Misbehaviour Detection for Position Falsification Attacks in VANETs Using Machine Learning

A Report Submitted
in Partial Fulfilment of the Requirements
for the Degree of
Bachelor of Technology
in
Computer Science & Engineering

by
Aditya Raju Darji [20194008]
Deepesh Manoj Rathi [20194066]
Aryan Khandelwal [20194142]
Anand Tripathi [20194119]
Adithya Balaji [20194135]

to the
COMPUTER SCIENCE AND ENGINEERING DEPARTMENT
MOTILAL NEHRU NATIONAL INSTITUTE OF TECHNOLOGY ALLAHABAD PRAYAGRAJ
April, 2022

UNDERTAKING

I declare that the work presented in this report titled “Misbehaviour Detection for Position Falsification Attacks in VANETs Using Machine Learning”, submitted to the Computer Science and Engineering Department, Motilal Nehru National Institute of Technology Allahabad, Prayagraj, for the award of the *Bachelor of Technology* degree in *Computer Science & Engineering*, is my original work. I have not plagiarized or submitted the same work for the award of any other degree. In case this undertaking is found incorrect, I accept that my degree may be unconditionally withdrawn.

April, 2022
Allahabad

_____

Aditya Raju Darji [20194008]

_____

Deepesh Manoj Rathi [20194066]

_____

Aryan Khandelwal [20194142]

_____

Anand Tripathi [20194119]

_____

Adithya Balaji [20194135]

CERTIFICATE

Certified that the work contained in the report titled "*Misbehavior Detection for Position Falsification Attacks Using Machine Learning*", by *Aditya Raju Darji [20194008], Deepesh Manoj Rathi [20194066], Aryan Khandelwal [20194142], Anand Tripathi [20194119], Adithya Balaji [20194135]*, has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

(Dr. Dinesh Singh)

Computer Science and Engineering Dept,
M.N.N.I.T, Allahabad

Preface

As a part of the B.Tech Curriculum and gain practical experience, we are required to make a report on “Misbehaviour Detection for Position Falsification Attacks in VANETs Using Machine Learning". The Basic Objective behind doing this project report is to prepare a model that can help detect attackers in VANETs and also learn the practical applications of various classification models in real life.

In this project report we use the results of a vehicular network simulation which is used to train the model. Algorithms like KNN, RF and Logistic Regression are used for this purpose and their results are combined using ensemble learning.

Doing this Project report helped us to enhance our knowledge of how Classification Algorithms work. We also learnt about VANETs and how an attacker can attack another vehicle using falsification attacks. The report helped us in understanding the importance of teamwork and devotion to the work.

Acknowledgements

It is a great pleasure to thank the giants on whose shoulders I stand. Firstly, I would like to thank my supervisor Dr. Dinesh Singh for giving us this opportunity and providing constant support, guidance and encouragement. His innovative ideas and zeal to motivate and help us have led to the successful completion of this project. He has provided us the opportunity to explore the content and aspects of this project. We felt privileged working under him.

We would also like to express our sincere gratitude to Prof. R.S. Verma, Director, MNNIT Allahabad, Prayagraj, Prof. D.S. Kushwaha, Head, Computer Science And Engineering Department and Dr. Shashank Srivastava, Head, Departmental Undergraduate Committee (DUGC) for providing us with the tools and facilities to complete the project.

Lastly, we would like to thank our friends and family for their support, encouragement and advice. Their support surely played a part in the completion of the project.

Contents

Chapter 1

Introduction

Cooperative Intelligent Transport Systems (C-ITS) allow effective communication through wirelessly with other vehicles to provide road safety. Vehicular ad hoc networks (VANETs) are mobile networks where each node in the network is a vehicle. They consist of a set of vehicles equipped with an on board unit (OBU) to enable communications.

Security is a major part of these communications. Attackers can exploit them to threaten both the security and safety of the passengers. There are different types of attacks like Sybil, Denial of Service, black-hole and data injection attacks. These attacks can be dealt with entity centric misbehaviour schemes and intrusion detection schemes.

Reputation schemes use the reputation of each node. The reputation of each node can be calculated by various approaches to detect less trusted vehicles, namely misbehavior vehicles. The lower the reputation of the vehicle, the lower the trust. This scheme is generally implemented to enhance the performance of routing protocols. mistrusted vehicles are detected and can be excluded from the route selection process.

