# *Pattern Recognition Project Report*

## Project Title: -

Internet firewall Action Classification and Prediction

## By

D Mabu Jaheer Abbas – S20190020209

Yaswanth Kande – S20190020220

Pattem Gaurav Naga Maheshwar – S20190020237

# Introduction: -

This is the project in which we have trained a model to classify and predict the Internet firewall action, and this classification and prediction is based on the data-set collected from the **'UCI machine learning Repository'**. Here we are going to split the data into appropriate ratio i.e., 3:1, and first part is used to train the model and another part is to test the model. According to the data we can say that it's multiclass classification, there are four actions given in the data-set, which can be understood as four classes.

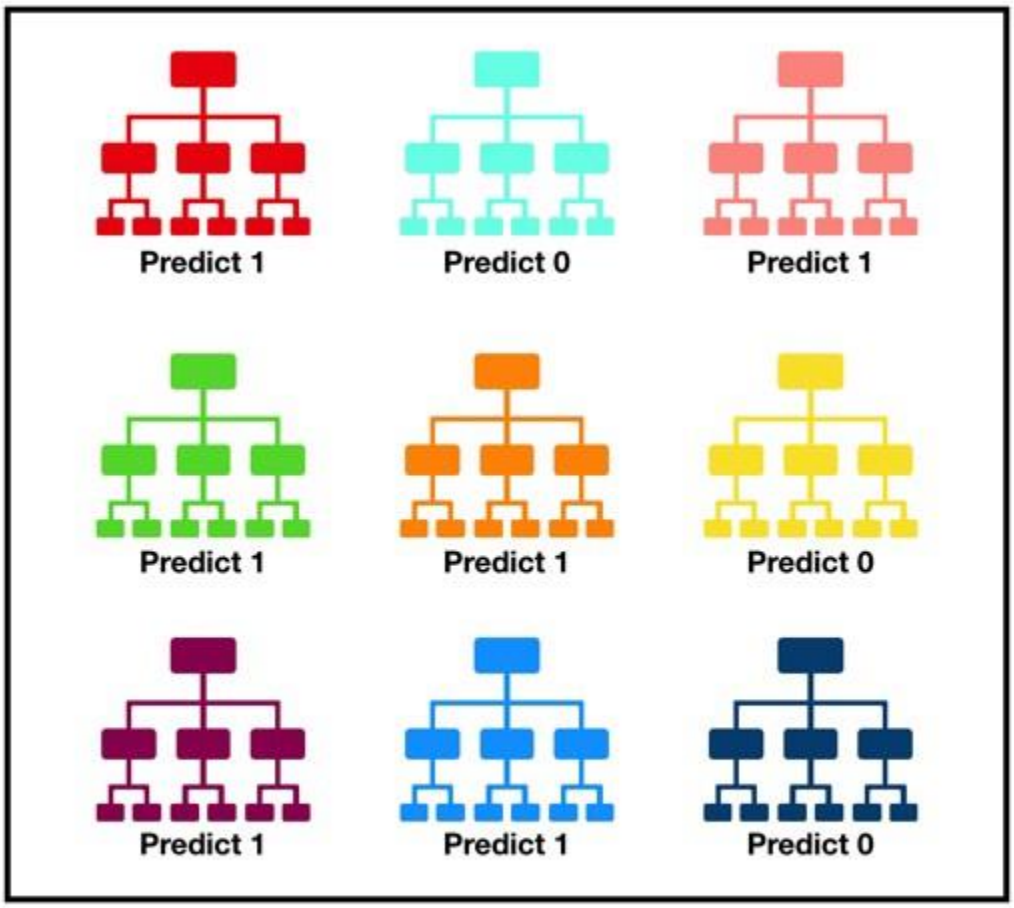
Details of Data-set: -

- No. of Attributes = 12
- No. of Instances >= 1000
- Attributes are –
    - Source Port
    - Destination Port
    - NAT Source Port
    - NAT Destination Port
    - Action – Target Attribute
    - Bytes
    - Bytes Sent
    - Bytes Received
    - Packets
    - Elapsed Time (sec)
    - pkts_sent
    - pkts_received

Therefore, the data-set satisfies all the fulfils all the requirements.

