Politecnico di Torino

Master's Degree in Automotive Engineering

Enhancing Road Safety and Energy Efficiency through Driving Behavior Detection Using Machine Learning Methods

Candidate: Hao Chen

Supervisors: Prof. Angelo Bonfitto
Prof. Shailesh Sudhakara Hegde

Academic Year 2023/24

Summary

A significant rise in traffic fatalities linked to aggressive driving behaviors has been noted, accentuating the imperative need for research in this domain. Hence, the detection of aggressive driving is increasingly advocated as a strategy not only to alert drivers about their perilous behaviors but also to potentially diminish the incidence of accidents. In addition, driving style significantly affects the fuel efficiency of traditional internal combustion engine vehicles, and modifying driving behavior can effectively improve fuel economy. Compared to traditional vehicles, driving style has a greater impact on the range of electric vehicles. In parallel, the widespread adoption of driving simulators in the automotive sector has unveiled their profound advantages, especially in terms of repeatability in a controlled environment, helping researchers significantly reduce the time and cost of development. Furthermore, driving simulators also provide a safe platform for testing new technologies and evaluating driver behavior in various scenarios.

This thesis presents a method for identifying aggressive driving by analyzing electric vehicle dynamics data (such as speed, acceleration, and steering angle) collected from simulation scenarios using the software SCANeR™Studio.

The algorithm uses iterative density-based spatial clustering of noise applications (an unsupervised learning technique) to cluster aggressive driving maneuvers and sub-classify driving behaviors based on comfort, safety, and efficiency. In addition, these labeled data are used to train Bayesian optimization-based long short-term memory neural network and a random forest model. The research results demonstrate that excels in accurately identifying energy-efficient driving behaviors and aggressive driving behavior, with a F-score 0.992 and 0.869, showing great potential in enhancing road safety as well as promoting vehicle energy conservation and sustainable driving practices.

Table of Contents

List of Tables

List of Figures

List of Algorithms

Acronyms

ACC

Adaptive Cruise Control

AD2

Aggressive Driving Detection

AI

Artificial Intelligence

CNN

Convolutional Neural Network

DBSCAN

Density-Based Spatial Clustering of Applications with Noise

EDB

Elementary Driving Behaviors

eSiM

ecoSituational Model

I-DBSCAN

Iterative Density-Based Spatial Clustering of Applications with Noise

ICE

Internal Combustion Engine

LSTM

Long Short-Term Memory

MP

Monitoring Period

NARX

Nonlinear Autoregressive Model with eXogenous Inputs

NGSIM

Next Generation Simulation Project

NHTSA

National Highway Traffic Safety Administration

PCA

Principal Component Analysis

RF

Random Forest

SRT

Speed Reduction Time

SVM

Support Vector Machine

TLC

Time to Lane Crossing

TTC

Time to Collision

V2X

Vehicle-to-Everything Communication

Chapter 1

Introduction

Referring to the most recent "Global Status Report on Road Safety 2023" released by the World Health Organization, despite a yearly decrease of 5% in the number of deaths from road traffic accidents since 2010, the figure has declined to 1.19 million per year by 2023. However, more than 2 people still lose their lives every minute, and over 3,200 people die each day, a staggering statistic. What is additionally concerning is that road traffic accidents remain a leading cause of death for individuals aged 5 to 29. Hence, road traffic accidents continue to represent an ongoing global health crisis [1].

According to the NHTSA's 2020 statistics, 66% of traffic fatalities in nationwide collisions were attributed to aggressive driving. More than 78% of U.S. drivers reported having engaged in at least one aggressive driving behavior at least once in the past year [2]. Based on research by the AAA, aggressive driving is typically defined by several driving behaviors [3]:

○ Speeding in heavy traffic

○ Tailgating

○ Cutting in front of another driver and then slowing down

○ Running red lights

○ Weaving in and out of traffic

○ Changing lanes without signaling

○ Blocking cars attempting to pass or change lanes

In addition, driving style significantly affects the fuel efficiency of traditional ICE vehicles, and modifying driving behavior can effectively improve fuel economy

[4]. Compared to traditional vehicles, driving style has a greater impact on the range of electric vehicles [5, 6].

1.1 Application of Driving Simulator

Identifying potential driving behaviors aids in predicting when drivers might engage in such actions. This alerts drivers to their behavior, promoting responsible and cautious driving habits. Consequently, it reduces individual drivers' risky behaviors, significantly lowering the chance of accidents.

However, When conducting research on actual roads, there are many obstacles, including time constraints, material costs, and safety issues. To address these challenges, the use of driving simulators in automotive research has become increasingly common in recent years. Driving simulators are devices that simulate real driving environments, typically consisting of a vehicle cockpit, steering wheel, pedals, display screens, and control and monitoring systems related to vehicle operation. By using driving simulators, researchers can simulate various road and traffic conditions in a controlled environment for tasks such as vehicle performance testing, driver behavior studies, and the development of intelligent driving systems. The use of these simulators offers numerous advantages. For instance:

- **Time and Cost Efficiency**: Driving simulators allow researchers to conduct a large number of tests in a shorter amount of time without the extensive time and resource investment required for real-world testing.
- **Safety**: Testing in a simulator eliminates the safety risks associated with driving on actual roads, protecting both researchers and test equipment.
- **Environmental Control**: Researchers can precisely control various road conditions, weather conditions, and traffic situations in the simulator to better understand vehicle performance and driver behavior in different scenarios.
- **Data Recording and Analysis**: Driving simulators typically feature data recording and analysis capabilities, allowing for accurate recording and analysis of driver behavior, vehicle performance, and system responses to support research with reliable data.
- **Repeatability**: Tests conducted through simulators can be easily repeated to ensure the reliability and consistency of research results.

By using driving simulators, researchers can simulate various road and traffic conditions in a controlled environment for tasks such as vehicle performance testing, driver behavior studies, and the development of intelligent driving systems. For instance, to study the influence of adaptive cruise control on human behavior [7] and the eye movements under various time periods and weather conditions [8] by using driving simulator. In addition, the driving simulator is also used carry out experiments by deploying machine learning algorithms to reduce traffic congestion [9].

1.2 State of the art

1.2.1 machine learning methods

By nature, the recognition of driving behavior is a complex and challenging task. A single data element may not provide sufficient information to accurately judge driving behavior. By utilizing multidimensional data, such as vehicle speed, acceleration, steering angle, and environmental information around the vehicle, a more comprehensive understanding of the driver's behavior can be achieved.

Machine learning techniques play a significant role in this area. By analyzing and training on large amounts of driving data, machine learning algorithms can learn the patterns and features of driving behaviors, thereby enabling the automatic recognition and classification of driving actions. For example, supervised learning algorithms can be used to train models that predict the driver's actions, such as accelerating, decelerating, and turning, based on the input multidimensional data. The use of sensor data such as vehicle and engine speed are used to train a support vector machine (SVM) classifier to identify aggressive driving behaviour yielding a detection rate of 93.1% [10]. An emotion detection method based on convolutional neural network (CNN) to identify driving behaviour is proposed in [11]. A random forest (RF) model-based analysis of time to lane crossing (TLC) in real-world scenarios was proposed, successfully identifying aggressive driving behavior on horizontal curves with an accuracy of 95.34% [12]. The extreme learning machine (Elman) neural network and electric vehicle motor efficiency were used to construct an economic evaluation model to explore the impact of driving behavior on energy consumption [13]. The ecoSituational Model (eSiM) prediction model uses digital map data, sensor data, and V2X communication information to predict vehicle speed distribution and optimize driving behavior through adaptive cruise control (ACC) systems, thereby reducing vehicle fuel consumption [14].

A detailed comparative analysis was performed on several state-of-the-art aggressive driving event detection algorithms (SVM, RF) across AD2 (Aggressive Driving

Detection) datasets encompassing diverse devices, vehicles, and locations [15]. Two deep learning models; convolutional neural network and recurrent neural network were deployed to categorize driving behavior into two classes: normal driving and aggressive driving [16]. A Long Short-Term Memory (LSTM) neural network optimized using Bayesian techniques was applied to 3,855 samples obtained from the NGSIM (Next Generation Simulation Project). This model yielded commendably high performance in predicting aggressive driving behavior [17]. A study improved the control performance of electric vehicle energy management by 27% compared to existing models by using an artificial neural network called nonlinear autoregressive model with eXogenous inputs (NARX) to model and predict driving behavior in the next 30 seconds [18]. On the other hand, research were carried to improve the comfort, safety and traffic efficiency using machine learning approach [19, 20] and control strategies to avoid collision [21].

By continuous advancements in machine learning technology and the accumulation of data, driving behavior recognition technology will become increasingly accurate and reliable, providing significant support for improving traffic safety and driving experience.

1.2.2 Other methods

Not only machine learning methods, but also threshold-based and anomaly detection methods are used to identify aggressive driving. The thresholds in longitudinal acceleration and deceleration, lateral acceleration and yaw rate for aggressive driving behavior is set in [22]. The method of recognizing race car driving skills based on wavelet transform and Lipschitz singularity detection theory is demonstrated in [23]. Authors in [24] proposed a method to describe the relationship between lateral, longitudinal acceleration and speed and provided a good direction for studying the classification of car driver behavior. Work in [25] analyzed time to collision (TTC) and speed reduction time (SRT) of drivers in the event of an emergency collision using a Generalized Linear Mixed Model and a Weibull Accelerated Failure Time model, respectively. It is reported that the TTC and SRT of aggressive drivers decreased by 82% and 38%, respectively when compared with normal drivers. The study on analyzing driving behaviors and identifying aggressive driving behaviors has been consistently followed and pushed forward by various scholars. Together, these methods provide effective tools for understanding and improving driving safety.

1.3 Energy consumption, safety, and comfort during driving

Existing studies have primarily focused on identifying aggressive driving; however, a comprehensive analysis of the impact of driving behavior on passenger comfort, and fuel/energy efficiency remains insufficient. Therefore, it is particularly important to further investigate the specific effects of aggressive driving on vehicle energy consumption and driving experience. Prior studies have demonstrated that modifications in driver behavior can substantially increase vehicle fuel/energy efficiency by as much as 30% [26]. Additionally, according to references [27, 28], vibrations transmitted to passengers and the probability of motion sickness have been used to optimize passenger comfort. The 100-kilometer power consumption calculated through the voltage, current, and sampling time of the vehicle motor was used to explore the impact of driving behavior on energy economic [13].

