Network Intrusion Detection System using Random Forest and Logistic Regression

This project implements a Network Intrusion Detection System (NIDS) using machine learning techniques. The system aims to classify network activities as either normal or an attack and further identify the type of attack using both binary and multinomial classification methods.

Table of Contents

Introduction

Network Intrusion Detection Systems (NIDS) are crucial for securing computer networks against malicious activities. This project uses machine learning algorithms to build a model that can detect network intrusions and classify them into specific attack types.

Dataset

The dataset consists of various types of network activities labeled as normal or different types of attacks. The data files used are:

- Data_of_Attack_Back.csv
- Data_of_Attack_Back_BufferOverflow.csv
- Data_of_Attack_Back_FTPWrite.csv
- Data_of_Attack_Back_GuessPassword.csv
- Data_of_Attack_Back_Neptune.csv
- Data_of_Attack_Back_NMap.csv
- Data_of_Attack_Back_Normal.csv
- Data_of_Attack_Back_PortSweep.csv
- Data_of_Attack_Back_RootKit.csv
- Data_of_Attack_Back_Satan.csv
- Data_of_Attack_Back_Smurf.csv

Installation

To run this project, you need Python 3.12.4 and the following Python libraries:

- pandas
- numpy
- scikit-learn
- imbalanced-learn

You can install these dependencies using pip:

```
pip install pandas numpy scikit-learn imbalanced-learn
```

Data Preprocessing

1. Load the datasets and assign appropriate column names.
2. Concatenate the datasets into a single DataFrame.
3. Handle missing values by either dropping rows with missing values or filling them.
4. Split the data into features (`X`) and target (`y`) variables.

5. Handle imbalanced data using the SMOTE technique.
6. Standardize the feature variables using `StandardScaler`.

Model Training and Evaluation

Binary Classification

The binary classification model aims to classify network activities as either normal or an attack.

1. Create a binary target variable where 0 represents normal activities and 1 represents attacks.
2. Train a Logistic Regression model on the training data.
3. Evaluate the model using the test data and print the classification report and confusion matrix.

Multinomial Classification

The multinomial classification model aims to classify network activities into specific attack types.

1. Use the original labels for target variable.
2. Create a pipeline that standardizes the data and applies a Random Forest classifier.
3. Train the model on the training data.
4. Evaluate the model using the test data and print the classification report and confusion matrix.

Results

Binary Classification

The Logistic Regression model achieved an accuracy of 99%. The precision, recall, and F1-score for both normal activities and attacks were near or at 1.00, indicating high performance.

Multinomial Classification

The Random Forest model achieved an accuracy of nearly 100%. Precision and recall were very high for most activity types, particularly for 'Neptune' and 'Normal' activities.

Conclusion

The implemented NIDS using Logistic Regression for binary classification and Random Forest for multinomial classification demonstrates high accuracy and performance in detecting and classifying network intrusions. The results show that machine learning techniques can be effectively used to enhance network security.

License

This project is licensed under the MIT License.