MP III -- Marshall Thompson

MBD System

CODE: https://github.com/mdt48/CSCI-6907-SecureAutonomousSystems/tree/main/mp3

Parsing Data

The first step of this MP was devising a method to accurately parse the data. Since the data was not valid JSON to begin with, this proved to be tricky! Using the `training_key.csv` file, I aggregated all of the incoming messages speed, acceleration, position, heading and label (0 for genuine, 1 for attacker) into a data frame. The x and y components for speed, acceleration, position, and heading, and the noise values associated with them are the features I used for classification.

My Approach

Overview

When this project was assigned to us, I immediately wondered whether a machine learning approach would be feasible for this problem. As the project specification notes, this is a binary classification problem. Either a car is an attacker, or it is genuine.

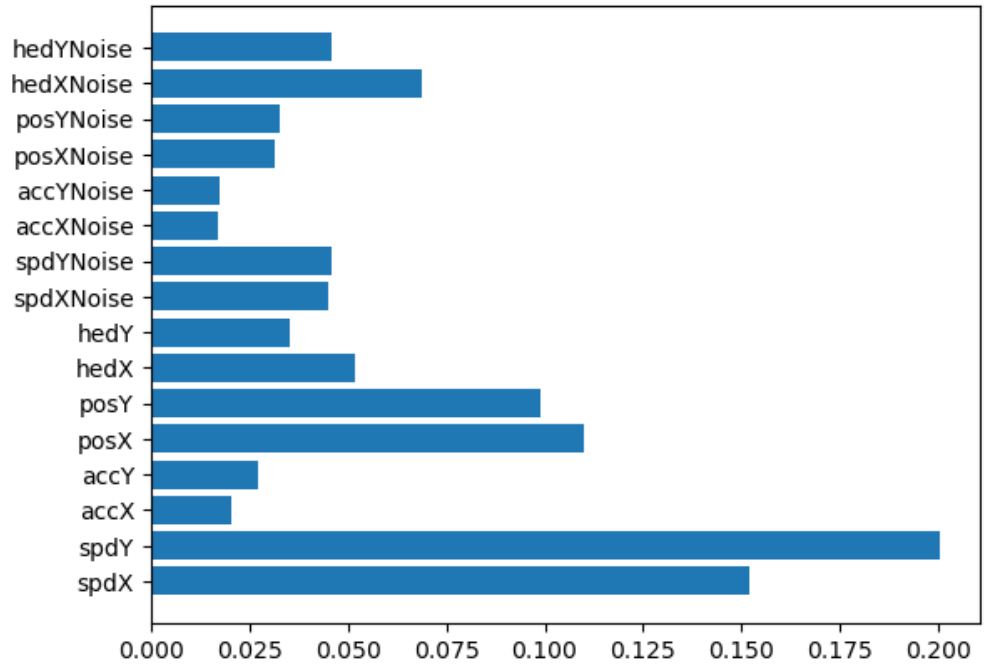
NOTE: The neural network and clustering algorithms were unable to be evaluated on the leaderboard as I started receiving server errors 😦)

Logistic Regression

As a naive, first approach I tried Logistic Regression to gather a better understanding of the problem. The once trained, the Logistic Regression model achieved an F1 score of 0.47. This means that I was doing worse than randomly guessing. A potential explanation of this result is due to the dimensionality of the data. Logistic Regression models are known to become overfit as the dimensionality of the data increases -- as more predictors/features are added to the data, it makes it harder for a Logistic Regression model to make a generalization.

Decision Tree

Since the Logistic Regression performed so poorly, I wanted to investigate whether certain features (speed, acceleration, etc.) were more important than others. To do this I trained a Decision Tree on the training data. Below you can see the level of feature importance calculated by the decision tree with entropy as the units.



As the graph suggests, speed seems to be more important to make a classification than others. This makes sense. Several of the attacks utilize some sort of modification to the vehicle's speed. Whether thats a random speed, random offset, fixed offset, or fixed speed. The same can be said about position. There are a few other attacks that focus on position, hence its relative importance compared to the other features.

"Feature Importance" can be rephrased as the average level of information that each feature gives us. This means that looking at speed first will give us the most information on whether a vehicle is attacker or genuine, followed by position and the other features.

Using a singular decision tree, I was able to achieve an F1 Score of 0.64.

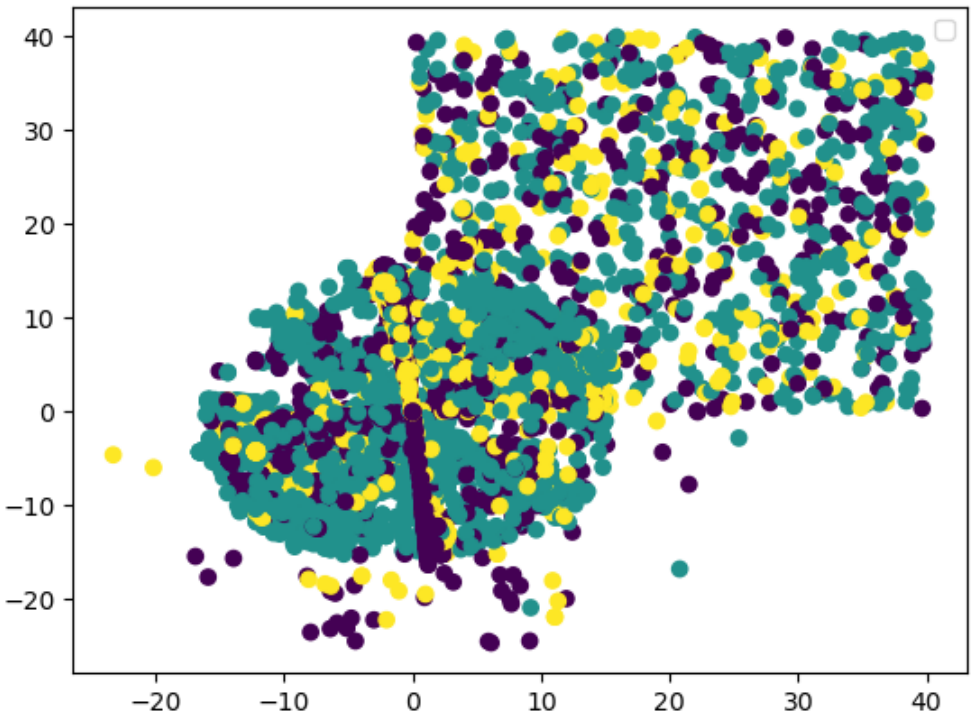
Random Forest

Since it was discovered that there are features that are more important (namely speed and position), a Random Forest (a collection of decision trees that "vote" for the best answer) is likely to provide a more accurate answer than a single decision tree. Furthermore, a single decision tree can be affected by its random initialization, and a collection of decision trees is more likely to yield an accurate result.

Using 100 decision trees in the forest, a F1 score of 0.677 was achieved.

Nerual Network

In the `NN.ipynb`, I created a binary classification neural network. This was a basic binary classification neural network written using Pytorch. As noted in the overview section, I was unable to evaluate this model on the test data on the leaderboard due to a server error.



Results and Code

Here is the link to the code: https://github.com/mdt48/CSCI-6907-SecureAutonomousSystems/tree/main/mp3

The results from the Random Forest that yielded the 0.677 F1 score are in the final.csv file in the above repo.

All models except for the neural network can be foundd in the `models.ipynb`. The neural network can be found in `NN.ipynb`.