Project: Intrusion Detection System Using Machine Learning

Project Overview:

In this project, you will work on a dataset commonly used for network intrusion detection. The dataset contains network traffic data, including both normal and attack behaviors. Your objective is to build and evaluate a machine learning model that can classify different types of network activities **as several types (multi-class classification)**. This project will test your ability to choose appropriate algorithms, preprocess data, and evaluate model performance.

Dataset Description:

The given dataset is composed of network traffic data captured over a period of time. Each instance in the dataset represents a connection, and it is characterized by 41 features that describe various properties of the connection, such as:

1. **Basic Features:**
   - Duration of the connection.
   - Type of protocol used (e.g., TCP, UDP).
   - Number of bytes sent and received.
   - Status flags of the connection.
2. **Content Features:**
   - Number of failed login attempts.
   - Presence of certain commands in the data.
   - Number of access attempts to sensitive files.
3. **Traffic Features:**
   - Number of connections to the same host in a specific time window.
   - Number of connections to different hosts.

The dataset is already labeled. The attack types can be grouped into four main categories:

1. **DoS (Denial-of-Service):** Attacks that flood a target system with traffic to exhaust resources.
2. **Probe:** Attacks that attempt to gather information about the network.
3. **R2L (Remote-to-Local):** Attacks that exploit vulnerabilities to gain unauthorized access from a remote location.
4. **U2R (User-to-Root):** Attacks that attempt to gain superuser access on a local system.
5. **…**

Project Requirements:

1. **Data Preprocessing:**
   - Analyze the dataset to understand the distribution of classes.
   - Handle missing values, if any.
   - Encode categorical features appropriately.
2. **Feature Selection:**
   - Perform feature selection or dimensionality reduction to improve model performance.
3. **Model Selection:**
   - Experiment with at least three different machine learning algorithms (e.g., Decision Trees, Support Vector Machines, Neural Networks).
   - Justify your choice of algorithms based on theoretical knowledge and the characteristics of the dataset.
4. **Model Evaluation:**
   - Evaluate the performance of your models using metrics such as accuracy, precision, recall, F1-score, and confusion matrix.
   - Compare the performance of different models and provide a comprehensive analysis.
5. **Hyperparameter Tuning:**
   - Use techniques like Grid Search or Random Search to optimize the hyperparameters of your chosen models.
6. **Report:**
   - Document your approach, decisions, and results in a detailed report.
   - Include visualizations of your findings, such as feature importance, performance metrics, and decision boundaries.

Deliverables:

1. A Python notebook (or script) with your code and results.
2. A report (PDF) explaining your approach, decisions, results, and conclusions.

Evaluation Criteria:

1. **Data Understanding and Preprocessing (20%)**
2. **Algorithm Choice and Justification (30%)**
3. **Model Performance and Evaluation (20%)**
4. **Hyperparameter Tuning and Optimization (20%)**
5. **Quality of the Report and Visualizations (10%)**