# Random Forest Classifier: -

Random forest, like its name implies, consists of a large number of individual decision trees that operate as an ensemble. Each individual tree in the random forest spits out a class prediction and the class with the most votes becomes our model's prediction.



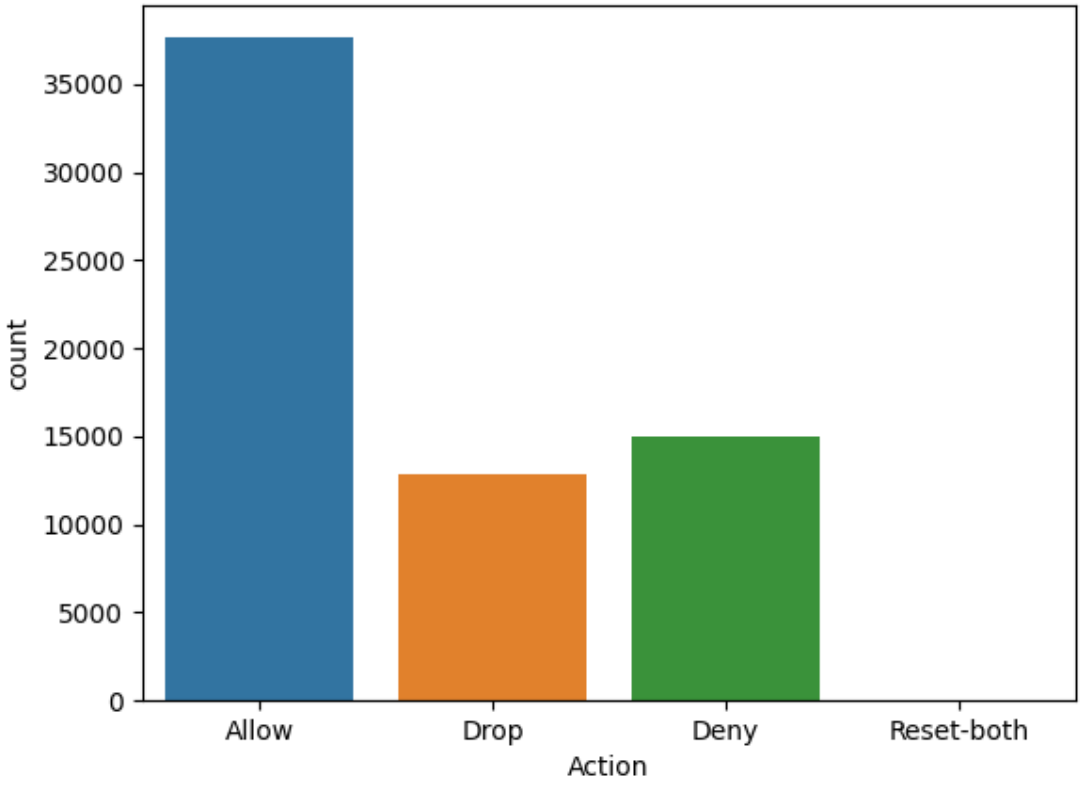

The fundamental concept behind random forest is a simple but powerful one — the wisdom of crowds. In data science speak, the reason that the random forest model works so well is:

***'A large number of relatively uncorrelated models (trees) operating as a committee will outperform any of the individual constituent models.'***

In this project we build a 'Random Forest Classifier' model to achieve good performance.

