Using Machine Learning Classification techniques to detect BotNet Traffic

Sai Datta Vara Prasad Vasireddy
Dept of Computer and Information Science and Engineering
University of Florida
Gainesville, FL
vasireddys1@ufl.edu

***Abstract*—The use of the internet has skyrocketed in the last decade. This evolution was aided by the ease of access to cell phones and internet connectivity. Many online resources have been made accessible to consumers because of the increased use of the internet. Along with the positive, there is an increasing amount of evil. There have been several cyber-attacks on web sites devised. These attacks appear to be regular requests to the sites and are difficult to differentiate from them. With advances in computing power and machine learning algorithms, these anomalies and cyber-attacks on web pages can be detected more easily.**

***Index* - BotNet, CyberTraffic, Machine Learning, Classification, DDoS**.

I. *Introduction*

Today cybersecurity is on a huge hike with more internet users worldwide. There have been various news technologies that are being evolved every day for stopping possible cyberattacks. With the increase in technologies and easy access to the internet, there are different cyberattacks being illustrated in n ways. But the big corporates are still using the old traditional ways for not being prone to cyberattacks. That is an intrusion detection system (IDS). IDS can be further classified into signature-based IDS and behavioral-based IDS. From the name itself, we can understand that signature-based IDS demonstrates the signatures that are malicious and that are more common. The signature-based IDS generally flags the signatures and removes them as they are malicious attacks. The drawback with this is it cannot stop all the attacks that are malicious. Only the know attacks with malicious signatures are flagged by this method. For instance, if a new attack with a different signature is generated then, that is not stopped by this method. Also, from the name again the behavioral-based IDS generally acts based on the behavior of the attack. So, whenever the behavior of the attack is different from normal requests, the flow is flagged and reported as a malicious attack. Similarly, the above method even the behavioral-based IDS has several drawbacks.

With the increase in new tools and technologies in the market, there is a new type of behavioral attack that can be generated. Behavioral-based IDS only could stop the attacks that are of known behavior. So, to be in trend with the new attacks new strategies and methods using machine learning are being used.

Supervised and Unsupervised learning are the two well-known machine learning strategies that are being used. When the data is flagged, we have the chance to use supervised learning strategies, and to divide the flows into various categories we can use unsupervised strategies.

My project speaks about the various training data that is being generated by using machine learning strategies. In this project, we use personal information from various requests and apply traditional methodologies. We see how using these methodologies can create various threats and disturb the flow creating various breaches in privacy.

Section 2 here speaks of the problem and provides a solution briefly. Section 3 has the evaluation and the main context behind this project followed by section 4, where I present my work in detail. Finally, the last section summarizes my whole view on these methodologies, and I provide my conclusions.

II. Project Approach

In a Network, the network data flows from one system to other system, for this project we use different machine learning approaches that differ botnet data flow from regular data flow.

-Choosing appropriate dataset

-Cleaning and extraction

-Applying machine learning approaches on the selected features.[4][5]

A. Choosing appropriate dataset:

We need to select a dataset with suitable size to make reliable predictions in-order to get accurate results. For this project we move on with CTU [1] dataset which promises to give reliable results. This data includes all the information regarding the various forms of traffic that come to different websites. The NetFlow that is being used in this project includes various request attributes such as source, destination, request length, number of bytes, and a mark that indicates whether the request is an attack or normal traffic.

B. Cleaning and extraction:

Various features from the NetFlow dataset can be extracted after it has been collected. As previously mentioned, the NetFlow dataset contains all essential traffic information. This traffic data will be translated into numeric vectors, which will be fed into the classifiers as input. These feature vectors which include, but are not limited to, the amount of time each request takes, the number of requests coming from a single source, the number of requests going to

a single source, and so on... The features that contribute the most to the classification process will be chosen from all the features extracted from the dataset. Although we can select suitable features, but they might not be well enough to predict our results up to good extent. In-order for us to achieve maximal throughput with the chosen machine learning approaches we need to extract some more features like mean, median, co-variance etc., of the dataset.

