DDoS Attack Detection Using Machine Learning

1. Introduction

This project aims to develop an interactive web-based system for detecting Distributed Denial-of-Service (DDoS) attacks using machine learning. The system is built using **Flask (Python) as the backend**, **HTML/CSS (Bootstrap) as the frontend**, and a **Naïve Bayes classifier** to predict potential DDoS attacks based on uploaded network traffic data.

2. Technologies Used

- **Programming Language:** Python (Flask Framework)
- **Frontend:** HTML, CSS, Bootstrap
- **Machine Learning:** Naïve Bayes Classifier
- **Dataset:** APA-DDoS-Dataset 2.csv
- **Libraries:**
    - pandas for data handling
    - numpy for numerical operations
    - sklearn for ML model training
    - flask for web framework
    - pickle for model persistence

3. System Architecture

1. **Frontend:** Allows users to upload network traffic data (CSV format).
2. **Backend (Flask):** Processes the file and sends it to the ML model.
3. **Machine Learning Model:** Predicts if the uploaded data contains a DDoS attack.
4. **Result Page:** Displays classification results.

4. Installation and Setup

4.1 Prerequisites

Ensure you have the following installed:

- Python 3.x
- Flask (pip install flask)
- Pandas (pip install pandas)

- Scikit-learn (pip install scikit-learn)
- Bootstrap (Included via CDN in HTML)

4.2 Running the Project

1. Clone the repository or place the project files in a directory.
2. Navigate to the project directory and run:
3. python ddos.py
4. Open your browser and go to http://127.0.0.1:5000/.
5. Upload a CSV file and click **Detect Attack**.
6. The system will display the result.

5. Implementation Details

5.1 Machine Learning Model

- **Algorithm Used:** Gaussian Naïve Bayes Classifier
- **Training Data:** Preprocessed APA-DDoS-Dataset
- **Feature Engineering:** Label encoding applied to categorical data
- **Accuracy Achieved:** ~95% on test dataset

5.2 Flask Backend

- Loads the trained model (ddos_model.pkl)
- Accepts user-uploaded CSV files
- Preprocesses the data and makes predictions
- Renders the appropriate result page (index1.html, index2.html, or index3.html)

5.3 Frontend

- HTML form for file upload
- Bootstrap-based UI for better responsiveness
- AJAX used for smoother interactions (optional enhancement)

6. Conclusion

This project successfully integrates a **machine learning model** into a **web-based system** to detect DDoS attacks from uploaded network traffic data. Future improvements can include **deep learning models, real-time monitoring, and API integration** for live traffic analysis.

7. Future Enhancements

- Implementing **real-time traffic analysis**
- Adding **visualizations** for detected threats
- Deploying the system on a **cloud server**

8. References

- Flask Documentation
- Scikit-Learn Documentation
- Bootstrap Framework