Cyber Security

(Semester Project)

(Session 2020-2024)

Department of Computer Science

Namal University Mianwali

Muhammad Ali Raza BSCS-2020-15

Instructor: Dr. Arshad Farhad

Table of Content

Title:

Anomaly Detection in Network Traffic Using Gaussian Distribution

Abstract:

The Gaussian distribution-based anomaly detection method for network traffic data is used to get complex data anomalies. The aim is to find odd patterns and departures from network communication's typical behaviour. The steps that comprise the approach are identifying pertinent attributes, normalising the data, estimating Gaussian distribution parameters, computing anomaly scores, and establishing an anomalous detection threshold. A scatter plot is used to display the data, highlighting the differences between aberrant and normal network activity.

Introduction:

In the realm of network security, identifying patterns in network traffic is crucial for identifying potential threats and vulnerabilities. Often, complex patterns of network activity are too complex to handle using conventional methods. The articles that I read in my proposal article look at a different approach that uses the Gaussian distribution to identify strange patterns in network traffic data.

Objectives:

This project's primary objective is to develop a strong anomaly detection system that can identify deviations from normal network behaviour. The goal is to use the Gaussian distribution, which assumes that the data is evenly distributed, to create a baseline for usual network activity and find cases that significantly deviate from this baseline & find the anomaly of any dataset from Wireshark related to traffic.

Goals:

Anomaly Detection:

Establishing a threshold and using the probability density function (PDF) of the Gaussian distribution to calculate anomaly scores.

Visualisation:

Showing the anomalies in graphic form to understand the position and checking in the dataset where anomalies exist.

Advance Threat Detection:

Early knowing the anomalies in network traffic leads to detecting the threat. It minimizes the impact of unusual activity in a network.

Improve Network Health:

Knowing the anomalies or vulnerabilities in traffic allows the admin or user to recover it before it impacts the network or system which can result in slowing down the system or in the last case getting access to the system.

Dataset Tool:

I take the dataset from the widely used network protocol analyzer Wireshark for this investigation. Wireshark is a vital tool for tracking and showing real-time network data flow, which is necessary for interpreting and evaluating network activities.

Implementation:

Firstly, I install and import some necessary libraries for data analysis, numeric operations, visualisation, and Gaussian-based anomaly detection. After that, I read the network traffic dataset which I fetched from Wireshark and extracted the features (Length &Time), which are important for communication in data related to network traffic flow and standardised these features for the next step. To calculate the anomaly score, I calculated the mean and standard deviation for each feature. Then, find the anomaly score by using the probability density function(PDF) for Gusaasion-Distribution. Last, after setting the threshold and identifying the anomalies from the dataset. Finally, by using a scatter plot to visualize the anomalies with red colour and normal data with simple blue colour. The reason for choosing the scatter plot is to easily understand the behaviour of normal and anomaly data as shown in the figure below:

