SentinelNet – AI-Based Network Intrusion Detection System

Introduction

In this project, I am developing a Network Intrusion Detection system using machine learning. The main goal is to identify whether network traffic is normal or malicious based on different features in the dataset. This system will help in detecting cyber-attacks automatically.

Dataset used

For this project, I used the NSL-KDD dataset, which is a standard dataset used for intrusion detection research. It contains labeled network traffic records with various features describing each network connection.

The files used:

- KDDTrain+.txt (training dataset)
- KDDTest+.txt (testing dataset)

Tools and Libraries Used

- Python
- Jupyter Notebook
- Pandas
- NumPy
- Matplotlib

Work Completed

1. Dataset Setup

First, I created a folder to store the dataset files. Then I downloaded both training and testing datasets from an online source using Python.

2. Loading the Dataset

After downloading, I defined the correct column names for the dataset and loaded both files into pandas DataFrames.

I checked:

- First few rows
- Last few rows
- Shape of dataset
- Data types of columns
- Statistical summary using describe()

This helped me understand the structure of the dataset.

3. Checking Data Quality

I checked:

- Whether there are any missing values
- Whether there are duplicate rows

There were no major missing values.

4.Protocol Analysis

The dataset contains three protocol types:

- TCP
- UDP
- ICMP

I analyzed how many records belong to each protocol and created a bar chart for visualization.

Observation:
Most of the traffic is TCP-based. UDP and ICMP have fewer records compared to TCP.

5.Statistical Analysis

I grouped the data based on:

- Protocol type and calculated average duration.
- Class label and calculated average duration.

From this, I observed that:

- Some attack types have higher average connection duration.
- Duration varies depending on protocol and attack type.

Data Visualization

I created several visualizations to understand the data better:

- Bar chart for protocol distribution
- Histogram for source bytes
- Boxplot to check outliers
- Heatmap for protocol vs flag
- Correlation matrix for numerical features
- Boxplots comparing duration across classes