1.4 Contribution

In the literature, driver behavior analysis is performed on a dataset with few features and often overlooks information such as pedal position. To overcome this, we use a driving simulator (see Fig. 1.1) to gather all relevant features for predicting driving behavior. In this research work, we employ Iterative Density Based Spatial Clustering of Applications with Noise (I-DBSCAN), an unsupervised learning technique for classification of aggressive behaviors. Subsequently, a threshold based on Jerk and TTC is used to further categorize driving behaviors into normal, aggressive, discomfort, and risky driving behaviors. This classification is used to train a recognition model called the "discomfort and risk driving recognition model".

Another recognition model, the "efficiency and aggressive driving recognition model", is trained by using a classification based on motor efficiency thresholds. It includes three classes: high-efficiency, low-efficiency, and aggressive driving behavior. Finally, a labeled dataset is fed into a Bayesian optimization-based LSTM neural network and an RF to predict driving behavior. The performance of the proposed models is evaluated using the F-score as the key performance indicator.

1.5 Outline

The rest of the thesis is organized as follows:

Chapter 2 provides a detailed overview of the driving simulator and software environment used for driving behavior data collection and analysis. By utilizing a driving simulator in conjunction with SCANeR^{TM}Studio, the study can accurately replicate real-world driving conditions and collect extensive data on various driving behaviors.

Chapter 3 elaborates on the data collection process, where data is gathered from a simulator equipped with sensors capable of capturing speed, steering angle, and other relevant metrics. The data is then preprocessed to remove noise and inconsistencies and segmented into elementary driving behaviors (EDB). K-means clustering is used to classify these EDB. The I-DBSCAN algorithm is used to label aggressive behaviors, Then further classify aggressive behaviors based on jerk and TTC thresholds into discomfort, risky, and aggressive driving behavior which do not influence comfort and safety for training the "Discomfort and Risk Driving Recognition model". Normal driving is classified based on motor efficiency thresholds into low-efficiency and high-efficiency driving behaviors for training the "Efficiency and Aggressive Driving Recognition model". Bayesian optimization LSTM neural network, and RF. These models are evaluated based on their performance in detecting driving behaviors.

Chapter 4 shows the training results of the "Discomfort and Risk Driving Recognition model" and the "Efficiency and Aggressive Driving Recognition model". The results indicate that both the LSTM neural network and the RF exhibit high performance, with the LSTM neural network excelling in the Discomfort and Risk Driving Recognition model, while the RF outperforms in the Efficiency and Aggressive Driving Recognition model, achieving high F-scores across all behavior labels. This confirms the feasibility of AI-driven methods in detecting driving behaviors, highlighting the potential of these technologies to enhance traffic safety and improve the driving experience through real-time monitoring and feedback. This research contributes to the evolving field of automotive AI, paving the way for the future development of intelligent driving assistance systems and proactive safety measures.

Chapter 5 are some conclusions of this thesis and prospects for future work.

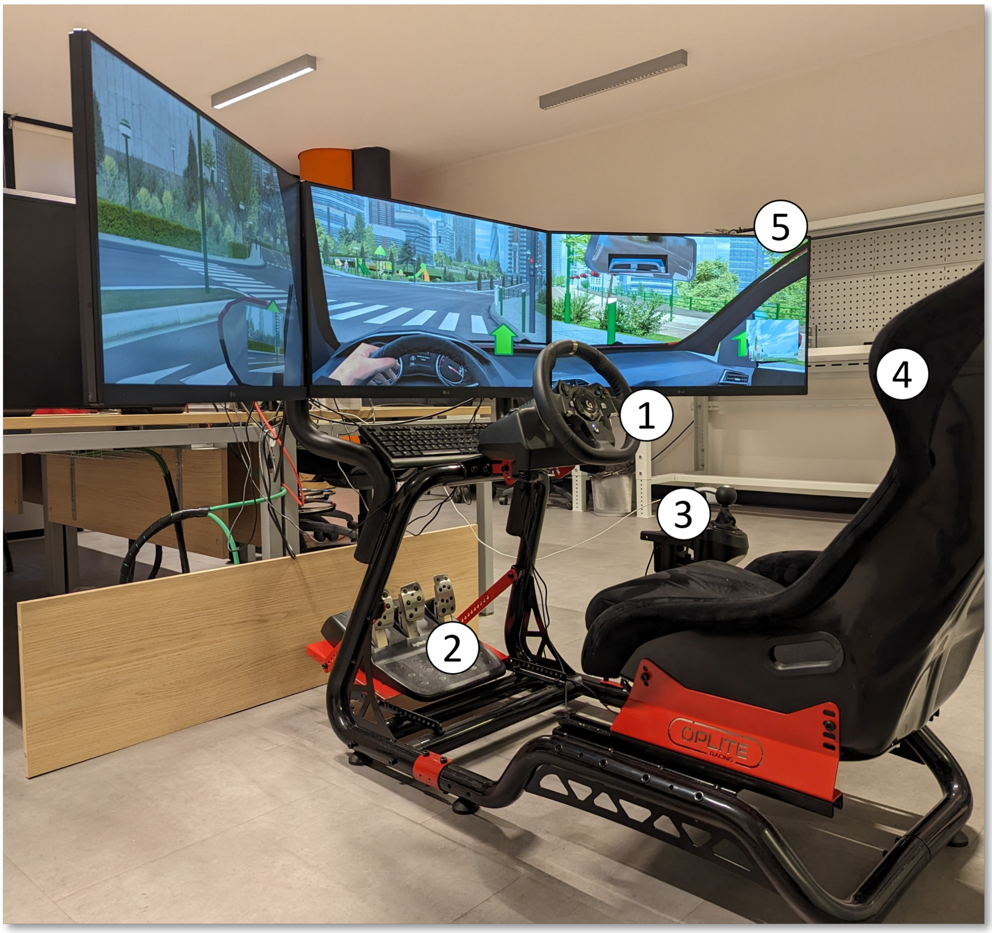


Figure 1.1: Driving simulator paired with SCANeR™Studio. 1: force feedback steering wheel, 2: pedals, 3: manual gearbox, 4: bucket seat, 5: high resolution screens.

Chapter 2

Virtual scene construction

2.1 Device introduction

As mentioned in Section 1.4, a driving simulator and SCANeR™Studio were utilized for vehicle data collection in a human-in-the-loop environment.

2.1.1 Hardware Setup

The driving simulator used in this experiment is equipped with a Logitech G920 steering wheel featuring vibration and force feedback. It also includes a clutch, accelerator, and brake pedals, a manual gearbox, and a bucket seat. Three LG high-definition monitors are used to replicate a realistic driving experience as closely as possible. Additionally, an NVIDIA GTX 3080Ti graphics card was selected to support the efficient operation of SCANeR™Studio.

2.1.2 Software

SCANeR™Studio is an advanced simulation and virtual reality software primarily aimed at research and development in the automotive industry. The software is widely used in the field of driving simulation to evaluate aspects such as driver behavior, vehicle performance, and traffic systems. SCANeR™Studio comes equipped with a comprehensive toolkit that can simulate complex traffic scenarios, various weather conditions, different vehicle types, and dynamic obstacles like pedestrians. It also supports integration with other engineering software such as MATLAB/Simulink, enabling users to conduct in-depth data analysis and system development. Whether used in car design, traffic research, or the development and validation of autonomous driving systems, SCANeR™Studio helps manufacturers and researchers evaluate and enhance vehicle performance and safety more effectively.

Its primary function is to offer users five distinct modes [29]:

1. **Vehicle mode**: For building mathematical representations of vehicles like cars and trucks. These representations rely on various parts such as suspension systems, braking mechanisms, lighting, tires, and wheels.
2. **Terrain mode**: For developing road systems that incorporate logical details like traffic signs, signal lights, and speed limit indicators, along with a three-dimensional visual environment.
3. **Scenario mode**: For designing training scenarios that utilize vehicles and terrain to enhance driver abilities, road infrastructure, and cockpit controls. This mode also allows for customizing situations, overseeing autonomous vehicles in the vicinity of the driver, ensuring compliance with instructions, and obtaining precise measurement data.
4. **Simulation mode**: For initiating a session and overseeing all simulator components. The simulator comprises both hardware and software components dedicated to sound, visualization, motion, and other functionalities.
5. **Analysis mode**: Utilized for analyzing exercise results, for instance: charts, 3D animations, and data tables.

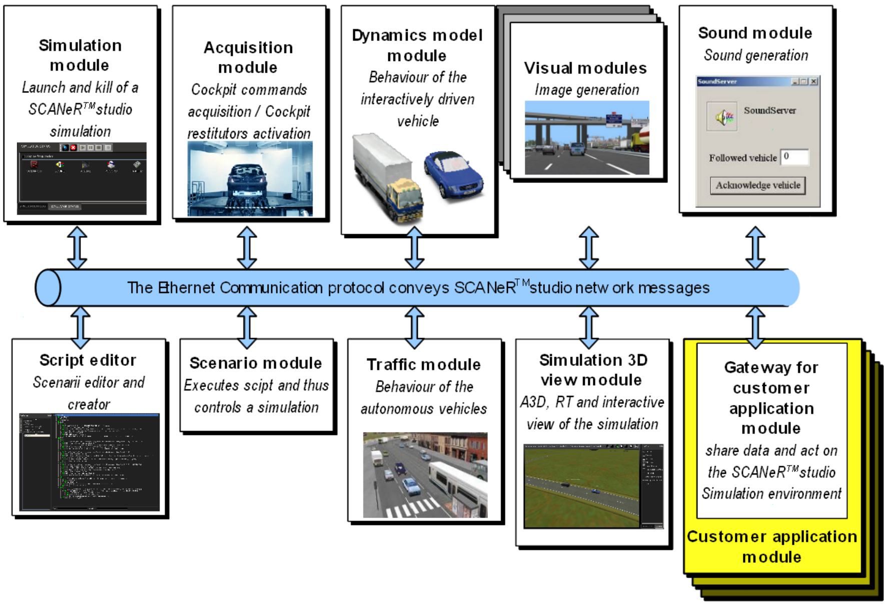

Simulation module
Launch and kill of a SCANeR™studio simulation
Acquisition module
Cockpit commands acquisition / Cockpit restitutors activation
Dynamics model module
Behaviour of the interactively driven vehicle
Visual modules
Image generation
Sound module
Sound generation
SoundServer
SoundServer
Followed vehicle 0
Acknowledge vehicle
The Ethernet Communication protocol conveys SCANeR™studio network messages
Script editor
Scenarii editor and creator
Scenario module
Executes scipt and thus controls a simulation
Traffic module
Behaviour of the autonomous vehicles
Simulation 3D view module
A3D, RT and interactive view of the simulation
Gateway for customer application module
share data and act on the SCANeR™studio Simulation environment
Customer application module

Figure 2.1: SCANeR™Studio modules

These software programs employ a shared communication protocol. Messages are exchanged among them via an Ethernet Network. Figure 2.1 illustrates the distributed architecture concept of SCANeR™Studio. Additionally, certain specific modules such as ACQUISITION (DriverHandler), MODELHANDLER (Dynamic model), MOTION, and SCANeR API can communicate with other key modules using shared memory, albeit requiring a higher exchange rate. Figure 2.2 illustrates some modules connected to shared memory.