So, we propose a machine learning model which takes advantage of Received signal Strength Indicator(RSSI) which is related to the sender's position and can help in the detection of attacks. We use K-Nearest Neighbour, Random Forest and Logistic Regression to detect the mistrusted vehicles. Further, we use ensemble learning which combines the results of these to get a more effective result.

1.1 Motivation

Today's transportation system has become an essential and extensive part of daily human life and is an important part of human activity.In a survey, It is estimated that 45% of people in the transit system spend an hour on average. All this time used for transportation gives ample opportunities for attackers to exploit the security of the VANETs. To prevent such attacks we use a Misbehaviour detection system which can help identify attackers by training a model based on previous attacker logs.

Not everyone has access and knowledge of the working of VANETs and classification algorithms like KNN, RF and Logistic Regression. So, our objective is to create an attack detection system which can easily be accessed by anyone.

We, as students of computer science, have decided that with the given data, we can use machine learning top make the system as it is very difficult for a human to classify each message from hundreds of message logs quickly and accurately

1.1.1 Some Wonderful Minds

SECIL ERCAN received the Ph.D. degree in industrial engineering from Istanbul Technical University, Turkey, in 2017. She has studied stochastic scheduling of energy systems in her Ph.D. thesis. She is currently a Research Engineer with the University of Reims Champagne-Ardenne, France. Her current research interests include optimization, mathematical modeling, data science, and artificial intelligence in intelligent transport systems.

MARWANE AYAIDA received the Diploma degree in electronics of embedded systems, Cergy-Pontoise, France and the Ph.D. degree. His research interests include interoperability modeling for embedded systems in the eld of transportation. Also, his work focuses on vehicular communications (V2V and V2I). Specifically, he studies the routing protocols in Vehicular Ad-hoc Networks (VANETs). He has been an Associate Professor with the University of Reims, since September 2013.

Chapter 2

Related Works

VANETs may suffer from several security issues related to authentication, privacy, data non-repudiation, etc. To overcome these issues, IDS, which were initially developed for classical computer networks, are more and more used to enhance the security in VANETs. This section aims to present a brief overview of the works related to misbehavior detection in VANETs with a specific focus on machine learning detection techniques. Moreover, on the existing approaches, IDS classification is applied according to the actor that decides whether a vehicle is an attacker or not.

2.1 Technology to use

Python language is versatile, easy to use, fast to develop and it's an open source with a vibrant community. Python also enables us to use multiple modules and it is the most used language for machine learning.

Matplotlib is the most powerful, versatile and easy to use data visualisation library in python and mainly it is compatible with other popular libraries such as

numpy and pandas.

Pandas is the most powerful data handling library, it has good support for popular data formats such as CSV, JSON, xlsx and XML and mainly it makes data flexible and customizable.

We had opted Sklearn for machine learning mainly because it is one of the most used ML libraries in python and contains lots of efficient tools for ML and statistical modelling including classification, regression, clustering and data preprocessing and easy to use.

Chapter 3

Proposed Work

To detect malicious vehicles, we propose to implement kNN, RF and Logistic Regression, which are mostly used in misbehavior detection and yield satisfying results. These three ML techniques will be performed together in order to obtain higher classification performances. Hence, an Ensemble Learning method is proposed to carry out the proposed feature combinations.

We use the dataset "**VeReMi**" to make a model for misbehaviour detection. **VeReMi** is a simulated dataset. It consists of message logs per vehicle, containing both GPS data about the local vehicle and BSM messages received from other vehicles though DSRC.

In this model we only need BSM messages as attacks are usually done Vehicle to Vehicle. The dataset can be accessed from github.com/VeReMi-dataset/VeReMi/releases/tag/v1.0.
The data set contains multiple simulation log files with simulation logs available for each vehicle.

Dataset file structure:

Each simulation log contains a ground truth file for every message and a set of message logs for every vehicle that received messages. The file name of a message log identifies the receiver by vehicle number and OMNeT++ module number.

Eg JSONlog-0-7-A0.json here A0 shows that vehicle is not an attacker whereas A1 means that it is an attacker. Each vehicle log contains message id, position, speed,RSSI, and noise.

In this case we have decided to use the first simulation log as it has a relatively simpler log. which makes it easier to process.

