Network Intrusion Detection using Decision Trees(DT) and Support Vector Machines(SVM)

This repository contains code for a network intrusion detection system that uses decision trees and support vector machines (SVMs) to classify network traffic as either normal or malicious. The system is based on the NSL-KDD dataset, which includes labeled network traffic data for use in training and testing machine learning models.

The purpose of this work is to explore and improve machine learning techniques that can be applied in the field of cyber security and networking. Also, encouraging other new developers in this field to collaborate and improve their knowledge.

DT-SVM:

The decision tree is constructed based on a training dataset that consists of input samples and their corresponding labels. The decision tree is built by recursively splitting the dataset into smaller subsets based on the values of the features. The splits are chosen to maximize the separation between the different classes of samples.

Once the decision tree is constructed, it can be used to extract relevant features from the input data. For example, if the decision tree splits the data based on the value of feature A, then feature A is likely to be an important feature for classification. By following the path through the decision tree for a given input sample, the relevant features can be identified and extracted.

The extracted features are then fed into an SVM classifier, which uses them to make a final classification decision. The SVM is trained on a separate dataset, which consists of the extracted features and their corresponding labels. The SVM learns a decision boundary that separates the different classes of samples based on the values of the features.

By using a decision tree to pre-process the input data and extract relevant features, the SVM can focus on the most informative features for classification, which can improve its performance. Additionally, the decision tree can reduce the dimensionality of the input data, which can improve the efficiency of the SVM. However, it is important to carefully tune and validate the decision tree and SVM to ensure that the overall system is effective.

Installation

To run the code in this repository, you will need to have Python 3 and several Python libraries installed, including scikit-learn, pandas, and numpy. You can install these libraries using pip:

```
pip install scikit-learn pandas
pip install pandas
pip install numpy
pip install matplotlib.pyplot
pip install seaborn
```

Usage

The main script for this system is DT_SVM.ipynb, which contains code for loading the NSL-KDD dataset, training the decision tree, and SVM models on the data, and evaluating the performance of various metrics models by comparing the accuracy and score.

You can also modify the script to experiment with different parameters or algorithms for training and testing the models. Finally, the tunning hyperparameters section can take a lot of time and computation power to provide results at the same by reducing the max_depth, max_features, C, and gamma take less time.

License

This code is released under the MIT License. See the LICENSE file for more information.