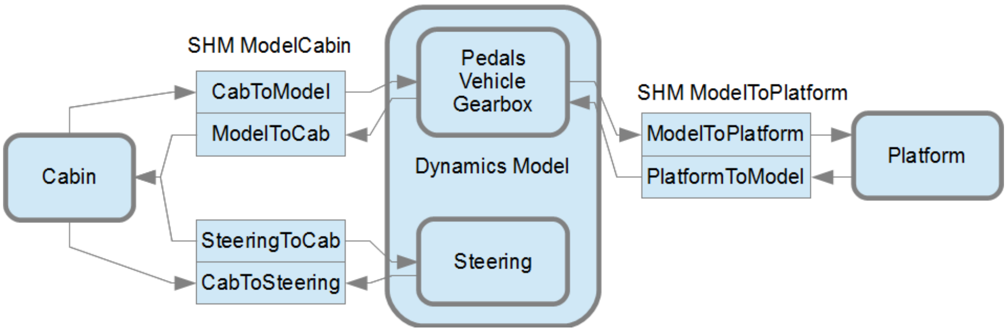


Figure 2.2: Modules connected to shared memory

2.2 Scenario construction

This section aims to provide a comprehensive overview of scenario construction for data collection, encompassing terrain, an array of objects such as vehicles and pedestrians, a set of parameters including initial conditions and recording settings, as well as storyboards essential for managing situations or events like accidents [29]. Crafting tailored scenarios facilitates the execution of diverse driving tasks.

2.2.1 Terrain construction

As mentioned before, selecting the appropriate terrain is the starting point for scenario construction, as it represents the simulated driving environment. In SCANeR™Studio, users can utilize preset terrain modules from the library or customize new terrains to meet experimental needs. The terrain mode is composed of the following three sub-modules:

1. **Environment Types**: Environment types encompass complex terrain scenarios such as highways, countryside areas, and even urban settings. This determines the driving experience of the simulator's driver.
2. **Road Infrastructure**: The road infrastructure module includes traffic lights, roundabouts, intersections, various levels of roads, obstacles, and bridges. The

presence of these infrastructures greatly ensures the realism and complexity of the driving experience in the simulated environment.

3. **3D Objects**: 3D objects such as road signs, decorative models (trees, buildings, billboards), can be set in the terrain. These objects enrich the details of the simulated environment, enhancing the driver's visual experience and making the scenario more realistic and vivid.

With these three modules, the realism and diversity of the simulated environment are guaranteed.

Recognizing the stark contrast in vehicle dynamics data distribution between urban and rural settings is paramount. Urban locales demand frequent deceleration, acceleration, and halting owing to congested thoroughfares, traffic signals, intersections, pedestrians, and the necessity for constant maneuvering. Conversely, rural areas witness diminished fluctuations in velocity, boasting wider turning radii and diminished vehicular and pedestrian traffic. Hence, it becomes imperative to execute tailored data gathering and analysis methodologies for disparate environments during the acquisition of vehicle dynamics data. By given this, the research will utilize the terrain modules named Lake City and Riviera from the default library for data collection.

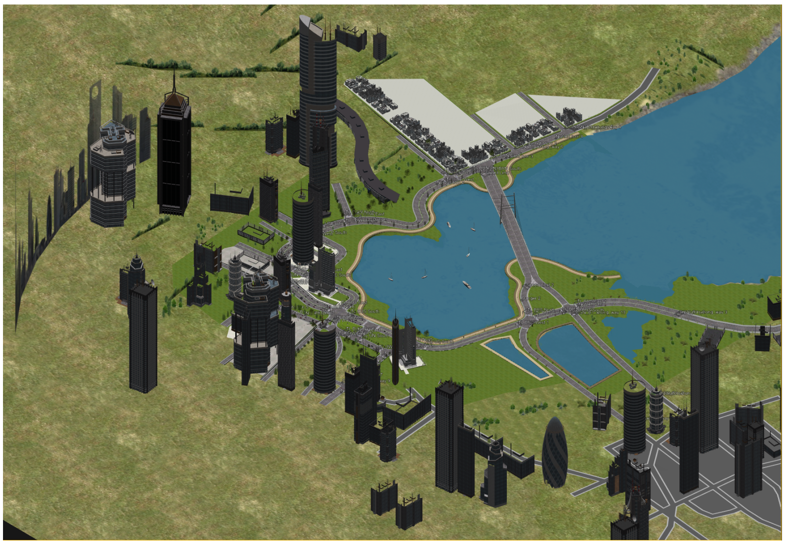


Figure 2.3: Lake City View

Lake City (see Fig. 2.4 and Fig. 2.3) module is described as a complex terrain (Right and Left Hand Traffic) where 3D were generated by computergraphic crew

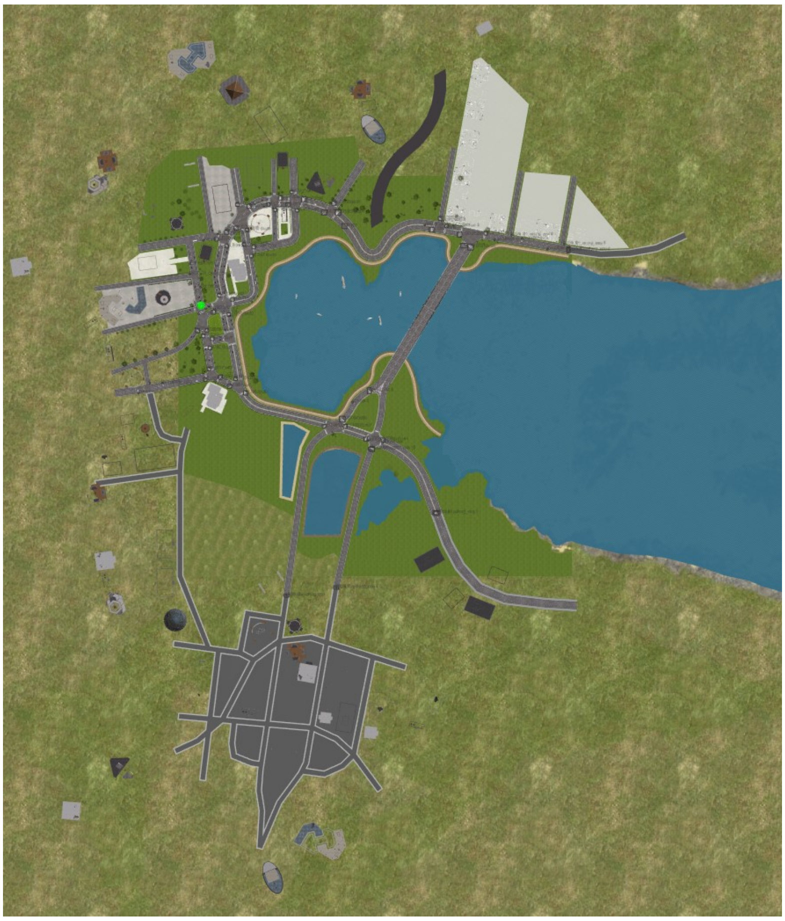


Figure 2.4: Lake City Top View

with specific options: occlusion map, bump map, specular map, specular. This terrain is dedicated to Advanced RenderingMode. This is a comprehensive terrain covering 3 environments: traffic jam in city center, the 500-meter-long sea-crossing bridge can be considered a highway and suburban areas with no traffic congestion. The terrain is configured with 47 traffic lights, and 28 intersections. The terrain contains physics modules for handling interactions between 3D objects (interactive vehicles, autonomous vehicles, pedestrians, bicycles, infrastructure objects, crash barriers) such as collisions and the vibrations caused by driving over rough surfaces. These physical behaviors are fed back to the driver in the form of vibrations through the simulator steering wheel.

Riviera (see Fig. 2.5 and Fig. 2.6) module is described as a simple terrain (Right and Left Hand Traffic with sea, seaside, tunnel, country, village (pedestrian crossings), mountains, forest elements. Several ambiances in a Mediterranean style.

Characteristics	Quantity
Road length [km]	6.1
Driving side	RHT LHT
Traffic lights	47
Barriers	0
Intersections	28
PHYSICS compliant	Yes

Table 2.1: Characteristics of the LakeCity Environment

This terrain is especially dedicated for demonstration or training purposes. Also, this module contains physics modules for handling interactions between 3D objects.

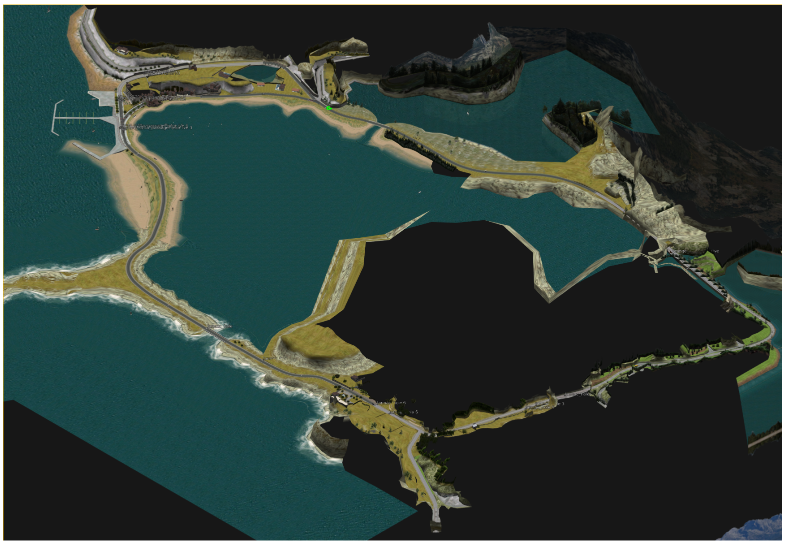


Figure 2.5: Riviera View

2.2.2 Resources selection

As mentioned at the beginning of this chapter, to ensure the realism and complexity of the simulation environment, the scene should include not only terrain but also 3D objects such as vehicles and pedestrians. In this section, we will briefly describe the driving vehicles and other vehicles in the scene.