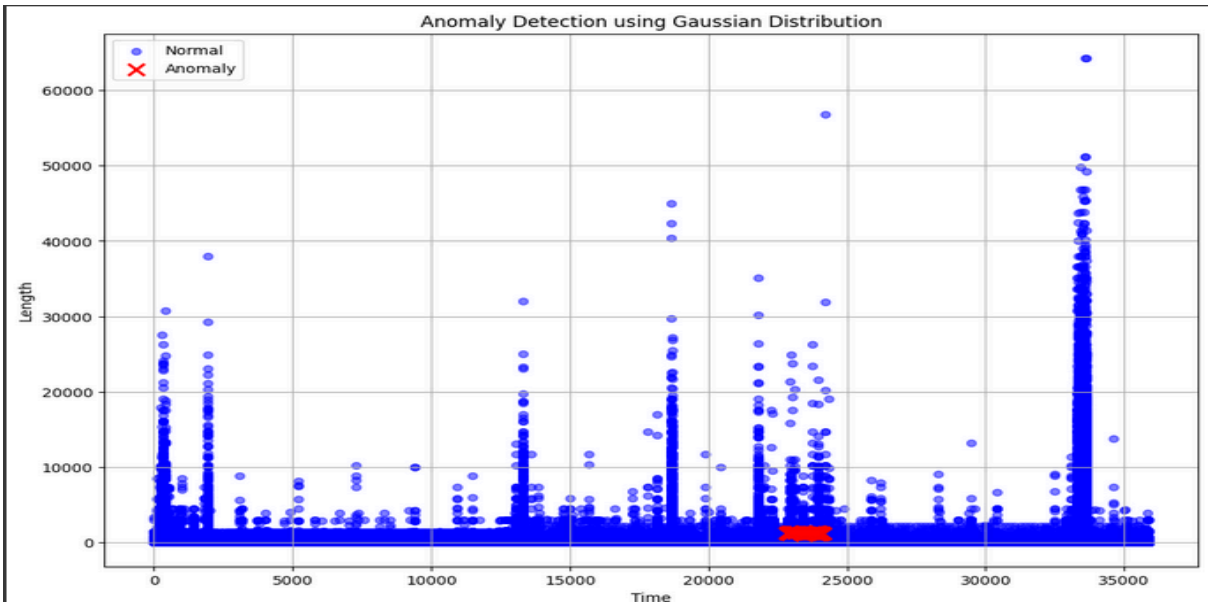

Fig1: Scatter plot to show an anomaly

Proceeding with it, a confusion matrix is then generated by comparing the ground truth labels(where 1 represents anomalies & 0 represents normal data) with the predictions from your anomaly detection approach. It provides the summary of model performance and gives the no of true positives, true negatives, false positives and false negatives. Then, I find the Precision and recall by using the truth positive, true negative, false positive & false negative. & F1 score which is the harmonic mean of precision and recall. These measures shed light on how well the model detects abnormalities while reducing false positives and false negatives. Eventually, I set the thresholds of 77 and 99 to check the difference between the ROC curve as follows:

