-term economic damage. There have been nearly 33,000 phishing attacks globally each month in the year 2012, totaling a loss of $687 million. The United States continued to be the top country hosting phishing sites during the third quarter of 2012. This is mainly due to the fact that a large percentage of the world's Web sites and domain names are hosted in the United States. Financial Services remains to be the most targeted industry sector by Phishers[1].

Recently, there have been several studies that tried to solve the phishing problem. Some researchers used the URL and compared it with existing blacklists that contain lists of malicious websites, which they have been creating, and there are others that have used the URL in an opposite manner, namely comparing the URL with a whitelist of legitimate websites. The latter approach uses heuristics, which uses a signature database of any known attacks that match the signature of the heuristic pattern to decide if it is a phishing website. Additionally, measuring website traffic using Alexa is another way that has been implemented by researchers to detect phishing websites.[2]

2. Data

The data consist of a collection of legitimate and phishing website instances. Each website is represented by the set of features which denote, whether website is legitimate or not. The total number of instances is 88,647. The number of legitimate website instances which is labeled as 0 is 58,000. Number of phishing website instances which is labeled as 1 is 30,647. Total number of features in our datasen are 111.[3]

There is no missing value in our dataset but there are some variables with constant value. Since constant variables isn't going to help in our work, we removed those variables. Then we have 98 features and 1 dependent variable.

3. Methods

To handle the classification problem, We'll use two types of machine learning methods. The first machine learning model is the ensemble learning method random forest classifier and the second is feedforward Neural Network Method. We will compare the performance of these two models.

3.1. Random Forest Method

Random forests or random decision forest is ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision tree at training time. For classification tasks, the output of the random forest is the class selected by most trees. For regression tasks, the mean or average prediction of the individual trees is returned. Random decision forests correct for decision trees' habit of overfitting to their training set. Random forests generally outperform decision trees, but their accuracy is lower than gradient boosted trees. However, data characteristics can affect their performance.[4]

We normalized the variables before applying random forest algorithm.The number of trees in this model is 100. Gini index function has been used to measure the quality of a split. The number of minimum sample split is 2. The minimum number of samples are required to be at a leaf note is set to be 1. The number of jobs (n_jobs) to run in a parallel is set to -1 means using all processors. The maximum depth (max_depth) of the tree is set to none means nodes are expanded until all leaves are pure. Output Results for Random Forest Model are shown in table 1.

3.2. Neural Networks

Deep feedforward networks, or feedforward neural networks, are computing systems inspired by biological neural networks. It is an adaptive system that changes its structure during learning. Feedforward NN is one of many different types of artificial neural networks. For phishing detection problem, which is basically a classification problem, we'll

Table 1. Output Result for Random Forest Model

Parameter	Value
fit_time	459.437823
Score_time	29.886651
test_accuracy	96.894424
test_recall	95.637428
test_specificity	97.558620
test_gmean	96.593154
test_f1	95.514162
test_roc	96.598024

Table 2. Hyperparameters for Neural Networks

Parameter	Value
Input_layer unit	{32, 64, 128}
Hidden_layer1 unit	{32, 64, 128}
Hidden_layer2 unit	{32, 64, 128}
Hidden_layer1 activation	{ReLU, sigmoid}
Hidden_layer2 activation	{ReLU, sigmoid}
Optimizer	{rmsprop, Adam}
Batch_size	{512, 1024, 2048}

use multilayer feedforward NN. Feedforward neural networks are called networks because they are normally represented by composing together many different functions. The model is associated with a directed acyclic graph describing how the functions are composed together. Feedforward NNs are better at modeling relationships between inputs and outputs.

Our simple feedforward neural network model consists of one input layer, two hidden layer and one output layer. All the layers are dense layer. The number of computational units in the input and output layers corresponds to the number of inputs and outputs. Different numbers of units in the hidden layers are tested in the following experiments. Rectified Linear Unit(ReLU) and sigmoid are used as activation functions. Batchnormalizations were used before the second hidden layer and before the output layer. For the training method, we attempted Root Mean Squared Propagation(RRSProp) and Adaptive moment estimation(Adam). Since it is a binary classification, so we used sigmoid as activation function in the output layer. [5]

3.2.1. Data Preprocessing

In order to ensure that each feature has an equal impact in the classification process, we custom-standardized the variables before applying neural network algorithm. We transformed the features as followed,

$$custom - standardized\ \ value = \frac{value - mean}{sd + 1}$$

We added 1 to the standard deviation(sd) because there were variables with variances near to zero. We could have removed those variables but we decided not to. We thought they might have impact on the model even a little.
The models are trained on 67.7 percent of the original dataset and evaluated on 32.3 percent of the initial dataset.