In SCANeR$^{\text{TM}}$Studio, vehicle models include several subsets such as simple models,

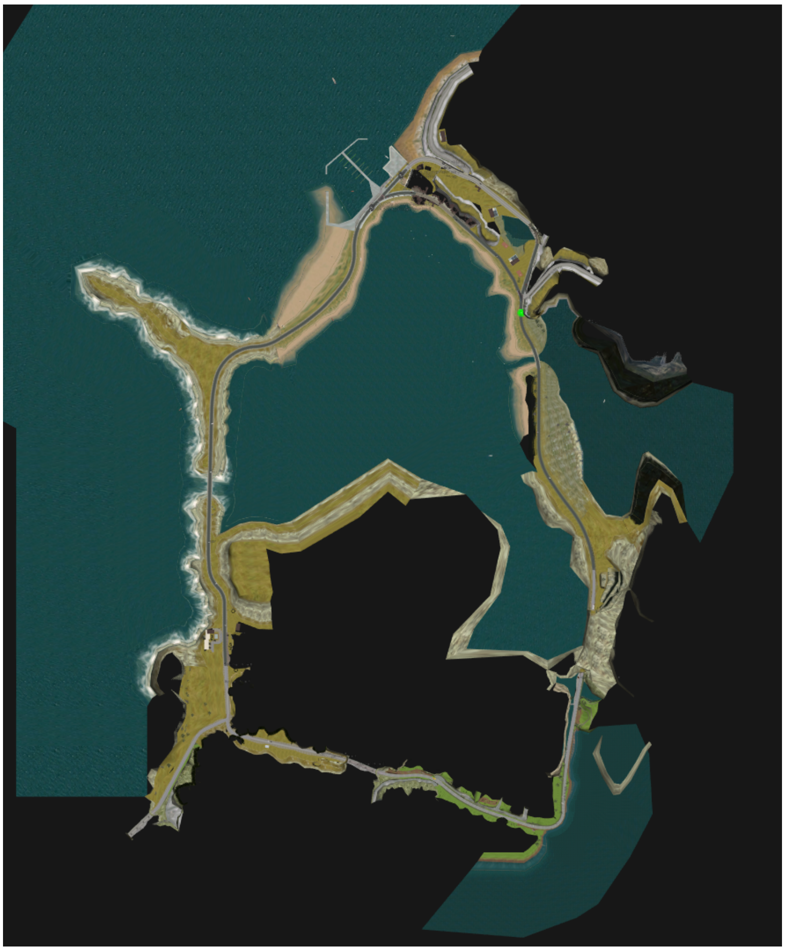


Figure 2.6: Riviera Top View

Characteristics	Quantity
Road length [km]	5.6
Driving side	RHT LHT
Traffic lights	0
Barriers	0
Intersections	12
PHYSICS compliant	Yes

Table 2.2: Characteristics of the Riviera Environment

Callas models, and CarSim models. The "simple" vehicle models lack detailed tire, suspension, and steering models and cannot adequately respond to subtle phenomena like rumble strips [29]. They do not include components models (physical models of engine, suspensions), do not have roll movement, and the lateral speed is set to zero, meaning they will not skid laterally. Additionally, they have some deficiencies in recognizing environmental obstacles.

The dynamics of the simple models are very basic [29]:

- Bi-axle (2 wheels per axle)
- Position of the vehicle is computed with 1 road picking
- Terrain following
- Engine
- Transmission
- Braking
- Steering

For autonomous vehicles present in the scene, they primarily rely on the traffic module to follow pre-planned routes. In such cases, simple models are often used for modeling these types of vehicles. In this study, all autonomous vehicles in the environment are composed of simple models from the default library. The scene includes Cars, Buses, Bicycles, Motorbikes, Trucks, Trailer assemblies. Their distribution is as shown in the Tab. 2.3. The driving behavior of autonomous vehicles is categorized into three types (Normal, Cautious, and Aggressive) as shown in the Tab. 2.4.

Vehicle type	**Vehicle distribution (%)**
Cars	65
Buses	5
Bicycles	10
Motorbikes	10
Trucks	5
Trailer assemblies	5

Table 2.3: Distribution of autonomous vehicles

For interactive vehicles that rely on hardware (driving simulator, keyboard and mouse) input, there are human driver controls. Experimental needs cannot be met using simple models. This often requires more complex dynamic models to simulate

Driving behavior	Distribution (%)
Normal	90
Cautious	5
Aggressive	5

Table 2.4: Driving behavior distribution of autonomous vehicles

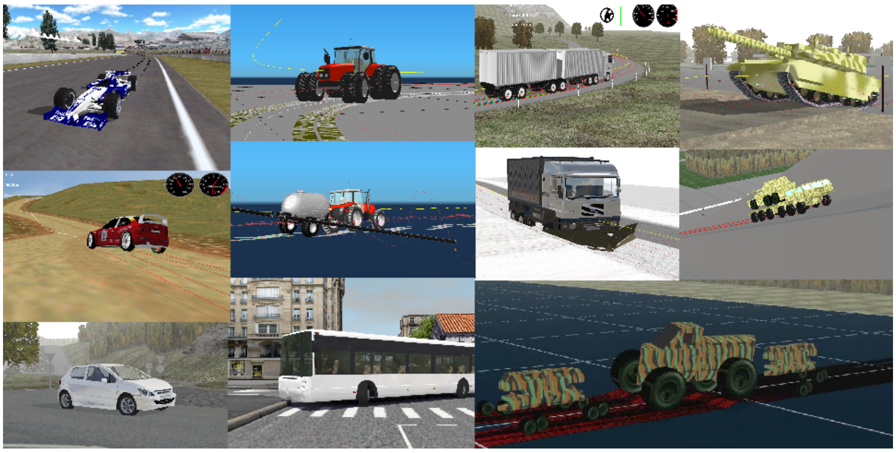


Figure 2.7: Callas Model

vehicle behavior in real driving. In this case the Callas model can be used.

Callas is the French name (Couplé A La Limite d'Adhérence au Sol) which represents dynamic model vehicles such as trucks, buses, cars and midgets, motorsport, machione, tractors, and military vehicles (such as tracked vehicles) (see Fig. 2.7) [29]. Unlike the simple model, it includes suspension and powertrain components, making it the most realistic and comprehensive dynamic model. Among them, suspension types can adopt all existing geometries: rigid axles, independent wheels, crawlers (tracks), hybrid drivetrains. Powertrains can be electric or combustion, with a full range of transmission options. Based on this, the study utilizes a Callas car model named 'SmallFamilyCarElectric' (see Fig. 2.8). Detailed parameters for this vehicle model is shown in the Tab. 2.5.

Figure 2.8: Small Family Car Electric

Engine	
Aspiration	Electric
Max Power (kW)	80
Electric Motor RPM (rpm)	10390
Max Torque (daN*m)	28
Transmission	
Transmission Type	Front Wheel Drive
Gearbox Technology Type	Auto
Front Gear Ratio Number	1
Rear Gear Ratio Number	/
Dimensions	
Length (mm)	4440
Width (mm)	1770
Height (mm)	1545
Weight (kg)	1523
Front Overhang (mm)	952
Rear Overhang (mm)	788
Wheelbase (mm)	2700
CoG Height from ground (mm)	520
Front Track / Rear Track (mm)	1540 / 1535

Ground Clearance (mm)	155
Driver Side	Left
Frame	
Steering wheel turn lock-to-lock	3
Steering diameter between sidewalks (m)	11
Steering diameter between walls (m)	11.8
Tires dimensions	195/65 15
Anti-Block Brake system	yes
Active yaw control	yes
Traction control	yes
Front suspension	Independent McPherson
Rear suspension	Twist Beam
Performances	
Max speed (km/h)	145
0-100 km/h (s)	25.6
Standing 400 m (s)	22.2
Standing 1000 m (s)	42.5
Specific suspension roll (°/G)	3.7
Specific suspension pitch (°/G)	3.97
Max Slope (%)	22
Max Banking (%)	36

Table 2.5: SmallFamilyCarElectric's technical document

In order to obtain the required vehicle driving data, the installation of sensors is necessary. SCANeR™Studio supports a variety of sensors such as cameras, lidars, and radars. Each sensor contains a variety of optional configurations. Users can use the default configurations in the library (such as Tesla sensor configurations) or customize sensor configurations according to experimental needs. In this research, a Long Range Radar Sensor (see Fig. 2.9 and Fig. 2.10) was equipped at the bumper position of the front of the car 0.5m from the ground to detect the straight-line distance to obstacles. Its technical specifications include a maximum detection range of 250 meters, a horizontal FOV of -30° to 30°, and a vertical FOV of -30° to 30°.

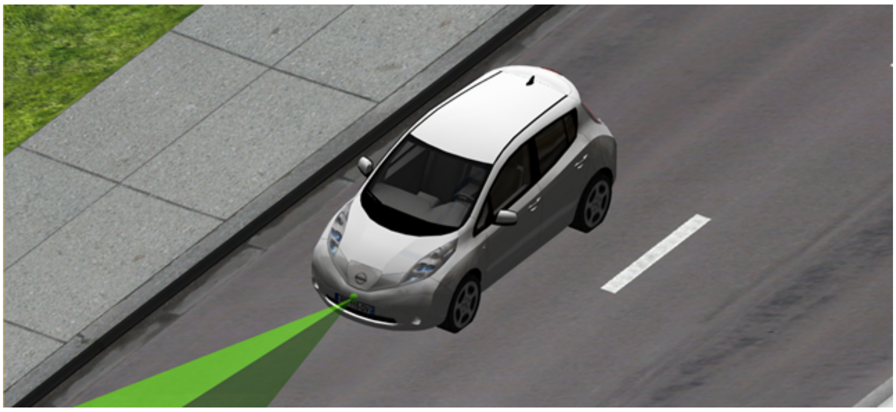


Figure 2.9: Zoomed view of radar sensor placement.

