Anomaly Detection in Network

Abstract: In our project, we build an intrusion detection model based on **Bot-Iot** dataset. Since the dataset is one of the largest datasets available online, to reduce the dimensionalities and good performance of the model, we collected the best 19 featured dataset available online. After getting most relevant features we got **99**% accuracy.

1.Data:

The Dataset **Bot-Iot** used for this project was collected from kaggle[1]. It is one of the largest datasets with **2934817** samples and **19** features. For training, we used training dataset and testing dataset separately to see whether it can detect accurately very new data or not.

1.1 Data Analysis :

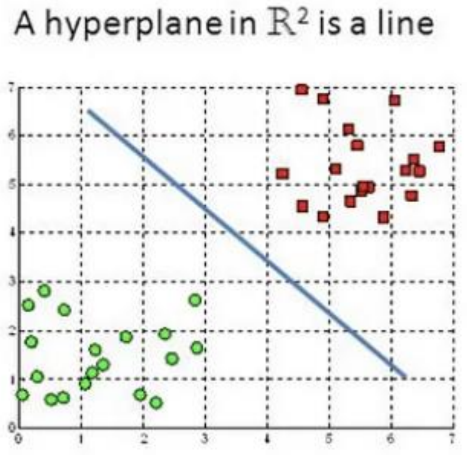
a)correlation with each column: we find the correlations of each column with another and based on the correlation, we dropped some features which are highly correlated(>=0.9)

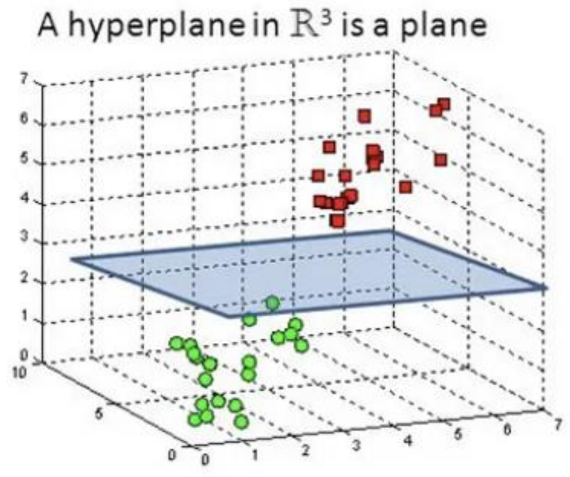
b)normalization: we normalize all the features range from 0 to 1. The goal of normalization is to transform features to be on a similar scale. This improves the performance and training stability of the model since we are applying several algorithms.

2. Models

2.1 Support Vector Machine

Support Vector Machine (SVM) is a relatively simple Supervised Machine Learning Algorithm used for classification and/or regression. It is more preferred for classification but is sometimes very useful for regression as well. Basically, SVM finds a hyper-plane that creates a boundary between the types of data. In 2-dimensional space, this hyper-plane is nothing but a line. In SVM, we plot each data item in the dataset in an N-dimensional space, where N is the number of features/attributes in the data. Next, find the optimal hyperplane to separate the data. So by this, you must have understood that inherently, SVM can only perform binary classification (i.e., choose between two classes). We make the target level as a binary classification 1. Anomaly(all the attacks) 2.Normal .





2.1 Gaussin Naïve Bayes

Naive Bayes makes the assumption that the features are independent. This means that we are still assuming class-specific covariance matrices (as in QDA), but the covariance matrices are diagonal matrices. This is due to the assumption that the features are independent.

So, given a training dataset of N input variables x with corresponding target variables t, (Gaussian) Naive Bayes assumes that the class-conditional densities are normally distributed

$$P(\mathbf{x} \mid t=c, \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c) = N\left(\mathbf{x} \mid \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c\right),$$

where μ is the class-specific mean vector, and Σ is the class-specific covariance matrix. Using Bayes' theorem, we can now calculate the class posterior

$$\overbrace{P(t=c|\mathbf{x}, \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)}^{\text{class posterior}} = \frac{\overbrace{P(\mathbf{x} \mid t=c, \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)}^{\text{class-conditional density}} \overbrace{P(t=c)}^{\text{class prior}}}{\sum_{k=1}^{K} P(\mathbf{x} \mid t=k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_c)\, P(t=k)}.$$

We will then classify x into class

$$\hat{h}(\mathbf{x}) = \underset{c}{\operatorname{argmax}}\, P(t=c|\mathbf{x}, \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c).$$

3. Experiments and Results

3.1 Suppor Vector Machine

Radial basis function kernel					
Accuracy			**99%**		
	Precision %	Recall %	F1 %	F/P %	F/N %
0	87	96	91	3.7	0.0
1	100	100	95		

3.2 Naïve Bayes

Gaussian					
Accuracy			**99%**		
	Precision %	Recall %	F1 %	F/P %	F/N %
0	71	99	82	0.9	0.0
1	100	100	100		

References:

[1]. Dataset: Bot_Iot (Link)

[2]. Nickolaos Koroniotisa, Nour Moustafaa,, Elena Sitnikovaa, Benjamin,Turnbull, 'Towards the Development of Realistic Botnet Dataset in the Internet of Things for Network Forensic Analytics: Bot-IoT Dataset, 2018.[Link]