Features:

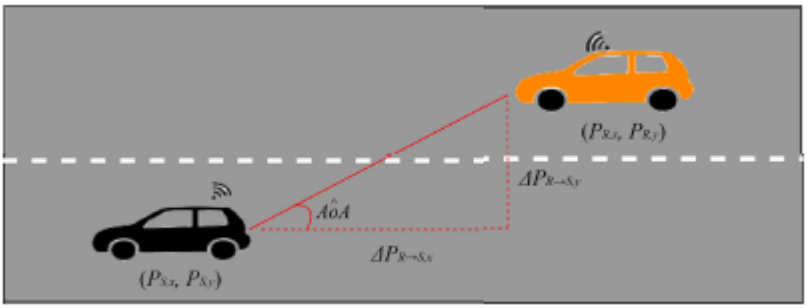
To detect position falsification attacks, the Position (P) and Speed (S) are considered in this work for both the receiver (R) and the sender (S). These features are commonly used in the previous works in the literature [41], and are respectively

denoted: $P_{R,\,a}$, $P_{S,\,a}$, $S_{R,\,a}$ and $S_{S,\,a}$, where x and y stand for the 2D map dimension axes.

$$\Delta P_{S,a}(i) = P_{S,a}(i) - P_{S,a}(i-1)$$
$$\Delta S_{S,a}(i) = S_{S,a}(i) - S_{S,a}(i-1)$$

$$\Delta P_{R,a}(i) = P_{R,a}(i) - P_{R,a}(i-1)$$
$$\Delta S_{R,a}(i) = S_{R,a}(i) - S_{R,a}(i-1)$$

$$AoA = \arctan \frac{\Delta P_{R \to S,y}}{\Delta P_{R \to S,x}}$$



$$RSSI = P_T - P_L(d)$$

Preprocessing the dataset:

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib as plt
import json

atk = pd.read_csv('data.csv')
print(atk)

atk.drop(['noise','spd_noise','pos_noise'], inplace=True, axis=1)
atk

atk1 = pd.DataFrame(atk['pos'].tolist(), columns=['pos-x', 'pos-y', 'pos-z'])
atk1.drop(['pos-z'], inplace=True, axis=1)
atk1

atk2 = pd.DataFrame(atk['spd'].tolist(), columns=['spd-x', 'spd-y', 'spd-z'])
atk2.drop(['spd-z'], inplace=True, axis=1)
atk2

frames1=[atk,atk1,atk2]
attack=pd.concat(frames1,axis=1)
attack

attack.drop(['pos','spd'], inplace=True, axis=1)
```

	type	rec_tim	send_time	sender_id	msg_id	RSSI	sender_p_x	sender_p_y	sender_spd_x	sender_spd_y	rec_p_x	rec_p_y	rec_spd_x	rec_spd
0	3.0	10900.0	10900.0	211.0	30600.0	3.050000e-09	3590.0	5640.0	-2.63	32.1	3650.0	5180.0	-5.80	37
1	3.0	10900.0	10900.0	175.0	30800.0	3.490000e-08	3640.0	5250.0	4.98	-31.6	3640.0	5200.0	-5.79	37
2	3.0	10900.0	10900.0	211.0	31000.0	4.440000e-09	3590.0	5670.0	-1.33	31.9	3640.0	5220.0	-5.79	37
3	3.0	10900.0	10900.0	205.0	31100.0	1.280000e-09	3580.0	5840.0	-0.04	39.0	3640.0	5220.0	-5.78	37
4	3.0	10900.0	10900.0	193.0	31100.0	1.330000e-09	3590.0	5720.0	-1.32	31.7	3640.0	5230.0	-5.78	37

send_time	sender_id	msg_id	RSSI	sender_p_x	sender_p_y	sender_spd_x	sender_spd_y	rec_p_x	rec_p_y	rec_spd_x	rec_spd_y	rec_id	attacker
10900.0	211.0	30600.0	3.050000e-09	3590.0	5640.0	-2.63	32.1	3650.0	5180.0	-5.80	37.2	217.0	0
10900.0	175.0	30800.0	3.490000e-08	3640.0	5250.0	4.98	-31.6	3640.0	5200.0	-5.79	37.1	217.0	0
10900.0	211.0	31000.0	4.440000e-09	3590.0	5670.0	-1.33	31.9	3640.0	5220.0	-5.79	37.1	217.0	0
10900.0	205.0	31100.0	1.280000e-09	3580.0	5840.0	-0.04	39.0	3640.0	5220.0	-5.78	37.1	217.0	0
10900.0	193.0	31100.0	1.330000e-09	3590.0	5720.0	-1.32	31.7	3640.0	5230.0	-5.78	37.0	217.0	0

Training the model for generating better results:

Splitting the dataset into training and testing dataset.

```
from sklearn.model_selection import train_test_split
#split data into train and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.35, stratify=y)
```

Splitting the dataset into training and testing dataset.