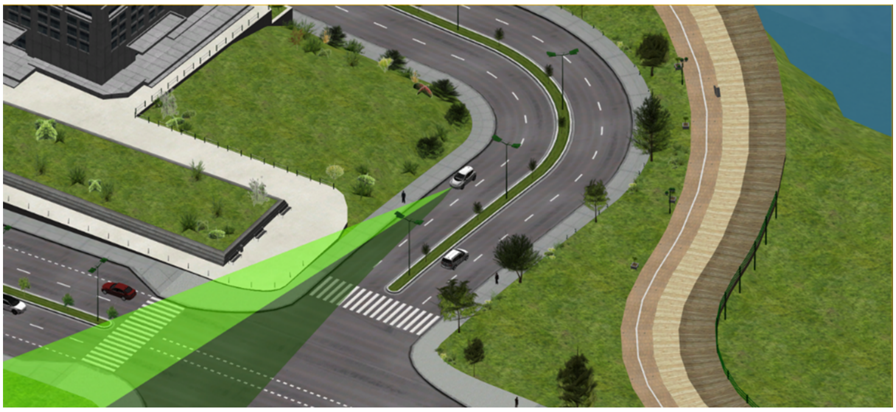


Figure 2.10: Long range radar sensor with maximum beam range of 250 m.

2.3 Simulation

After building the scenario, we can finally start running the simulation to collect vehicle dynamics data. Before you start driving a car, you need to start the following modules:

a) **Traffic**: Its role is to control the movement of other autonomous vehicles in the scene and changes in road signs (traffic lights).

b) **WalkerTraffic**: This module is used to manage pedestrian traffic; controlling the movement of pedestrians in the scene according to predefined scripts.

c) **Visual**: This module is dedicated to simulating the driver's perspective while driving; It conveys visual information about the environment, including vehicles, pedestrians, and road traffic conditions, to the driver through a display. This ensures the driver receives the most realistic driving feedback (see Figure 2.11).

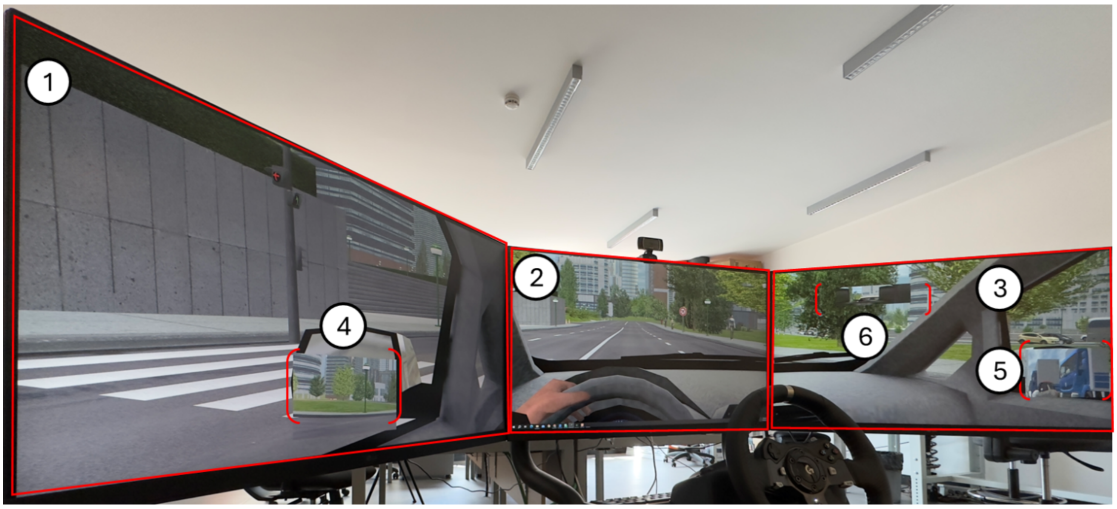


Figure 2.11: Visual module with 6 views. 1: Side view, 2: Front view, 3: side view, 4: left rear view, 5: right rear view, and 6: center rear view.

d) **Dashboard**: It completely replicates the dashboard of a real vehicle, including current vehicle speed, motor speed, and other information(see Figure 2.12).

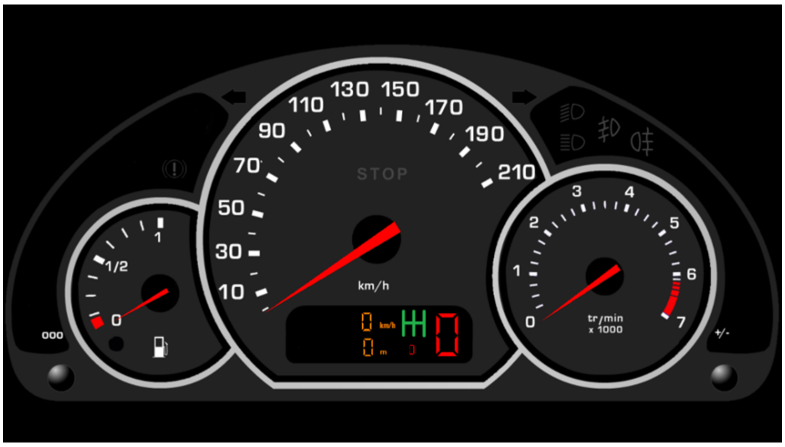


Figure 2.12: Close up of the interactive vehicle's dashboard.

e) **Sound**: This module is used to generate sounds from both the vehicle itself and the environment, such as the sound of the motor, the car horn, and the noise of wheels slipping. Through this module, the driver can gain a more comprehensive understanding of the current vehicle status and surrounding road conditions by combining audio cues with the visual module.

f) **Acquisition**: This module is dedicated to inputting signals from the driving simulator (such as the steering wheel and pedals) into the SCANeR™Studio vehicle, allowing the driver to control the vehicle in real-time.

g) **Physics**: As we mentioned in the previous section, this module is used to provide the driver with physical feedback on vehicle behavior (such as collisions, bumps caused by bad road conditions).

h) **ModelHandler**: This module is used to manage the interaction between the vehicle model and the road.

i) **Sensors**: It is dedicated to simulating the cameras, lidar, ultrasonic radar and other sensors mentioned in the previous section.

Chapter 3

Methodology

In the study described, we propose a methodology consisting of five main steps to extract and analyze driving behavior data from a simulated vehicle driving environment. This method is based on the SCANeR™studio software environment, aiming to comprehensively process and evaluate driving behavior through the following steps:

a) **Data Acquisition**: First, data is collected from the simulated vehicle driving environment of SCANeR™studio along with driving simulator.

b) **Data Preprocessing**: Next, the collected data is preprocessed, including cleaning, filtering, and formatting, to facilitate further analysis.

c) **Data Segmentation**: Subsequently, the data is segmented into "Elementary Driving Behaviors (EDB)" based on parameters such as vehicle speed and steering wheel angle.

d) **The dataset labeling**: The I-DBSCAN algorithm is used to analyze each EDB to identify aggressive driving behaviors. Based on specific threshold methods, identified aggressive driving behaviors are further refined into uncomfortable behaviors and potentially dangerous driving behaviors.

e) **Model Training**: Finally, the preprocessed data is fed into a LSTM neural network and RF for model training. Through this step, the main goal is to establish a machine-learning model capable of accurately predicting and analyzing driving behaviors. The following will provide a detailed explanation of each step.

3.1 Data Acquisition

In the simulation, drivers can control an interactive vehicle (the real human driver changes the vehicle's motion through physical pedals and steering wheel in the driving simulator), designated as a "small family electric car" to mimic authentic driving behaviors within the scenario.

Signals	**Unit**
Longitudinal Speed	m/s
Longitudinal Acceleration	m/s^2
Lateral Acceleration	m/s^2
Accelerator pedal	[-]
Brake force	N
Motor speed	rad/s
Motor efficiency	%
Steering wheel angle	rad
Steering wheel speed	rad/s
Jerk	m/s^3
Distance to collision	m
Time to collision	s

Table 3.1: Signals collected from vehicle

The data is collected from the interactive vehicle for three different drivers. The characteristic variables reported in Tab. 3.1 are highly instrumental in identifying aggressive and energy efficiency-related driving behavior. This study uses signals other than Motor efficiency to train the discomfort and risk driving recognition model. Uses motor efficiency instead of motor speed and other signals to train Efficiency and aggressive driving recognition model.

Figure 3.1 illustrates a glimpse of the small data that was collected. The longitudinal and lateral acceleration, as well as the vehicle velocity, are shown in the Figure 3.1. The efficiency map of the interactive vehicle is displayed in Figure 3.2. Using the current and torque of the motor, we can determine the operating efficiency of the electric motor at any given moment.

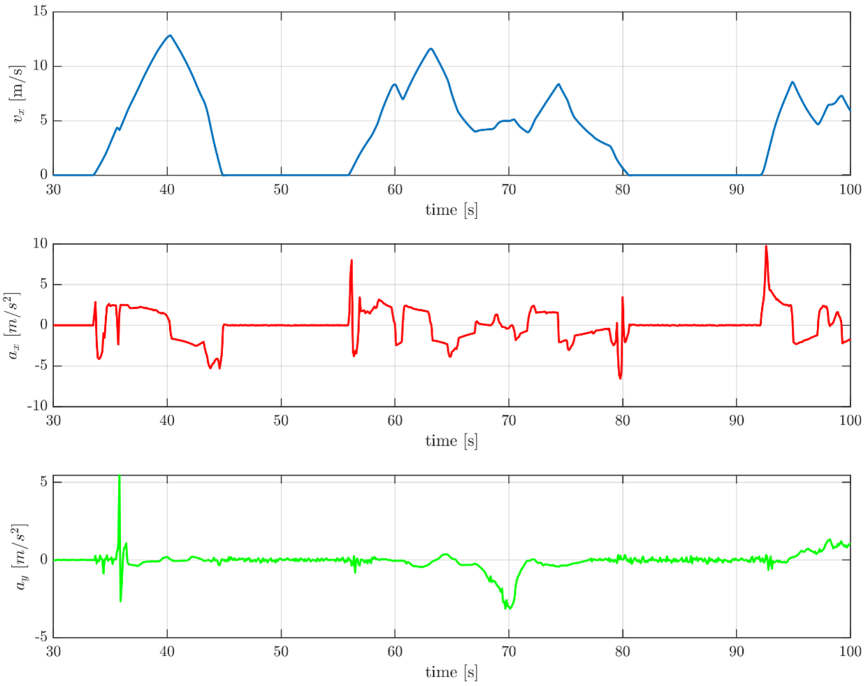


Figure 3.1: Data recordings by one of the drivers. 1st row: vehicle velocity; 2nd row: longitudinal acceleration; and 3rd row: lateral acceleration.