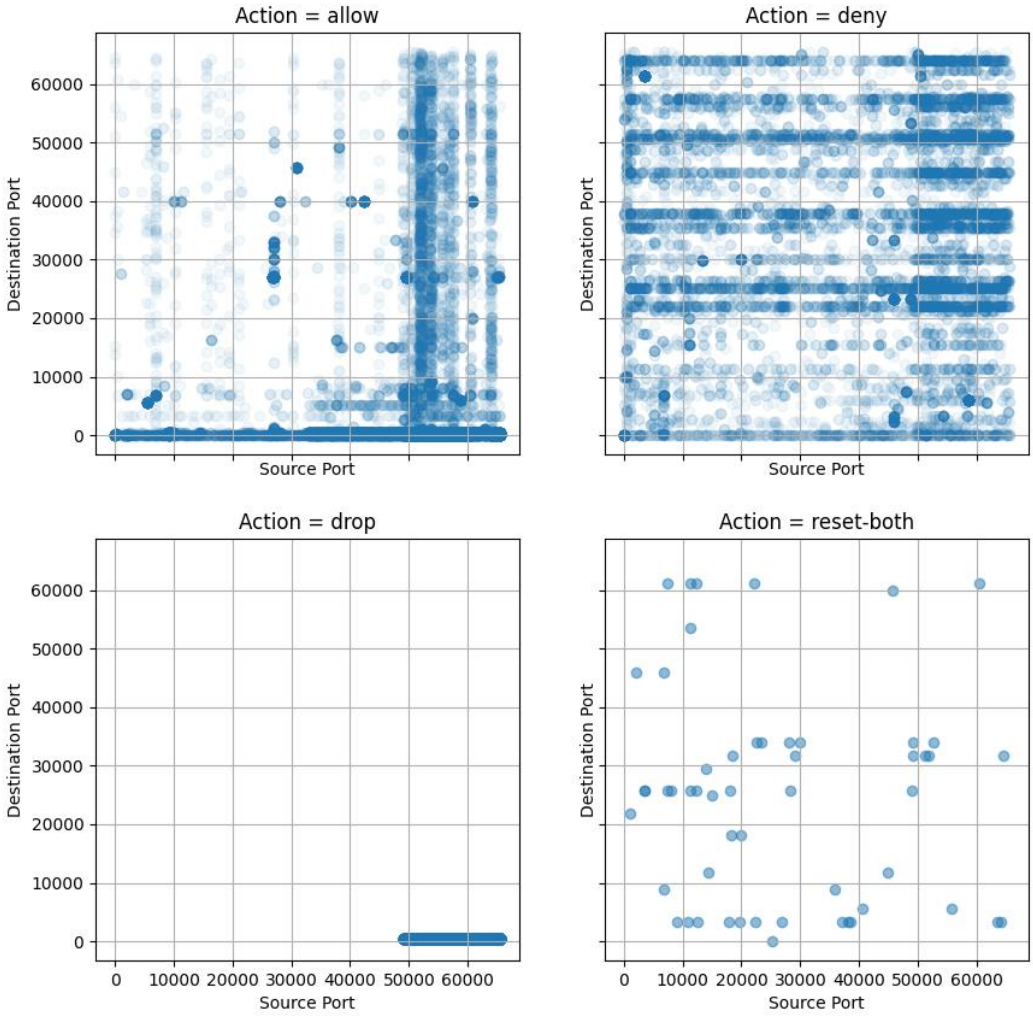
## Implementation: -

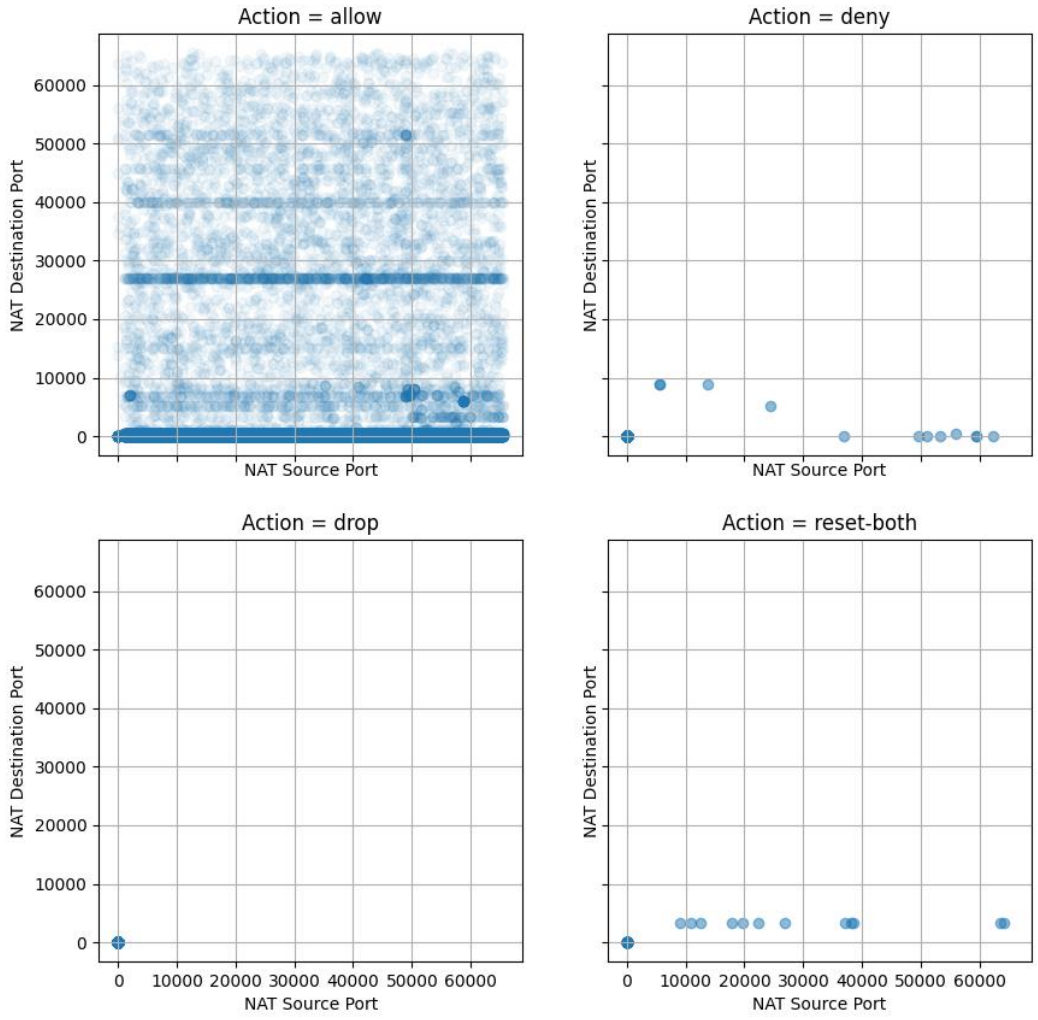
- First, we tried to read .csv data-set file, and understood the basic information like shape, info (to find any missing vale is present or not).
- As the data do not have any type of missing values, we need not to fill any of the places in data-set.
- As we know that Action is the target attribute, we found no. of unique values in that Action attribute and plotted the count of each unique Action.



