RGUKT IIIT

Cyber Security Threat Detection Using ML

Welcome to the Project Presentation of “**Cyber security Threat Detection Using Machine Learning**”, Here we are going to explore how this ML model can classify the network packets as Malicious or Benign.

---Let’s Go---

- **E.Vignesh (R201065)**

RGUKT IIIT

Search

Project Overview

This project focuses on the development and evaluation of a high-performance Machine Learning (ML) model for the **offline analysis and classification of already captured network flow data**, specifically targeting the identification of Botnet attacks.

Which pose a significant threat to modern cyber infrastructure. The objective was to design a robust model capable of accurately classifying historical network traffic records **into Benign (normal) and Bot (malicious) categories.**

RGUKT IIIT

How this Project Flow Works

Classification Flow...

1 : Collecting the Dataset, in our case it's "CIC_workflow.csv" file.

2 : Preprocessing the dataset

3 : Selecting the best ML Model.

4 : Training the model with the cleaned 80% the data.

5 : Testing the ML model with rest 20% data.

6 : Preparing for an unseen dataset for Performence analysis.

7 : Making Evaluation with .pkl file on this data and seeing performence of the model.

RGUKT IIIT

Collection of Dataset

For our ml model to be trained accurately with confident we need to use best collection of dataset, so, we have imported **CIC WorkFlow** dataset from Kaggle, which contains around 1.1L rows of data with 80 features of the packet, with which, our model is trained better to give better results.

We Used **pandas lib** from python to use in our development process, which can effeciently loads big csv files into the project.

RGUKT IIIT

Preprocessing The Data

The raw data is not useful for the development of ML model as it contains empty cells, INF values etc... it leads to incorrect predictions.

So, the data has to be preprocessed, we have use numpy module of python, which works incredebly with the datasets to manipulate the data effectively,

We have removed the rows containing INF values, NAN values, and redundent rows, which leads to cleaner dataset for the ml model to be trained.

RGUKT IIIT

Selecting the **ML Model**

As we have many classification models to be used for ml training, but we have to take the optimal one which gives best results.

So we have taken “**Random Forest Classifier**” which is trained on the preprocessed data.

A random forest classifier is an ensemble learning method that combines multiple decision trees to make a classification. It improves accuracy and prevents overfitting by building numerous trees with random subsets of data and features, then using a majority vote from the individual trees to determine the final prediction.

RGUKT IIIT

Model **Training**

Now the core part comes, Model training,
The model is trained well with the cleaned dataset which, internally it forms many decision trees each used subset of data to be trained.

For training we have used 80% of the dataset which is sufficiently enough to train the model, the rest 20% is allocated for the testing the predictions of the model.

For training we have used Random Forest Classifier, because of it uses many decision trees which improves accuracy and prevents overfitting.

RGUKT IIIT

Model **Testing**

With the rest 20% of the data, we have tested the trained pkl model, fortunatly we have best results got, which are use full for further model evaluation, we have used matplotlib library of python used for visual representations of the results.

We have generated Confusion_Matrix,

Borcelle

Search

Prepraing
For unseen data

For further analysis, we have taken a subset of data from CIC workflow dataset, this subset contain randomly selected 1000 rows, in which the last target column is deleted so that we can measure the correctness of the ML model.

We have given this subset to the .pkl file, which predicts the target values of those 1000 packets as Malicious or Benign.

RGUKT IIIT

Model **Performence Evaluation**

After the model has predicted on the subset, we have extracted the evaluation metrics of the model performence, Fortunately we got 100% accuracy on the model performance, with good confidence.

RGUKT IIIT

Results

1. Loading data from unseen_predictions.csv and extracting the last two columns...
2. Encoding labels...
   Labels encoded: {'Benign': np.int64(0), 'Bot': np.int64(1)}

3. Analyzing Model Performance and Metrics...

--- Key Summary Metrics ---
Model Accuracy (Overall): 100.0000%
Balanced Accuracy: 100.0000%
Matthews Correlation Coefficient **(MCC): 1.0000**

--- Classification Report (Detailed Metrics) ---
precision recall f1-score support

Benign 1.00 1.00 1.00 500
Bot 1.00 1.00 1.00 500

accuracy 1.00 1000
macro avg 1.00 1.00 1.00 1000
weighted avg 1.00 1.00 1.00 1000

--- Confusion Matrix (Raw Numbers) ---
[[500 0]
[0 500]]
Confusion matrix plot saved as final_performance_confusion_matrix.png

4. Error Analysis (Misclassifications) ---
Zero misclassifications found! (Perfect model or very small test set)

RGUKT IIIT

Thank **You**

Vignesh Enaganudla (R201065)