3.2 Data Preprocessing

After the data collection was completed, this study adopted a preprocessing step aimed at further analysis by dividing the time series data into equal length monitoring periods (MP) [30, 31]. Each MP was set to a length of 0.3 seconds, considering the data sampling rate of 100Hz, each MP contained 30 time steps (see Fig. 3.3). Moreover, there was a 0.1 second interval between MP, meaning there was an overlapping part between adjacent MP. Then, feature engineering was performed on the samples in each MP to extract statistical features (the statistical functions used are shown in Tab. 3.2). This method not only reduced the number of samples and computational costs but also enhanced the model's interpretability.

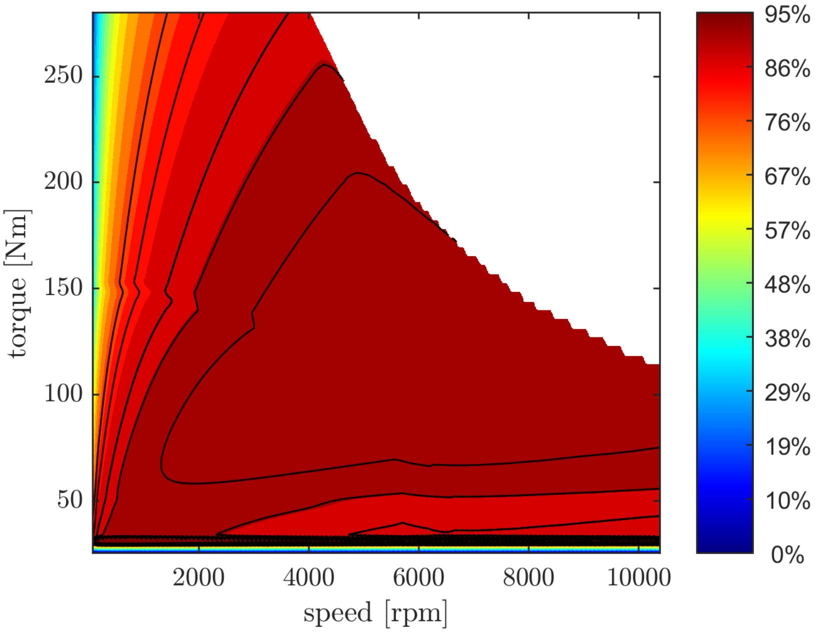


Figure 3.2: Efficiency map

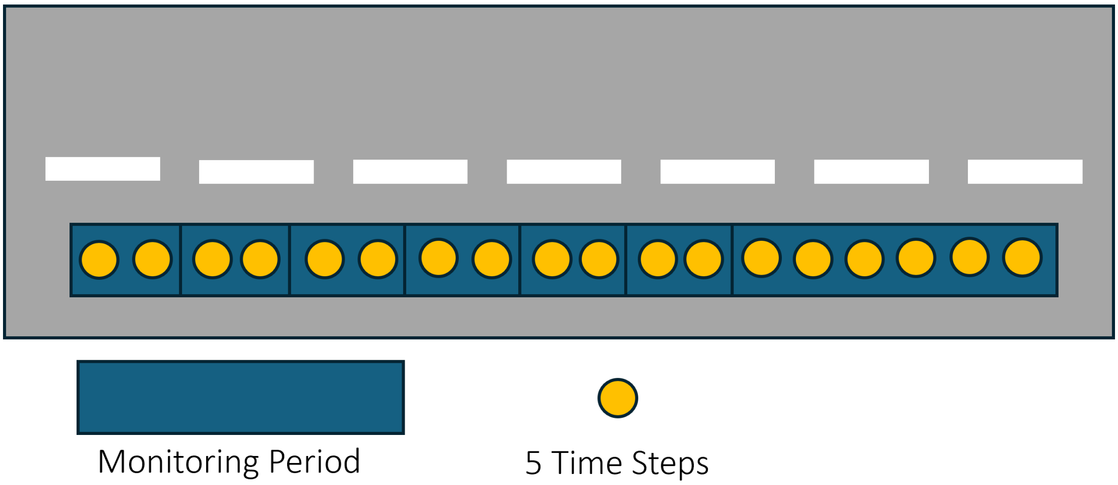


Figure 3.3: Monitoring period

Statistical Function	**Description**
Mean	Mean of a signal
Min	Minimum value of a signal
Max	Maximum value of a signal
Variance	Square of the standard deviation of a signal
STD	Standard deviation of a signal
RMS	Root mean square
Q_1	25th percentile
Q_2	50th percentile
Q_3	75th percentile
Peak amplitude	Difference between the maximum and minimum value of the signal

Table 3.2: statistical functions for feature engineering

3.3 Data Segmentation

3.3.1 Creation of elementary driving behaviors

To gain a deeper understanding of driving behaviors. For the "discomfort and risky driving recognition model", this study segmented the original dataset into 15 EDB, defining subsets of behaviors, including straight driving, slight left, and right turns, and left and right turns under low, medium, and high-speed conditions. A data profile describes each EDB. Even with a comparable data profile, aggressive behavior will stand out from the pattern of average behavior [31, 32].

In the "efficiency and aggressive driving recognition model", the study utilized the K-means clustering algorithm and threshold limits of steering angle and TTC [33] to divide the original dataset into 30 EDB, encompassing straight driving, slight left and right turns under low, medium, and high-speed conditions, with or without collision risk. Among these, 15 EDB with collision risk were directly labeled as aggressive driving behaviors. For the remaining 15 EDB, each one is described by a data profile, where aggressive driving behaviors exceeded the average behavior profile of each EDB.

3.3.2 K-means clustering

To achieve the subdivision of these behavior subsets, the study employed the K-means clustering algorithm and threshold limits of steering angle. The K-means clustering algorithm (see Alg. 1) is a widely used unsupervised learning algorithm designed to divide v observations into k clusters, with each observation assigned to the nearest cluster center (i.e., the point that minimizes the variance within the cluster). The mathematical expression of the K-means algorithm is as follows:

$$E = \arg\min_{s} \sum_{j=1}^{k} \sum_{v \in S_j} |v - \mu_j|^2 \tag{3.1}$$

In this context, S_j represents the cluster formed based on the value of k, E is the squared error. The mean vector of cluster S_j, also known as the centroid, is denoted by μ_j. The expression for the centroid is as follows:

$$\mu_j = \frac{1}{|S_j|} \sum_{v \in S_j} v \tag{3.2}$$

The centroids are initialized randomly, and each sample point is assigned to the centroid that minimizes the Euclidean distance. The centroids are then updated by calculating the mean of all points in each cluster [34].

3.4 The dataset labeling

3.4.1 Iterative DBSCAN

After successfully defining subsets of EDB, this study employs the DBSCAN algorithm for further analysis and separation of aggressive driving behaviors within each EDB. Before delving into this step in detail, it is necessary to provide a brief introduction to the DBSCAN algorithm. The pseudo-code of the DBSCAN algorithm is reported below.

DBSCAN identifies clusters based on the estimated density of data points. The core mechanism of this algorithm is the definition of a neighborhood radius (ϵ) and the criteria for a point to become a "core point" which is that there must be a specified number of m (minPts) in the range of ϵ. Algorithm divides the data set D into three types of points (Core Point, Border Point, Noise Point), and directly density-reachable (if q is in the ϵ-neighborhood of p, and p is the core point), density-reachable (if q is in the ϵ-neighborhood of p within the domain, and both p and q are core points), density-connected (if p and q are both non-core points, and p and q are in the same cluster class) three connection rules form a

Algorithm 1 K-means clustering pseudocode

```
Initialise Cluster Centers
for each iteration $l$ do
    Compute $r_{nk}$:
    for each data point $x_n$ do
        Assign each data point to a cluster:
        for each cluster $k$ do
            if $k == \arg\min ||x_n - \mu_k^{l-1}||$ then
                $r_{nk} = 1$
            else
                $r_{nk} = 0$
            end if
        end for
    end for
    for each cluster $k$ do
        Update cluster centers as the mean of each cluster:
        $\mu_k^l = \frac{\sum r_{nk} x_n}{\sum r_{nk}}$
    end for
end for
```

cluster class [35]. Because of these characteristics, DBSCAN particularly suitable for identifying clusters in different density regions and effectively separating noise from cluster regions.

Core Point: for a dataset D, a sample p is considered a core point if the ϵ-neighborhood of p contains at least MinPts samples. The mathematical expression for this is as follows:

$$\begin{gathered} N_\varepsilon(p) \geq \text{ MinPts} \\ N_\varepsilon(p) = \{q \in D \mid \operatorname{dist}(p,q) \leq \varepsilon\} \end{gathered} \tag{3.3}$$

$N_\epsilon(p)$ represents the number of points in the ϵ-neighborhood of point p.

Border Point: for a sample h that is not a core point, if h is within the ϵ-neighborhood of any core point p, then sample h is referred to as a border point. The mathematical expression for this is as follows:

$$h \in N_\varepsilon(p) \tag{3.4}$$

Noise Point: for a sample n that is not a core point, if n is not within the ϵ-neighborhood of any core point p, then sample n is referred to as a noise point. The mathematical expression for this is as follows:

$$n \notin N_\varepsilon(p) \tag{3.5}$$

Algorithm 2 DBSCAN pseudocode

```
Input: DB: Database
Input: $\epsilon$: Radius
Input: $minPts$: Density threshold
Input: $dist$: Distance function
Data: $label$ Point labels, initially undefined
for each point $p$ in database DB do do
    if $label(p) \neq$ undefined then
        then continue
    end if
    Neighbors $N \leftarrow RangeQuery(\text{DB}, dist, p, \epsilon)$
    if $|N| < minPts$ then
        $label(p) \leftarrow$ Noise
        continue
    end if
    $c \leftarrow$ next cluster label
    $label(p) \leftarrow c$
    Seed set $S \leftarrow N \setminus \{p\}$
    for each $q$ in $S$ do do
        if $label(q) =$ Noise then
            $label(q) \leftarrow c$
        end if
        if $label(q) \neq$ undefined then
            then continue
        end if
        Neighbors $N \leftarrow RangeQuery(\text{DB}, dist, q, \epsilon)$
        $label(q) \leftarrow c$
        if $|N| \geq minPts$ then
            then continue
        end if
        $S \leftarrow S \cup N$
    end for
end for
```

Figure 3.4 illustrates the theory of DBSCAN clustering. The violet points (A) are classified as core points. Points B are designated border points since they are located within a certain distance from a core point. C are noise points because they are neither in the epsilon range nor core points.