```
from sklearn.ensemble import RandomForestClassifier
#create a new random forest classifier
rf = RandomForestClassifier()
#create a dictionary of all values we want to test for n_estimators
params_rf = {'n_estimators': [50, 100, 200]}
#use gridsearch to test all values for n_estimators
rf_gs = GridSearchCV(rf, params_rf, cv=5)
#fit model to training data
rf_gs.fit(X_train, y_train)
```

Applying KNN classification on the dataset.

```
import numpy as np
from sklearn.model_selection import GridSearchCV
from sklearn.neighbors import KNeighborsClassifier
#create new a knn model
knn = KNeighborsClassifier()
#create a dictionary of all values we want to test for n_neighbors
params_knn = {'n_neighbors': np.arange(1, 25)}
#use gridsearch to test all values for n_neighbors
knn_gs = GridSearchCV(knn, params_knn, cv=5)
#fit model to training data
knn_gs.fit(X_train, y_train)
```

Applying RF classification method on the dataset.

```
clf = RandomForestClassifier(n_estimators = 10, max_depth=5, random_state=0)
clf.fit(feat_x, Y)

X_eval = my_data[40:,:-1]
print(X_eval.shape)
feat_x_eval = x_to_feat(X_eval)
print(feat_x_eval.shape)

Y_eval = my_data[40:,-1]
print(Y_eval.shape)
print(clf.predict(feat_x_eval))
print(Y_eval)
print(clf.predict(feat_x_eval)-Y_eval)
```

Applying Logistic Regression on the dataset.

```
from sklearn.linear_model import LogisticRegression
#create a new logistic regression model
log_reg = LogisticRegression()
#fit the model to the training data
log_reg.fit(X_train, y_train)
```

Predicting final result using ensemble learning model (Voting Classifier is Used).

