DDOS DETECTION USING MACHINE LEARNING

TEAM MEMBERS

Akshat Garg <ag2193@rit.edu>

Bharadwaj Sharma Kasturi <bk5953@rit.edu>

Megha Gupta <mg9428@rit.edu>

Shaista Syeda <ss7810@rit.edu>

Index

Contents

Introduction

Advances in technology have led millions of people to connect in some form of network and exchange critical data. Therefore, the need for security to protect the integrity and confidentiality of data is rapidly increasing. Efforts have been made to protect data transmissions, but attack technologies to infiltrate networks have continued to be developed simultaneously. Therefore, there is a need for a system that can adapt to these ever-changing attack techniques. In this paper, we have developed a system based on machine learning. Our goal is to find a suitable machine-learning algorithm to predict network attacks with the highest accuracy and develop a system to detect network intrusions using this algorithm. The algorithms compared are Naive Bayes, Decision Tables, K-nearest Neighbors, Random Forest, and AdaBoost. The dataset used to train the model is the KDD99 dataset. The reason I used machine learning is to give the system flexibility. For example, if a new type of attack is developed in the future, you can train your system to predict that attack. There are several types of intrusion detection systems, but our system is a knowledge-based intrusion detection system, also known as an anomaly-based system. Register anomalies and predict that such malicious networks will send alerts in the future. In this way, the network can be disconnected from such connections, and only secure connections are possible.

Problem Description

Information technology has advanced at a breakneck pace in the last two decades. Industry, business, and different aspects of human life all use computer networks. As a result, IT managers must focus on establishing reliable networks. On the other hand, the rapid advancement of information technology has created various problems in the laborious process of constructing trustworthy networks. Computer networks are vulnerable to various threats that jeopardize their availability, integrity, and confidentiality. One of the most widespread destructive attacks is the Denial-of-Service attack.

Data Research

This is the data set used for The Third International Knowledge Discovery and Data Mining Tools Competition.The task was to build a predictive model capable of distinguishing between ``bad'' connections, called intrusions or attacks, and “good” normal connections. The raw training data was about four gigabytes of compressed binary TCP dump data from seven weeks

of network traffic. This was processed into about five million connection records. Similarly, the two weeks of test data yielded around two million connection records.

It is important to note that the test data is not from the same probability distribution as the training data, and it includes specific attack types not in the training data. This makes the task more realistic. Some intrusion experts believe that most novel attacks are variants of known attacks and the "signature" of known attacks can be sufficient to catch novel variants. The datasets contain a total of 24 training attack types, with an additional 14 types in the test data only.There was a data quality issue with the labels of the test data, also there was high imbalance in the data. Finally we learnt a classification model capable of distinguishing between legitimate and illegitimate connections in a computer network.

Literature Review

Being a classification problem, we came up with a supervised learning algorithm. Firstly, the class imbalance problem was overcome using a combination of oversampling as well as undersampling. When the skewness of the data was taken care of, various machine learning models were fed with the data to yield an efficient and usable yield. Algorithms like Regression, Naive Bayes, Decision Trees, Random Forest, Isolation Tree, XGBoost were utilized and conclusions were drawn considering the test scores like recall, precision and accuracy.

Analysis strategy

After studying the distribution of data, we identified that there were different subcategories of DDoS attack based on the layer of the network connection they attempt to attack. The result column had 60% of the normal data. and the rest 40% attack types were unevenly distributed. So, clearly the major challenge was to handle the class imbalance problem. Our approach was to implement different sampling techniques to get the classes balanced. Since, for class imbalance problems, accuracy is not an appropriate metric for model evaluation because the accuracy score would be high and heavily biased towards the majority classes (normal class for KDDCup dataset). Hence our main goal was to identify the minority class (attack sub-classes). So, we focused on precision-recall and FPR(fallout rate) for the evaluation of the machine learning models. In other words, our aim was to minimize a bad connection that gets classified as normal.