The choice of the minPts parameter is generally two times the number of

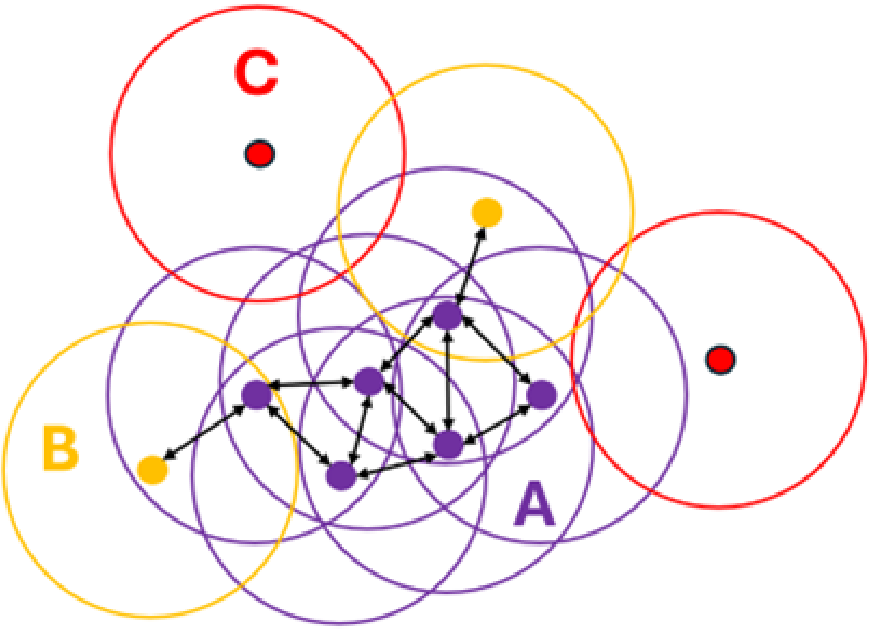


Figure 3.4: DBSCAN cluster principle.

significant features obtained after performing Principal Component Analysis (PCA) on the original data [36].

$$minPts = 2 * Feature \tag{3.6}$$

The selection of ϵ involves using the Elbow Method [37], which consists of determining the distance from every point in the dataset to its K$^{\text{th}}$ nearest neighbor (where K$^{\text{th}}$ is the minPts value). These distances are then arranged in descending order and plotted, resulting in what is known as a sorted k-distance plot. In this research, I-DBSCAN is used to separate anomalous driving behaviors from normal behaviors within each EDB. When dealing with high-dimensional datasets, the performance of clustering algorithms often deteriorates as the number of dimensions increases, a phenomenon known as the "curse of dimensionality". The introduction of PCA reduces the dimension of the input data set [38].

The steps of I-DBSCAN are as follows [30, 31]:

1. Determine the parameters necessary for DBSCAN: minPts and ϵ.

2. Set normPercent as the minimum percentage required for normal driving clusters. Execute DBSCAN and verify that the percentage of normal driving exceeds normPercent.

3. Segment the clusters into normal clusters, anomalous clusters, and noise.

4. If the proportion of normal driving observations exceeds normPercent, the procedure is finished. Otherwise, run another iteration on the normal driving cluster.

3.4.2 Threshold limits related to comfort, safety, and efficiency

To get labeled data to train the "discomfort and risk driving recognition model". Aggressive driving behaviors were further divided into three types: behaviors that affect passenger comfort, behaviors that pose a potential danger and aggressive driving behavior that does not include the above two characteristics. According to the study [39], by analyzing Jerk values (i.e., the rate of change of vehicle acceleration), behaviors with Jerk values greater than 1.07 m/s^3 or less than -1.47 m/s^3 can be classified as a aggressive driving behaviors that affect passenger comfort. Such behaviors, often characterized by the driver's frequent acceleration or deceleration, may cause passengers to experience motion sickness. Moreover, aggressive driving behaviors in everyday driving are also associated with a higher risk of collision, primarily because these drivers have shorter TTC compared to drivers engaged in normal driving [25]. TTC is defined as the time required for two vehicles to collide if they continue at their current relative speed and direction unchanged. The mathematical expression for TTC is as follows [33]:

$$TTC = \frac{X_{i-1}(t) - X_i(t) - l_l}{\dot{X}_i(t) - \dot{X}_{i-1}(t)} \quad \forall X_i(t) > X_{i-1}(t) \tag{3.7}$$

where, X_{i-1}, $\dot{X}_{i-1}$ represents the position and speed of the front of the leading vehicle, l_i represents the length of the leading vehicle, $X_i(t)$, $\dot{X}_i(t)$ represents the position and speed of the front edge of the trailing vehicle.

In the research by W. Zhao [40], the threshold for the TTC, denoted as $TTC_{(s)}$, is defined as 5 seconds. This means that driving behavior is considered potentially dangerous and likely to result in a collision if $TTC < TTC_{(s)}$. Based on this threshold, the aggressive driving dataset can be filtered to instances where TTC is less than 5 seconds and label these instances as aggressive driving behaviors with a risk of collision. The purpose of this classification is to enable timely warnings or interventions for drivers when potentially dangerous aggressive driving behaviors are detected, thereby effectively preventing possible accidents.

For the "efficiency and aggressive driving recognition model", normal driving behaviors were further divided into two clusters: driving behaviors with high efficiency and driving behaviors with low efficiency by using 25th percentile of motor efficiency in the normal driving cluster as the threshold. Values below this threshold are classified as low-efficiency driving behavior and those above it as high-efficiency driving behavior.

3.5 Model Training

In this study, after obtaining a labeled dataset, bayesian optimization-based LSTM neural network and RF algorithms are used to predict driving behavior.

3.5.1 Bayesian optimization

Bayesian optimization-based LSTM is particularly suited for optimizing hyperparameters with additional computational costs when compared to standard LSTM. It relies on Bayesian inference principles by constructing a probabilistic model of the objective function and using it to select hyperparameters. It primarily consists of two components: Gaussian Process and acquisition functions. This approach is advantageous for optimizing complex functions, especially when samples are scarce. Bayesian optimization approach is detailed in [17].

3.5.2 Long short-term memory neural network

LSTM is an enhanced type of recurrent neural network that introduces gates such as forget gate f_t, input gate i_t, and output gate o_t (see Fig. 3.5). The forget gate allows irrelevant information to be ignored, and the input gate and output gate control the access to information.

These mechanisms address the issues of vanishing and exploding gradients in long sequence data training, effectively modeling the long-term dependencies in time series data. The Hidden state z_t and cell state c_t is given by:

$$\begin{aligned} z_t &= o_t \odot \tanh\left(c_t\right) \\ c_t &= f_t \odot c_{t-1} + i_t \odot \tanh\left(W_{c1} z_{t-1} + W_{c2} x_t \ + W_{c3} + b_c\right) \end{aligned} \tag{3.8}$$

The current hidden state z_t is updated by the Hadamard product of the current cell state and the output gate. The current hidden state c_t is determined by the three gates, the previous hidden state z_{t-1}, the weight matrix W, the bias term b and the static metadata s, x_t is input vector for current time step t and y_t present future values of a target [41]. The mathematical expression of input gate is:

$$i_t = \sigma\left(W_{i1} z_{t-1} + W_{i2} y_t + W_{i3} x_t + W_{i4} s + b_i\right) \tag{3.9}$$

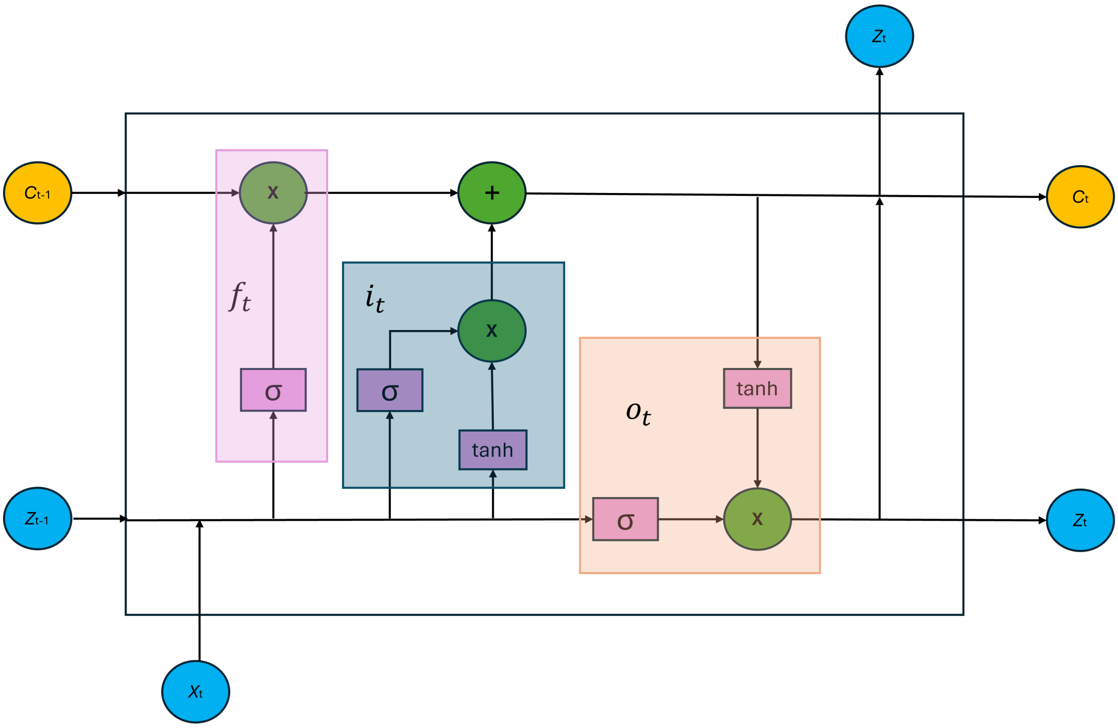


Figure 3.5: LSTM structure.