C. Applying machine learning approaches on the selected features:

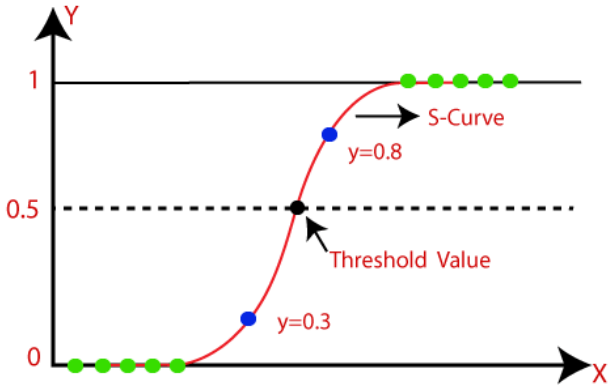
These feature vectors are then given to the classifiers as a test input [3][4][5]. The role of defining botnet traffic can be achieved using a variety of classifiers. Some of the machine learning approaches that will be used to complete the task include: logistic regression, support vector machines, random forest classifiers, and artificial neural networks. After all the classifiers have been trained on the dataset, their output will be evaluated, and the classifiers that perform best on the dataset will be used for any subsequent tasks. Once the best classifier has been determined, it can be used to identify botnet traffic in real-time data coming from live online sites. The prior step in the process i.e., extracted features are now added to machine learning models.

1.Logistic regression

Logistic regression is used for classification problem in machine learning field, here in this project we need to classify between botnet data and network data. This approach basically needs to have input label and an output label. Output label will be a decider for whether the outputted results botnet data or network data. It clearly shows us that this machine learning approach is a supervised learning technique. It generates a range of values among 0 and 1. This is commonly used in classification problem to divide data into two categories. The results of a linear regression are calculated using sigmoid functions, which vary from 0 to 1. The classifier is trained using the labels from the training results.

By using regressor, the key purpose of this training process is to reduce the cost function when adjusting the inputs. The metrics produced throughout this phase are perhaps the most accurate for training dataset and yield better results for test data.

Below is an example for logistic regression model, we can see that there is a threshold value, and we usually consider that threshold value as a benchmark for our classification. We classify the results in binary classes, data that maps above the threshold value is considered to be in one class and the data that maps below the threshold value is classified to the other class.



Evaluation and Results:

Logistic regression is a simple method that predicts the outcome using a linear number of factors. The model is learned to determine the weights of each feature, and the weights are determined and adjusted using the gradient boosting algorithm. Cross validation is used to determine which features subset is best for training. Cross validation is performed on the subsets once more to extract the best functions. The model employs a gradient boosting algorithm to determine and modify the weights of each algorithm.

The model has an accuracy of 97.52% on both train data and test data.

Training:

Precision: 77.22%

Recall : 95.06%

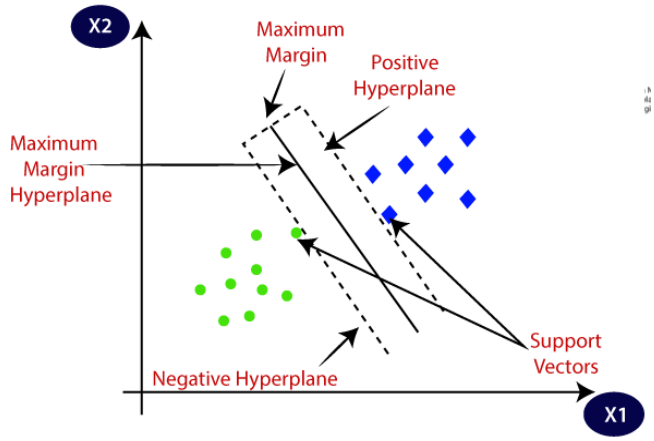
Testing:

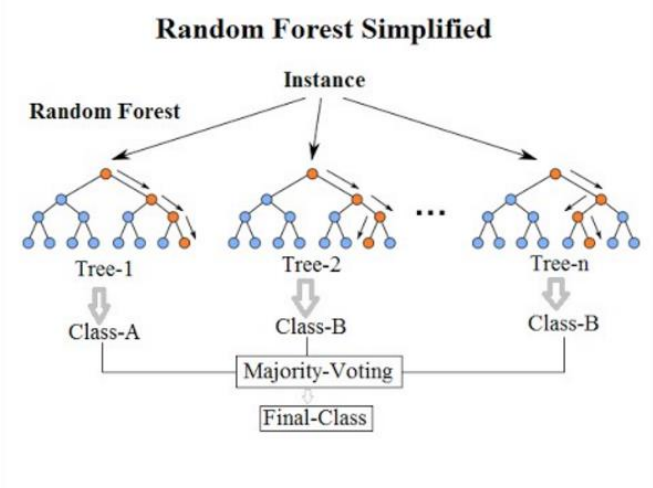
Precision: 77.32%

Recall: 92.43%

2.Support Vector Machine

Logistic regression technique in machine learning classifies the dataset into two labelled classes whereas the support vector machine technique classifies into n distinct labelled classes, thus creates different output classes in n dimensions which result the output as a hyperplane and SVM is also a supervised machine learning approach like logistic regression. Support vector machine optimal way classifies the input into specified groups by constructing a support vectors (vectors that form a hyperplane). Like logistic regression SVM is also used for classification problems and as well as for regression problems. The hyperplane is straight line in 2-dimensional space. The hyperplane would be a p-1 dimensional sub space in p-dimensional space likewise a hyperplane would a 2-dimensional subspace in a three-dimensional space. SVM classifies data by identifying the hyperplane that provides the maximum the difference between two types. The SVM model examine the 2-dimensional straight line first and then tests if it matches the results. Then it continues including another dimension until it achieves a reasonable level of precision. By splitting the dataset into its groups using the hyperplane, this procedure produces an ideal hyperplane.





