Network Intrusion Detection System (NIDS)
Using NSL-KDD Dataset

Project Members:

- Ashutosh Nayak – D1(47)
- Sai Sujal Patra – D1(63)
- Mritynjay Parida – D1(65)
- Arijit Mohanty – D2(8)

PROJECT RECORD — Network Intrusion Detection System (NIDS) using NSL-KDD Dataset

1. Introduction

Network intrusion detection is essential for protecting systems from modern cyber threats. This project uses machine-learning techniques to classify network traffic from the NSL-KDD dataset into normal or attack categories. The work completed so far includes preprocessing, encoding, scaling, training several ML algorithms, and evaluating their performance. SMOTE will be applied in the next stage to handle class imbalance.

2. Objective

- Understand the NSL-KDD dataset.
- Preprocess dataset (encoding + scaling).
- Train and compare ML models.
- Plan to apply SMOTE (not applied yet).
- Analyze initial model performance.

3. About the NSL-KDD Dataset

NSL-KDD contains 41 input features and 1 target label. It includes both categorical and numerical features.

Attack categories:
• DoS
• Probe
• R2L
• U2R

The dataset is imbalanced, especially for R2L and U2R classes. SMOTE will be used later to balance the training data.

4. Work Completed So Far

4.1 Data Loading

Loaded training and test files into pandas DataFrames. Verified structure and counts of each label.

4.2 Categorical Encoding

Encoded ‘protocol_type’, ‘service’, and ‘flag’ using One-Hot Encoding. Feature count expanded (around 120+ dimensions).

4.3 Feature Scaling

Applied StandardScaler to all numerical/encoded features to ensure equal contribution across algorithms.

4.4 Handling Class Imbalance

SMOTE is planned but NOT applied yet. Minority classes remain imbalanced in the current evaluations.

4.5 Machine Learning Models Trained

The following models were trained without SMOTE:
- Logistic Regression
- Random Forest
- Support Vector Classifier (SVC)
- k-NN
- Decision Tree
- Naive Bayes

5. Initial Results (Without SMOTE)

Table 1: Accuracy Summary of Initial Models (Without SMOTE)

Model	Observed Accuracy
Logistic Regression	~80%
Random Forest	~77%
Support Vector Classifier (SVC)	~81%
k-NN	~79%
Decision Tree	~82%
Naive Bayes	~57%

Observations:

- SVC performed the best in initial testing.
- Random Forest performed consistently.
- Recall for R2L and U2R remains low because SMOTE has not been applied yet.

6. Next Steps

- Apply SMOTE to improve minority class detection.
- Perform hyperparameter tuning.

- Re-evaluate all models with precision, recall, F1-score, and confusion matrices.
- Analyze feature importance using Random Forest.

7. Conclusion

So far, the NSL-KDD dataset has been successfully preprocessed and multiple ML models have been trained. Initial observations show promising performance from SVC and Random Forest. Applying SMOTE and tuning hyperparameters will significantly improve results in the next stage.