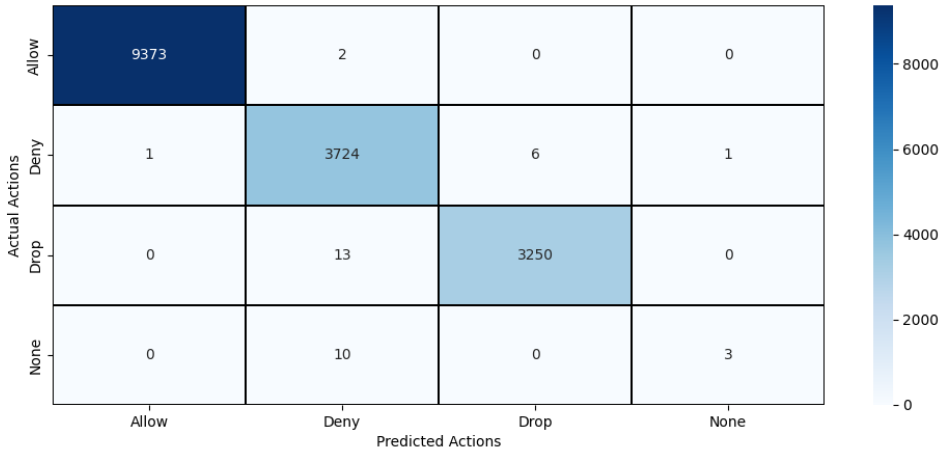

- After this we divided the data-set into two part, they are 'num_features', 'cat_features' and understood the spread of values for each attribute.
    - **Cat_features –** ['Source Port', 'Destination Port', 'NAT Source Port', 'NAT Destination Port']
    - **Num_features –** All remaining attributes.
- And also, we have done plotting between the Source Port and Destination Port for each Action.

- Similarly, we also done the plotting between NAT Source Port and NAT Destination Port.






