**Detection**

**Input:-The dataset- modified KDD format**(for training)**,the forecasted dataset(work dataset),text file containing the ip addresses**

**Output:-A text file containing the types of connections**

**Detection Process:**

**a)Training the machine learning code:-**

The dataset we obtained was directly taken from the official KDD cup website which was used after some modification.

The efficiency of the various machine learning algorithms used is determined by comparing their accuracy values. Higher the accuracy value the more efficiently an algorithm performs.

For calculating the accuracy ,the KDD dataset was divided into **training** and **testing** portions in a **80:20** ratio.

**i)Calculating the accuracy**

Accuracy is one metric for evaluating classification models. Informally, accuracy is the fraction of predictions our model got right. Formally, accuracy has the following definition:

**Accuracy = Number of correct predictions/Total number of predictions**

For binary classification, accuracy can also be calculated in terms of positives and negatives as follows:

**Accuracy = (TP + TN)/(TP + TN + FP + FN)**

**Where TP = True Positives, TN = True Negatives, FP = False Positives, and FN = False Negatives.**

By dividing the dataset into two parts:-

-- 80% for training the code

-- 20% was used for calculating accuracy

Only taking this accuracy parameter into consideration, it was identified that the **Decision Tree Classifier** algorithm had the highest efficiency.

The machine learning code output:-

i) A table with **5** columns :-

**Column 1 :** The different types of connection-related threats

**Column 2 (Precision): TP / (TP + FP)**

**Column 3( Recall ): TP/(TP+FN)**

**Column 4( f1-score) : 2\*TP / (2\*TP + FP + FN)**

**Column 5( Support ) :** The number of instances of a particular connection-related threat in the training part of the dataset( in our case 80% of the dataset)

**b) Additional code to increase the efficiency of the machine learning algorithm**

**i) Rescaling :-** Attributes are often rescaled into the range between 0 and 1.**Column-wise rescaling.**

**ii)Standardize Data :-** Standardization is a useful technique to transform attributes with a Gaussian distribution and differing means and standard deviations to a standard Gaussian distribution with a mean of 0 and a standard deviation of 1.

**iii)Normalization:-**Normalizing in scikit-learn refers to rescaling each observation (row) to have a length of 1 (called a unit norm in linear algebra).**Row-wise rescaling.**

**iv)Binarize Data (Make Binary):-**You can transform your data using a binary threshold. All values above the threshold are marked 1 and all equal to or below are marked as 0.

All the accuracy values with and without rescaling was recorded into a matrix.

After comparing the accuracy values the **Decision Tree Classifier algorithm with no rescaling** was found to be the most efficient one.

-----------------------**End of training and selecting the most efficient algorithm--------------------------------**

Step 3: The **work** dataset is passed to the machine learning code to determine the type of connection.

Step 4: For every connection listed in the dataset the machine learning algorithm determines the type of attack. The output is **normal** if it is a secure connection , otherwise prints the type of the **malicious** connection.

The types of connection are stored in a separate file and is passed onto the **blocking** module.