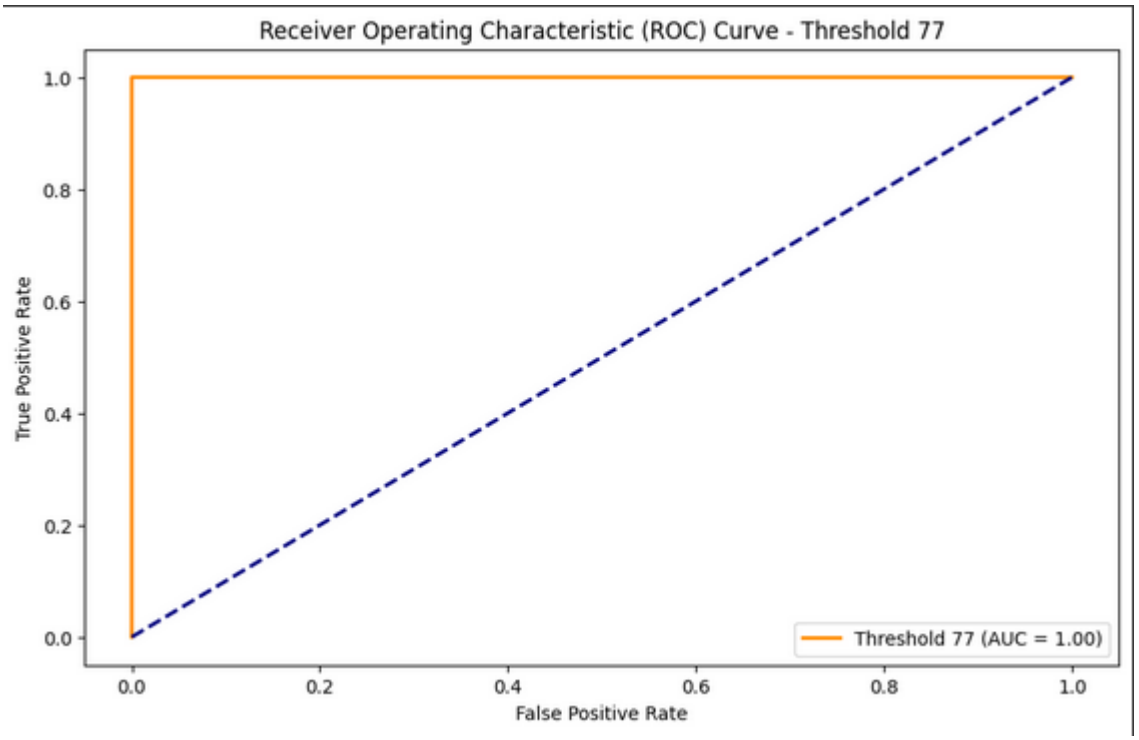

Fig2: ROC Curve- Threshold 77

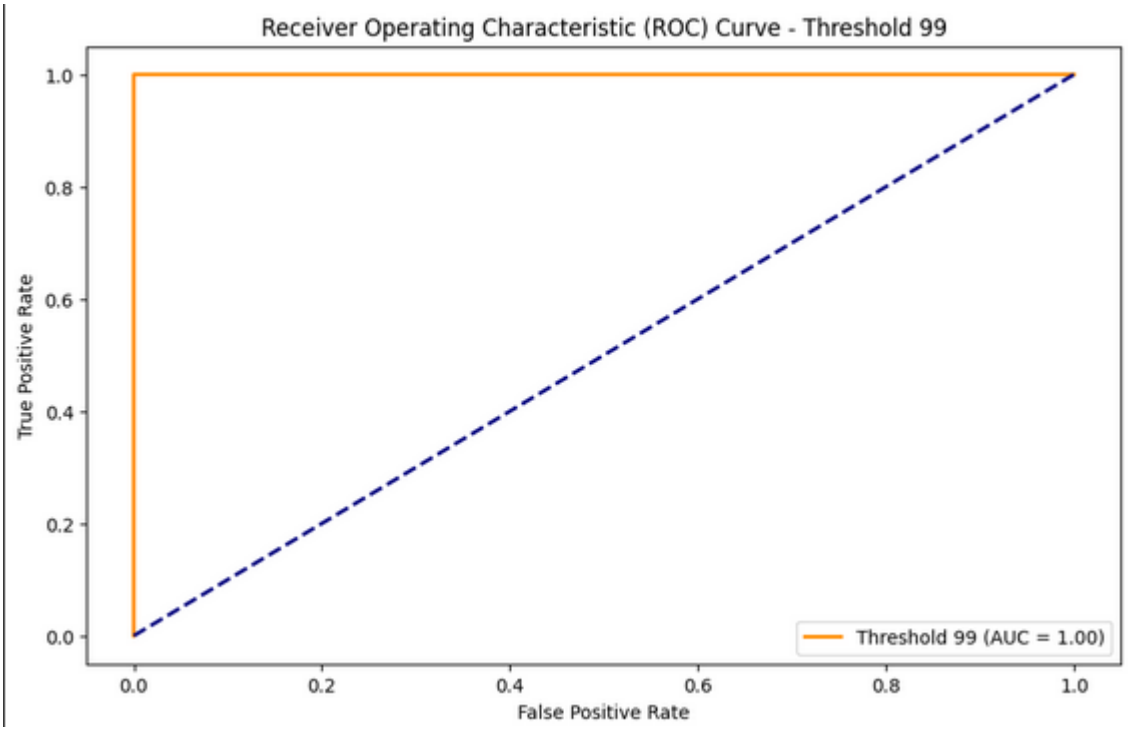

Fig3: ROC Curve- Threshold 99

Now, will look at the confusion matrix and matrics including precision, recall, F1-score and AUC value to check the overall performance of the model as shown below:

```
# @title Look at Confusion Matrix
# Display confusion matrix and metrics
print("Confusion Matrix:")
print(cm)
print("\nPrecision: {:.2f}".format(precision))
print("Recall: {:.2f}".format(recall))
print("F1 Score: {:.2f}".format(f1))
print("AUC: {:.2f}".format(roc_auc))
```

```
Confusion Matrix:
[[386319      0]
 [     0 115394]]

Precision: 1.00
Recall: 1.00
F1 Score: 1.00
AUC: 1.00
```

Fig4: Confusion Matrix & Matrics

Key Findings:

- From normal traffic data, we use the Gaussian distribution to find the deviation in the form of anomalies.

- Anomalies are behaving as vulnerabilities which can be the weakness of the system where threats can occur in future. So, we can use these anomalies for further investigation to set the precautions to avoid any threat or any unusual activity on the system.

Conclusion:

This work demonstrates how well Gaussian distribution-based anomaly detection can identify deviations from normal network behaviour. With the aid of this practical tool, network administrators and security analysts may increase the safety and security of their networks. Seeing irregularities makes it easier to analyse the data and proactively reduce risks.

Note:

Please find the attached **Cyber_project.ipynb** file to see all implementations and also the dataset file named **network_traffic_dataset.csv.**