- Now, we again divided whole data-set into four part,
  - X_train - data used to train the model.
  - Y_train - Action values used for training model.
  - X_test - data used for testing of model.
  - Y_test - Action values for comparing with y_pred.
- As this is multi-class problem we have used Random Forest Classifier to classify and predict the values (to get high accuracy).
- After training and testing we have plotted the confusion matrix i.e., between Actual Actions and Predicted Actions.




- From this confusion matrix we have calculated the accuracy percentage,
    - Accuracy percentage we have acquired is – 99.58%

# Updates Done are

## Ranking Method to select features: -

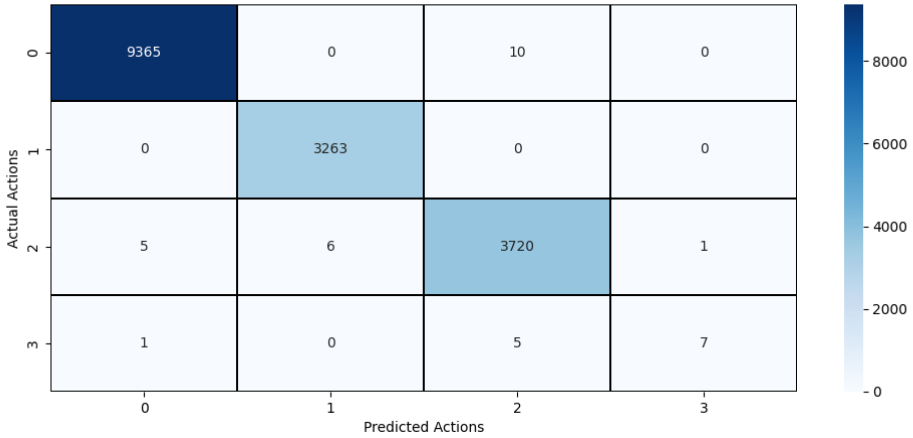
- In this process we can get ranks for each attribute present in the dataset.
- We have used Random forest classifier to select the features based on ranks.
- Among all those ranks we have used top 5 features to train the model (Random forest classifier model).
- After using Ranking to select feature the accuracy of model is almost same, but varies in decimal values.
- Here are the ranks for each features and top 5 features among then are,

```
Printing all the ranks in sorted order ::
[(100, 'Action'), (89, 'Destination Port'), (87, 'Bytes Sent'), (87, 'Bytes'), (79, 'NAT Source Port'), (79, 'NAT Destination Port'), (79, 'Elapsed Time (se
c)'), (77, 'Packets'), (74, 'pkts_received'), (74, 'Bytes Received'), (61, 'Source Port'), (57, 'pkts_sent')]

The 5 best features selected by this method are :
Destination Port
Bytes Sent
Bytes
NAT Source Port
NAT Destination Port
```

- We have neglected Action attribute even it's top most one, it's the target feature.

- Updated Confusion Matrix is,




- Here the accuracies in both cases,

```
|-----------------------------------------------------------|
 Accuracy Percentage for test data by Random Forest model is :: 99.58493560397973%
|-----------------------------------------------------------|


|-----------------------------------------------------------|
 Accuracy Percentage for test data by Random Forest model after Using Ranking for feature selection :: 99.82909113105048%
|-----------------------------------------------------------|
```

- In the Process to calculate Ranks we have to convert Actions into numbers.
- Converted Actions into numbers are,
  - Allow – 1
  - Drop – 2
  - Deny – 3
  - Reset-both – 4

Link to GitHub Repository

Link to Data-Set

# References: -


- https://towardsdatascience.com/solving-a-simple-classification-problem-with-python-fruits-lovers-edition-d20ab6b071d2
- https://www.kaggle.com/docxian/internet-firewall-analysis/data

- https://stackabuse.com/random-forest-algorithm-with-python-and-scikit-learn/

## Contributions: -

- **Pattem Gaurav Naga Maheshwar** - Getting Dataset and some references to identify correct model for classification and basic analysis on data.
- **Yaswanth Kande** - Analysing the data by pre-processing and Cleaning of dataset if needed to make data ready for classification.
- **D Mabu Jaheer Abbas** - Analysing the features and training the model to get good performance on test data.