Evaluation and Results:

The best one of the datasets from the cross validation upon the subsets of the whole dataset will be used to train the classifiers. The test dataset is then used to correctly divide the results into their own groups using this hyperplane. SVM divides the data into classes and uses kernels to convert it into space. It then finds the right hyperplane that best matches the data.

The model has an accuracy of 92.5% on both train data and test data.

Training:

Precision: 98.5%

Recall : 85.04%

Testing:

Precision: 98.99%

Recall: 85.01%

3.Random Forest

Random forest consists of numerous decision trees which acts as key players in the random forest. Each decision tree has number of patterns to predict, and each pattern will result in favor to unique class prediction. The more favorable class prediction among all the patterns in numerous decision trees will be our model prediction. We input the training data to decision trees in the random forest and the input data is classified with the features that we have selected and extracted. Each selected feature acts as way to achieve different class predictions. Thus, decisions tress is simply formulated as classification trees. By using different individual features in different decision trees in random forest approach we might get low accuracy with some decision trees, moderate and high accuracy for other decision trees. So, we take mean of each decision tree results in-order to achieve overall high performance. To achieve high accuracy for the selected model we should train our decision trees as much as possible, but when an anomaly in the training data would be resulted as one of the major drawback to the random forest model as it disturbs the well trained decision trees.

Evaluation and Results:

To find the output, this approach employs many decision trees. Decision trees compare features through training phase, and each tree takes a set of features. To mark each element, the random forest classifier employs its own feature of determining feature value. The features that are closest to the display are the ones that are wanted. The importance of the features with respect to the label can be seen in the graph shown.

The most important features are put first. 13th feature (Tot Bytes) being most important and 7th being least.

The model has an accuracy of 100% on train data and 97.5% on test data.

Training:

Precision: 100%

Recall : 100%

Testing:

Precision: 100%

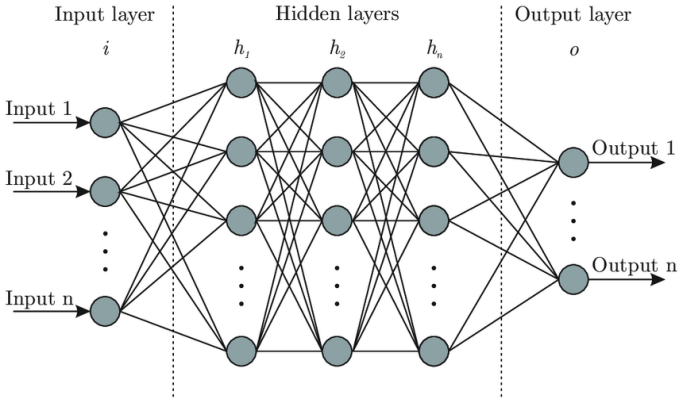
Recall: 95.12%

4.Artificial Neural Networks

The idea behind artificial neural network is how the human brain operates, examine, and produce the results. This artificial neural network, trains on data using neurons. The input data is compounded by this weight on the neurons. These links enable changed data to be transferred between one neuron to the other.

The neurons weights are calculated. With the increase in iterations, we also see an increase in the hidden layers. The network to train time also increases. Input is generated and modified by giving the float numeric format and we transfer the input weight of previous layer neurons to other neurons by multiplying it with the weight on that each neuron possess.

With increase in hidden layers there will e lack of accuracy with the results. So, to get accurate results we need to make sure that only required number of hidden layers serving the model.



Evaluation and Results:

Neural Networks hyper tune the parameters during training. In this project, we chose a neural network with 2 hidden layers. The first layer with 256 neurons and other with 128 neurons. These networks are trained in batch normalization procedure. The activation function used is ReLU. We iterate 20 times and sigmoid function acts as function for output layer. The improvement of precision, recall and f1 scores with the iterations can be seen in the figure.

IV. Related Work

There are a few limitations for the customary botnet detection software's as all of them are controlled by request's behavior and signature that arrive at the server. These models depend on the pre-defined data. So, when there is a new behavior or when the hackers generate any new signatures these conventional botnet detection software's does not work and the necessity of new algorithms and the invention of new methods to detect bad traffic is increased.