3.2.2. Hyperparameter tuning

First we fit a model with fixed parameters. Then we used grid search for fine tuning. We chose grid search in stead of random search because our model isn't very big and grid search is more reliable than random search because grid search uses every combinations of the tuning parameters instead of random combinations in Random search. That's why grid search is relatively slower than random search. We attempted 648 combinations of the parameters. In Table 2 shows the taken parameters.

We chose the parameters based on the val_accuracy. In our test model with Input_layer1 unit= 128, Hidden_layer1 unit=64, Hidden_layer2 unit=64, Hidden_layer1 activation= ReLU, Hidden_layer2 activation= ReLU, Optimizer= Adam and Batch_size=512 gave us the highest val_accuracy. So, we used these parameters for our final model. The models are trained on 67.7 percent of the original dataset and evaluated on 32.3 percent of the initial dataset.

3.3. Evaluation[6]

The evaluation of these models will be done using the accuracy, precision and recall. These measures will also be used in the comparison of the two methods. The accuracy is defined as the proportion of accurately predicted observations and the total number of observation.

$$Accuracy = \frac{tp + tn}{tp + tn + fp + fn}$$

Precision, Recall and Specificity are defined as:

$$Precision = \frac{tp}{tp + fp}$$

$$Recall = \frac{tp}{tp + fn}$$

$$Specificity = \frac{tn}{tn + fp}$$

Here,
tp = true positive
tn = true negative
fp = false positive
fn = false negative

Table 3. Performance metrics.

METHOD	LOSS	ACCURACY	RECALL	PRECISION
RF	–	0.9689	0.9564	–
NN	0.1134	0.9607	0.9479	0.9391

4. Performance

For the random forest model the accuracy is 96.89%. The test recall is 95.64% meaning that 95.64% of observation out of all positive observations has classified as positive. The specificity is 97.56% means 97.56% observations are classified as negative.

For the neural networks model the accuracy is 96.07%, recall and precision are respectively 94.79% and 93.91%. Figure 1 shows the performance in different metrics. From the graph we see that it's stop significant improvement after fifth iteration. All the metrics are close. So, the result is reliable though the data isn't balanced.

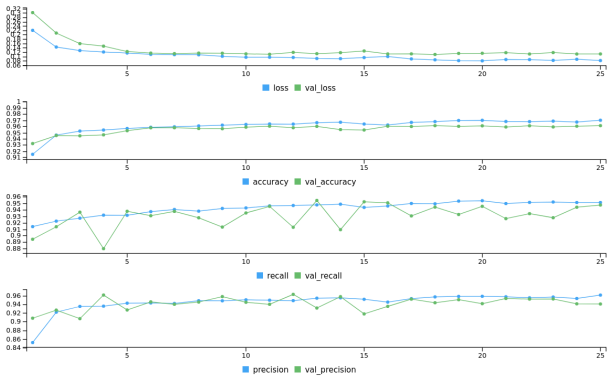

Figure 1. Performance matrics for neural networks.

So, in out case random forest performed better. Table 3 shows the results of two methods.

5. Conclusion

The two machine learning methods correctly classified more than 96%. The method performed the best both in accuracy and recall was the random forest. But the difference is pretty small. Based on the results, the two learning models seem to be able to classify phishing websites very accurately. The results of these two methods are very similar. So, applying these methods on different data would help us to find the difference between them. The sample size isn't very small but larger sample is beneficial for machine learning methods. Then we may find significant difference between these methods. Another thing is that the data is not balanced. A balanced data also helps to find the optimum result. More tuning parameters or extra layers in neural networks model would also help to get better parameter for better performance.

6. References

1. Phishing Trends Report for Q3 2012, Anti Phishing Working Group. http://antiphishing.org.

2. L. A. T. Nguyen, B. L. To, H. K. Nguyen and M. H. Nguyen, "Detecting phishing web sites: A heuristic URL-based approach", 2013 International Conference on Advanced Technologies for Communications (ATC 2013), pp. 597-602

3. Vrbančič, Grega (2020), "Phishing Websites Dataset", Mendeley Data, V1, doi: 10.17632/72ptz43s9v.1

4. https://en.wikipedia.org/wiki/Random_forest

5. DL: Goodfellow, Ian, Bengio, Yoshua, and Courville, Aaron. Deep learning. MIT Press, 2017.

6. https://en.wikipedia.org/wiki/Precision_and_recall