**Project Report**

**Network Intrusion Detection System using KDD Dataset**

**Team:**

**Tharun Kumar Nagipogu , Rizmi S , Leela Rajiv kalyan , Jhanvi**

**Introduction**

The objective of this project is to build a machine learning-based system to classify network traffic into normal and attack categories using the KDD dataset. The dataset includes various features to analyze network traffic behavior and classify it into attack types such as Denial of Service (DoS), Probe, User-to-Root (U2R), and Remote-to-Local (R2L).

**What I have done in the project:**

1. **Dataset Loading and Preprocessing:**
   * Loaded the KDD dataset for training and testing purposes.
   * Assigned descriptive column names to improve readability and usability.
   * Mapped the target variable attack to binary categories (0 for normal and 1 for attack) and further categorized attack types into DoS, Probe, U2R, and Sybil classes.
2. **Attack Categorization:**
   * Created a mapping function to classify attacks into specific categories.
   * Added new columns attack\_flag and attack\_map to the dataset for binary and multi-class classification tasks.
3. **Data Visualization:**
   * Visualized the distribution of attack types across different protocols (TCP, UDP, ICMP) using pie charts.
   * Analyzed the distribution of flags and services for normal and attack traffic.
   * Explored the target variable distribution and feature correlation matrix to understand the dataset better.
4. **Feature Encoding and Scaling:**
   * Applied one-hot encoding to categorical features (protocol\_type, service, flag).
   * Standardized numerical features (duration, src\_bytes, dst\_bytes) to improve model performance.
5. **Model Training and Evaluation:**
   * Used Random Forest for binary classification (normal vs. attack) and evaluated its performance on the validation set.
   * Implemented and compared additional classifiers, including Logistic Regression and K-Nearest Neighbors, to identify the most effective model.
   * Visualized model performance using box plots for better understanding.
6. **Error Analysis:**
   * Conducted detailed error analysis by generating a confusion matrix for the binary classification task.
   * Identified false positives and false negatives to evaluate model reliability.
7. **Exploratory Data Analysis (EDA):**
   * Plotted the distribution of numerical features (duration, src\_bytes, dst\_bytes) to analyze their patterns.
   * Visualized the correlation matrix to identify relationships between features and the target variable.
8. **Comparative Analysis of Models:**
   * Compared the accuracy of different classifiers using cross-validation.
   * Documented the strengths and weaknesses of each model, focusing on Random Forest's superior accuracy.
9. **Conclusion and Insights:**
   * Summarized the findings and highlighted the effectiveness of Random Forest in binary classification tasks.
   * Emphasized the importance of feature encoding, scaling, and visualization in improving model performance.
   * Discussed potential improvements based on error analysis, such as tuning hyperparameters and exploring ensemble methods.

**Dataset and Preprocessing**

A screenshot of a computer

Description automatically generatedThe dataset is loaded, and column names are assigned to improve readabilityA screenshot of a computer

Description automatically generated

* Separate datasets are used for training and testing.

**Target Variable Transformation**

A screenshot of a computer

Description automatically generatedThe target variable is transformed to represent normal as 0 and attack as 1 for binary classification.

**Categorization of Attack Types**

A screenshot of a computer

Description automatically generated

* Attacks are categorized into four types: DoS, Probe, U2R, and Sybil.
* A mapping function is used to create a multi-class target variable.

**Data Visualization**

**Attack vs Protocol Type**

* A screenshot of a computer

  Description automatically generatedVisualizes the distribution of attack types across different protocol types using pie charts.

**Flag Distribution A screenshot of a computer

Description automatically generated**

* Analyzes the distribution of flags for normal and attack traffic.

**Feature Encoding and Scaling**

A screenshot of a computer program

Description automatically generated

* Categorical features are encoded using one-hot encoding.
* Numerical features are standardized to improve model performance.

**Model Training and Evaluation**

**Random Forest for Binary Classification**

* Random Forest is used to classify traffic as normal or attack.A screenshot of a computer

  Description automatically generated
* The model achieves an accuracy score on the validation set.

**Comparison of Classifiers**

A screenshot of a computer code

Description automatically generated

* Compares Random Forest, Logistic Regression, and K-Nearest Neighbors classifiers.
* Results are visualized using box plots.

**Error Analysis**

A screenshot of a computer program

Description automatically generated

* Error analysis is conducted by visualizing the confusion matrix.
* False positives and false negatives are identified for further improvement.

**Confusion Matrix** Now, Summarizing the performance of a classification algorithm.A screen shot of a computer code

Description automatically generated

A blue and yellow squares with numbers

Description automatically generated

**Target variable distribution:**

A screenshot of a computer code

Description automatically generated

A graph of a graph

Description automatically generated

**Feature correlation matrix:**

**A collage of images of different colored squares

Description automatically generated**

**Conclusion**

This project successfully demonstrates the application of machine learning techniques for network intrusion detection using the KDD dataset. Key takeaways include:

1. Random Forest achieves a high accuracy in binary classification tasks.
2. Feature scaling and encoding significantly enhance model performance.
3. Comparative analysis highlights the importance of model selection.
4. Visualization provides critical insights into data distribution and model behavior.