Analysis code

1. Data Exploration

Since our objective was to cover the majority of the attack types we combined the test and train data from the KDDCup dataset.

1. Identified the dataset for the null values. We found that there were no null values.

2. Checked for the duplicates in the data frame, around 70% of the data was duplicate so we dropped this.

3. Analyzed the attributes of the dataset and worked upon the numerical and categorical features individually.

2. Feature Selection

After plotting the correlation matrix, there were a total 9 pairs of highly correlated features, we selected one from each pair. After which there were a total 32 numerical attributes.

3. Data Preprocessing

1. Numerical attributes: total count=32

We standardized the numerical attributes which had the range greater than 1.

2. Categorical attributes: total count=3 (service, flag, protocol type)

For the columns service and flag had a high number of subcategories. On converting numerical value using one hot encoding result would have resulted in the addition of a column per subcategory. In this case it would result in adding 67 + 11 + 3 - 3 = 78 columns. This would have added to the complexity of the model. Hence, we used baseN encoding which highly reduces the dimensionality as the value of N increases.

3. We used SMOTE (Synthetic Minority Oversampling Technique) for balancing the classes.

4. Model Selection

We used the following algorithms for training the model and hyper tuned them.

1. Decision tree
2. Naïve Bayes
3. Random forest
4. Logistic regression

5. Model Comparison

Hypertuned decision tree performed the best. (This needs to be completed will finalize this section after the python notebook)

Work Planning and organization of each team member

We devised this project as an opportunity not only to apply the techniques we have learned in the class, but also to broaden our knowledge on each component. Every member of the group contributed individually to this project, and whatever method was most effective according to them was used either in Data preprocessing, Data Cleaning or in Feature Selection.

Everyone not only mastered the techniques they worked on, but taught them to others as well.

We collectively chose the best option to improve our model once everyone had finished their parts.

Improving teamwork and collaboration

From the initial steps of the project, we as a team came up with our own inputs and had them discussed with the teammates in the weekly meetings. The most optimal methodology among the proposed ideas was considered and gave us exposure to how a specific task could be tackled in different ways. This collaborative approach has helped us to learn collectively and help us have end-to-end knowledge of the project.

Individual Contribution

As part of Data Manipulation and Analysis, I have conducted the basic data analysis like checking for null values and different datatypes in the dataset. The Output from Data

Manipulation and Analysis is that the dataset contains three columns with categorical values which are Protocol type, Flag and Service.

As a part of Feature Selection, I have used Correlation and Mutual Information classification. In our dataset we have a large number of features because of which the complexity increased. And the probability of overfitting is very high.

By Using Correlation, we can understand the relationship between multiple variables and attributes.So I plotted the correlation heatmap, and got the correlation percentages between the features and then dropped the features which are highly correlated. By dropping the features, the problem of overfitting was avoided.

Next, I used Mutual Information classification. By using the mutual information classification I got the most relevant features which contributed more to the output. But we did not take this into consideration as we decided to go with all the features available in the dataset after dropping a few correlated features.

As a part of Data Preprocessing Categorical variables are known to hide and mask lots of interesting information in a data set and also Most of the algorithms produce better results with numerical variables. So now I tried converting categorical values to numerical values and the first method I used was Label Encoding.

By working with Label Encoder I was able to achieve what I desired. However, the disadvantage was that the numerical values were misinterpreted by the algorithms as they have hierarchical order in them.

So the next method I used was One Hot Encoding. The drawback I had while using One Hot Encoding was it created a column per subcategory. This resulted in the addition of a total 77 columns. This added complexity to the model. Our aim while doing data preprocessing is to decrease the complexity as much as possible.

Atlast, we used BaseN encoding which reduced the dimensionality of the dataset.

After all this was done, I trained the model using Logistic Regression. The Accuracy I got was 77.24% with precision value 0.82, Recall of 0.77 and F1-Score of 0.75.