In recent years, the availability and use of large datasets along with the development of new algorithms and the improvement and growth in computing power led to an unprecedented surge in machine learning. These machine learning algorithms are being applied in all the fields including cyber security. Machine learning algorithms allow cyber security systems can analyze patterns and learn from them to prevent attacks and respond to changing behavior. These models require some previous data to train to detect patterns. Usually, state-of-art models use the data packets from live websites which come along with private data within them. But for our purpose where we use machine learning models, private data is not needed and only some major features would be sufficient. NetFlow data collection is a method which is first introduced by cisco is used for the collection of these desired features. This, when integrated to any website behaves as an Application Program Interface to collect all the essential features like the protocol that is used, start time, Duration of communication, IP address of source and destination, port address of source and destination, type of source and destination service, total number of bytes that come from source, the measure of bytes and packets involved in the total communication. Data labelling can also be done from the extracted features as it can be done just by source identification which can be carried if we know the botnet's IP address. The output of the NetFlow does not bring in any private data with the packets like the signatures, packet contents etc.

Although there are many conventional methods to eliminate these attacks, each method has its own limitation and none of them completely serve the purpose. So, machine learning algorithms were introduced to the field of cyber security to obtain efficiency and solve this problem. These techniques are practiced where machine learning models are supplied with the data that is collected manually from various sites. The approach that is explained and followed in this paper is implemented on the data generated by CT University. The output obtained which contains the desired features is particular to this project and does not assure to work on other data or projects.

Previous studies on this area involves only a single classifier to solve the problem. But the approach that is described in this paper makes use of distinct classifiers and compares the results obtained from each of them to identify the classifier the delivers best results for this data. The classifiers used in this approach can also be used on this type of data in the future.

V. Conclusion

On comparison of all the algorithms that are implemented in this project, Random forest has better performance with 97.5% test accuracy. The comparison results of different algorithms are tabulated as follows:
To summarize, this project is an example of how machine learning models can be used in cybersecurity domain to detect botnet traffic. Various models are trained and implemented to identify the best approach. Comparisons between these models is also conducted to aid in any further development of algorithms related to this field.

The classifiers used for this approach delivers better results for the dataset used which does not necessarily mean that the same classifiers will work and gives the results with same accuracy for other datasets as well because of the factors that may alter like distribution of data and other features. To get better results training of classifiers which is described in this paper should be done.

This project also helps in preventing the Distributed Denial of Service attack (DDoS) which is one of serious attacks experienced in cybersecurity these days. This attack is done by flooding many botnet requests to the server and keeping it busy serving those requests so that it could not respond the real incoming traffic. The whole purpose of the attacker is to make the online site deny all the incoming requests that it must receive and be idle. In future, this project can be extended, by training the model on different traffic generated by different botnets.

VI. References

1.The CTU-13 Dataset. CTU University. A Labeled Dataset with Botnet, Normal and Background traffic. 2011. url: https://www.stratosphereips.org/datasets-ctu13.

[2] A. Mukkamala, A. Sung, and A. Abraham, “Cyber security challenges: Designing efficient intrusion detection systems and antivirus tools,” in Enhancing Computer Security with Smart Technology, V. R. Vemuri, Ed. New York, NY, USA: Auerbach, 2005, pp. 125–163.

[3] P. Garcia-Teodoro, J. Diaz-Verdejo, G. Macia-Fern ´ andez, and E. ´ Vazquez, “Anomaly-based network intrusion detection: Techniques, ´ systems a

[4] S. X. Wu and W. Banzhaf, “The use of computational intelligence in intrusion detection systems: A review,” Appl. Soft Compute., vol. 10, no.1, pp. 1–35, 2010.

[5] B. Morel, “Artificial intelligence and the future of cybersecurity,” in Proc. 4th ACM Workshop Secur. Artif. Intell., 2011. pp. 93–98.

[6] H. Zhengbing, L. Zhitang, and W. Junqi, “A novel network intrusion detection system (NIDS) based on signatures search of data mining,” in Proc. 1st Int. Conf. Forensic Appl. Techn. Telecommun. Inf. Multimedia Workshop (e-Forensics ‘08), 2008, pp. 10–16.

[7] H. Han, X. Lu, and L. Ren, “Using data mining to discover signatures in network-based intrusion detection,” in Proc. IEEE Comput. Graph. Appl., 2002, pp. 212–217.

[8]. Algorithms that are stated in this paper are referred from https://www.javatpoint.com/machine-learning

[9]. A detailed decision tree theory resources https://towardsdatascience.com/decision-trees-and-random-forests-df0c3123f991.

[10]. provide detailed info about random how random forest model is useful. https://towardsdatascience.com/understanding-random-forest-58381e0602d2.
.