```
from sklearn.ensemble import VotingClassifier
#create a dictionary of our models
estimators=[('knn', knn_best), ('rf', rf_best), ('log_reg', log_reg)]
#create our voting classifier, inputting our models
ensemble = VotingClassifier(estimators, voting='hard')
```

Chapter 4

Experimental Setup and Results Analysis

4.1 Setup

Downloading the database:

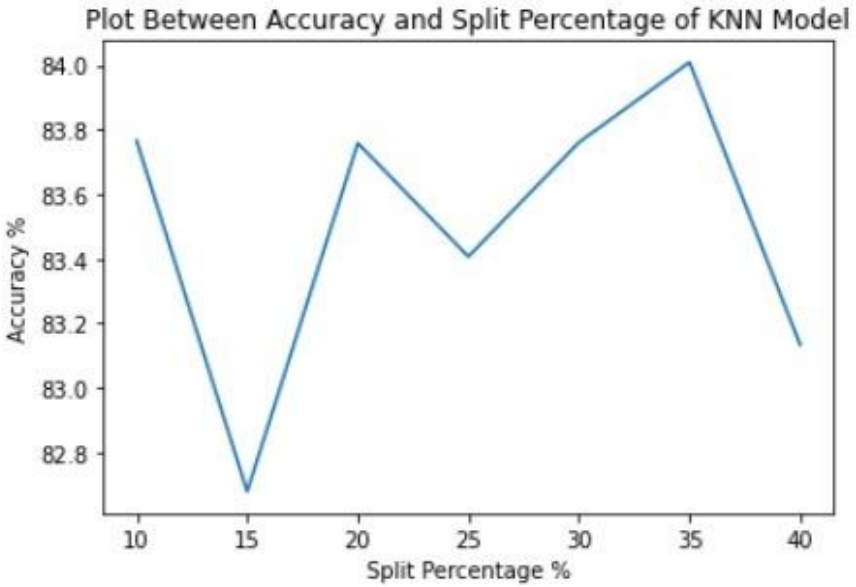
The dataset can be accessed from github.com/VeReMi-dataset/VeReMi/releases/tag/v1.0.
The data set contains multiple simulation log files with simulation logs available for each vehicle.

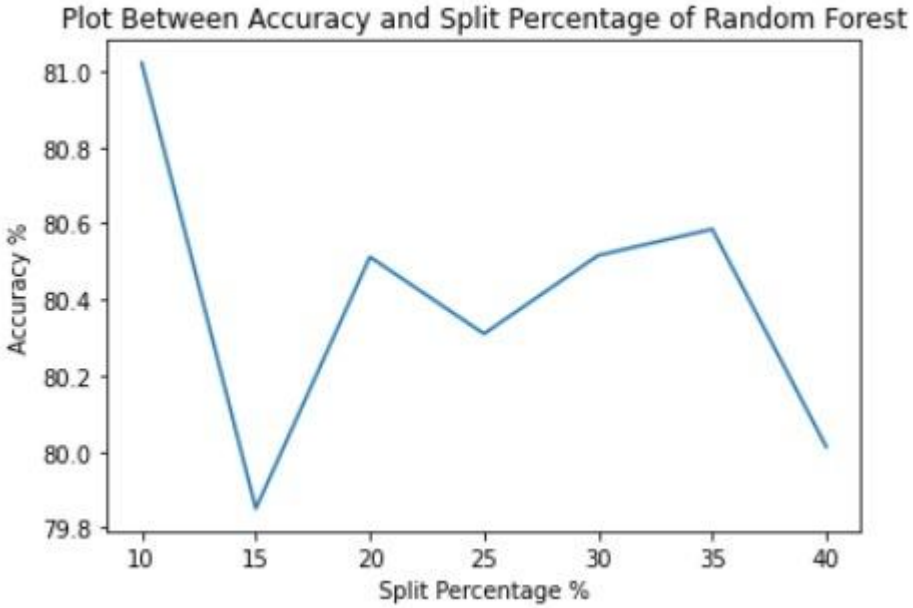
Setting up the ML environment:

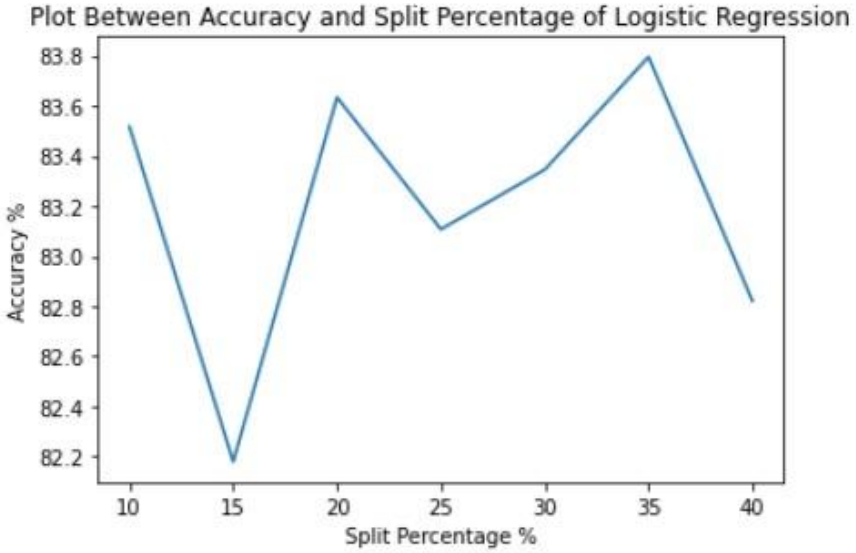
1. Download Python 3 from https://www.python.org/downloads/
2. Install jupyter notebook
3. Install various packages (numpy, matplotlib, scikit-learn, etc)
4. You are good to go.

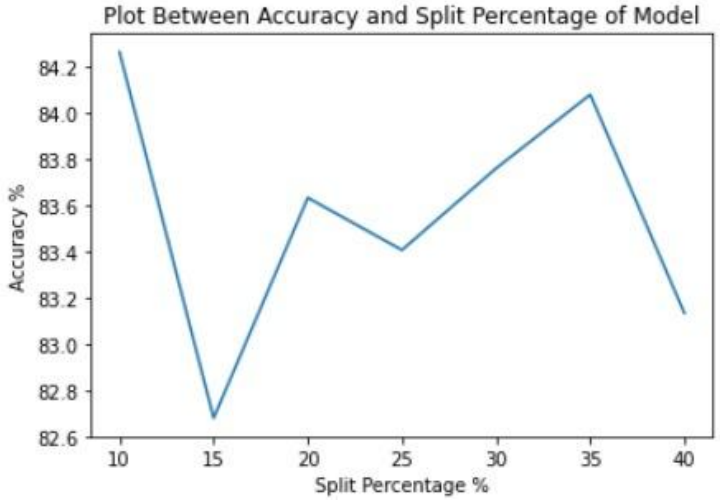
4.2 Result Analysis

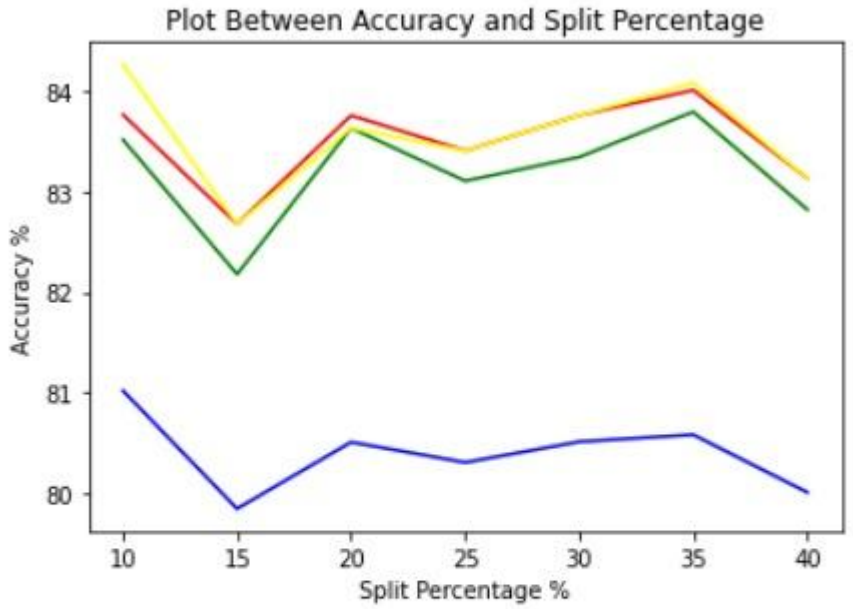
Following are some graphs plotted against various split percentages to get the accuracy of the model on various splits.




Plot Between Accuracy and Split Percentage of Random Forest
Accuracy %
81.0
80.8
80.6
80.4
80.2
80.0
79.8
10
15
20
25
30
35
40
Split Percentage %


Plot Between Accuracy and Split Percentage of Logistic Regression
Accuracy %
83.8
83.6
83.4
83.2
83.0
82.8
82.6
82.4
82.2
10
15
20
25
30
35
40
Split Percentage %


Plot Between Accuracy and Split Percentage of Model
Accuracy %
84.2
84.0
83.8
83.6
83.4
83.2
83.0
82.8
82.6
10
15
20
25
30
35
40
Split Percentage %



Red - KNN
Blue - RF
Green - LR
Yellow - Ensemble

Data Plotted:

[

KNN = [83.76558603491272, 82.67886855241264, 83.75780274656679, 83.40659340659341, 83.7603993344426, 84.00855920114123, 83.13358302122348]

RF = [81.02244389027432, 79.85024958402662, 80.51186017478152, 80.3096903096903, 80.51580698835274, 80.5848787446505, 80.01248439450686]

LR = [83.51620947630923, 82.17970049916805, 83.63295880149812, 83.10689310689311, 83.34442595673876, 83.79457917261056, 82.82147315855181]

Ensemble = [84.2643391521197, 82.67886855241264, 83.63295880149812, 83.40659340659341, 83.7603993344426, 84.07988587731811, 83.13358302122348]

Percentage Split = [10, 15, 20, 25, 30, 35, 40]

]

As it can be inferred that individually these classification techniques have some holdbacks. But when we apply ensemble learning over these techniques there is a boost in accuracy. Evidently, the model generated has better prediction techniques.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

After collecting, analysing, cleaning the data and making the model to predict the Misbehavior Detection for Position Falsification Attacks. We understand that Machine learning has a potential to make a big impact on the security of VANETs and passengers' lives ultimately.

In conclusion, we would like to say that our work is just a beginning, it can be considered as a starting step for a bigger and better platform which might be the one-stop solution for VANETs security.

Also a big Thank You to our mentor for guiding us and motivating us for this project.

5.2 Future Work

Currently, we worked on the dataset and model generation. We will provide a user interface to this model, so that users can interact with the app and we could take road safety to their door.

We will also work on our model to improve the accuracy a little more. We would also study different types of attacks to a vehicle and how to detect them as well.

There is a scope of taking a track to hardwares and creating a mock VANET attack environment. The boundary is limitless and we are the explorers.

References

[1] Dataset : https://github.com/VeReMi-dataset/VeReMi/releases/tag/v1.0

[2] How to access Dataset : https://veremi-dataset.github.io/

[3] Model used for machine learning: sklearn.svm.SVR — scikit-learn 0.24.2 documentation