σ is the sigmoid activation function, which limits the output value to between 0 and 1. In this form, the amount of new input information can be controlled to enter the unit state c_t. The mathematical expression of forget Gate is:

$$f_t = \sigma\left(W_{f1} z_{t-1} + W_{f2} y_t + W_{f3} x_t + W_{f4} s + b_f\right) \tag{3.10}$$

It determines which information in unit state c_{t-1} should be retained or forgotten. Output Gate - determines how much information about the unit state c_t will be used to calculate the hidden state z_t at the current time step. Its mathematical expression is:

$$o_t = \sigma\left(W_{o1} z_{t-1} + W_{o2} y_t + W_{o3} x_t + W_{o4} s\ + b_o\right) \tag{3.11}$$

Through this gating mechanism, the LSTM network can learn and retain long-term dependencies, resulting in better performance when processing long sequence data.

3.5.3 Random forest

RF is an ensemble algorithm proposed by Leo Breiman that constructs multiple decision trees and combines their results to perform classification. With the help

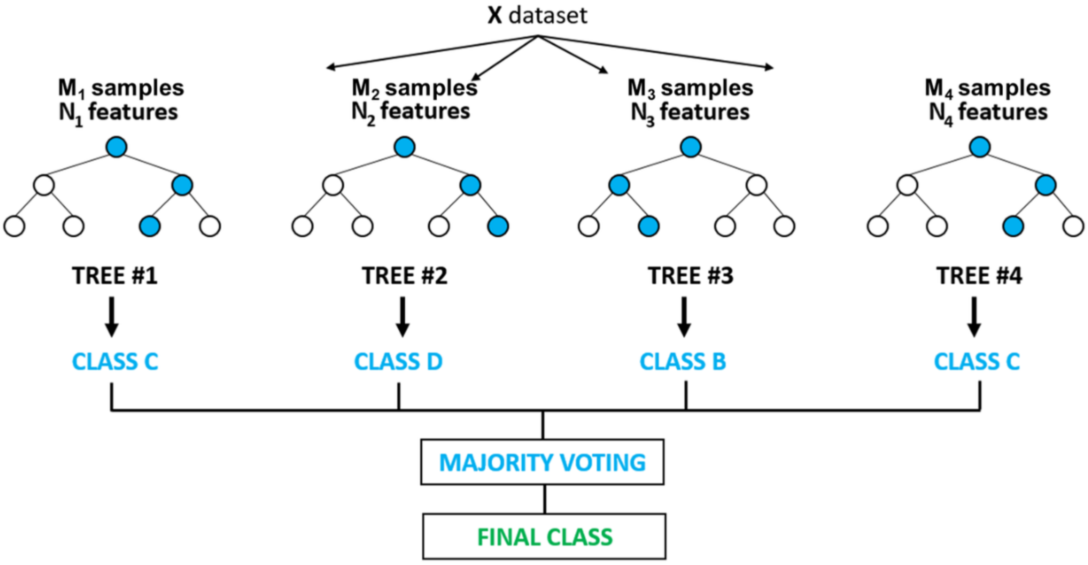


Figure 3.6: RF structure.

of decision trees, the algorithm introduces randomness, thereby improving overall predictive performance and generalization ability. When the algorithm receives an input vector x (sensor data in our case), it constructs H regression trees and averages outputs. If H number of trees $T(x)_1^H$ are created, the regression predictor of RF is[42, 43]:

$$f_{RF}^{H}(x) = \frac{1}{H}\sum_{h=1}^{H} T_h(x) \tag{3.12}$$

To reduce the correlation among trees, RF creates training data sets by randomly sampling multiple subsets from the original data set. The process of training a model on each subset to increase the diversity of the tree is called "Bagging". It is worth noting that each subset is generated by "sampling with replacement", which means that a data point may be selected multiple times, and some data points may never be selected. Ultimately, the results of these models are combined through voting or averaging (see Fig. 3.6 [44]). This not only improves the accuracy of the models but also enhances their stability, allowing them to maintain good performance even when facing small changes in input data. Additionally, when constructing a decision tree, the algorithm selects the best feature from a subset of input features as the splitting point. This characteristic helps to reduce the correlation between different trees [42].

Chapter 4

Performance evaluation and Discussion

4.1 Discomfort and risk driving recognition model

The 933,417 samples collected were consolidated into MP with 0.1 second intervals and 0.3 second lengths, through which the raw data were transformed into 46,678 MP containing 30 samples per group. Subsequently, we followed the second computational process (Data processing) mentioned in the methodology, which resulted in the statistical characteristics presented in Tab. 3.2. Finally, the raw data were transformed into a new dataset of 46,678 MP containing 110 features each for further computations.

Using the k-means clustering algorithm with parameter k equal to 3, the 46,678 MP are divided into three clusters: low, medium, and high speed based on the average speed (see Fig. 4.1). Based on this, we further segmented the data of these three clusters using threshold limits of steering angle. Specifically, data with steering angles less than 10 degrees (0.17 radians) were classified as straight ahead; those with steering angles between 10 and 45 degrees (0.17-0.79 radians) were identified as slight steering; and those with steering angles greater than 45 degrees (0.79 radians) were categorized as significant steering. Positive and negative values of the steering angle indicated that the car was turning to the right or to the left, respectively. In this way, we ended up with an EDB of 15 different combinations of speed and steering maneuvers.

Before applying the I-DBSCAN algorithm to each EDB, we first screened the features of the raw data by PCA method to identify the most critical feature components. Our analysis showed that only 17 key feature components were needed

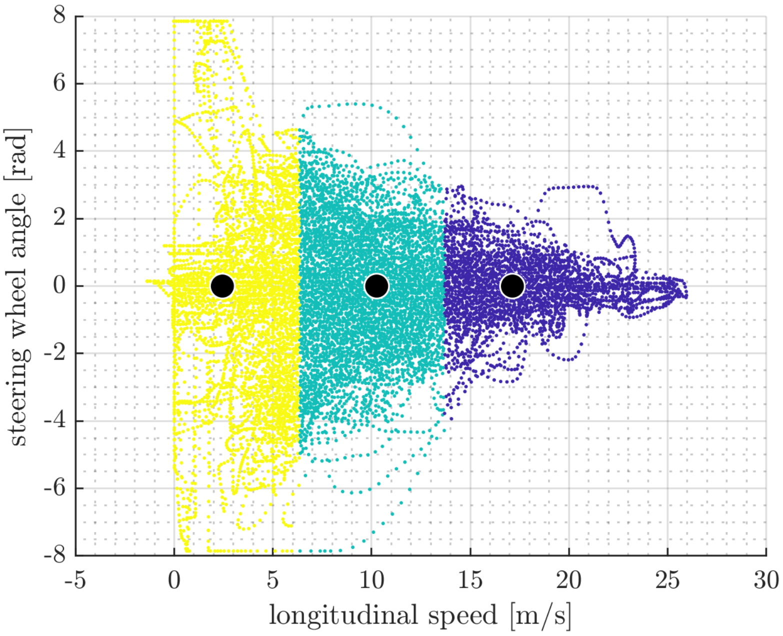


Figure 4.1: K-means clustering of speed. Black dot: centroid, yellow: low-speed data point, cyan: medium-speed data points, purple: high-speed data points.

to explain 90% of the variance in the data. Figure 4.2 illustrates the outcome of PCA which helps in selecting the key components/features. After processing the input data, two key parameters for the DBSCAN algorithm were defined: minPts and radius (ϵ). The minPts was set to twice the number of principal component variables, i.e., 34. In order to select appropriate ϵ values, we plotted elbow diagrams for all 15 EDB (see Fig. 4.3), and we eventually chose a range of ϵ values centered on the range between 3.2 and 14.2. Subsequently, we execute the DBSCAN algorithm at least three times for each EDB, after finding clusters covering 90% (normPercent = 90%) of the EDB data. The purpose is to ensure that the algorithm stops after finding a cluster containing 90% of the EDB data. The results show that we identified 43,761 normal driving behaviors marked as 0, and 2,917 aggressive driving behaviors marked as 1.

To explore the differences between normal and aggressive driving behaviors, we extracted and compared some input features (average, mean of maximum and minimum values) of EDB (see Tab. 4.1). The analysis revealed that mean maximum longitudinal acceleration and mean minimum longitudinal acceleration differed significantly between the two types of driving behaviors, which highlights the effectiveness of the clustering algorithm. Values in parentheses indicate the

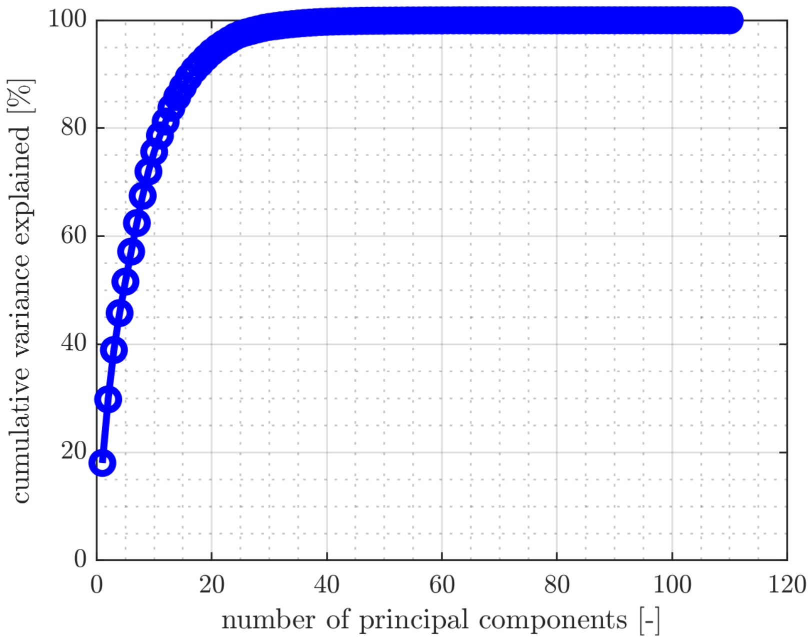


Figure 4.2: Principal component analysis.

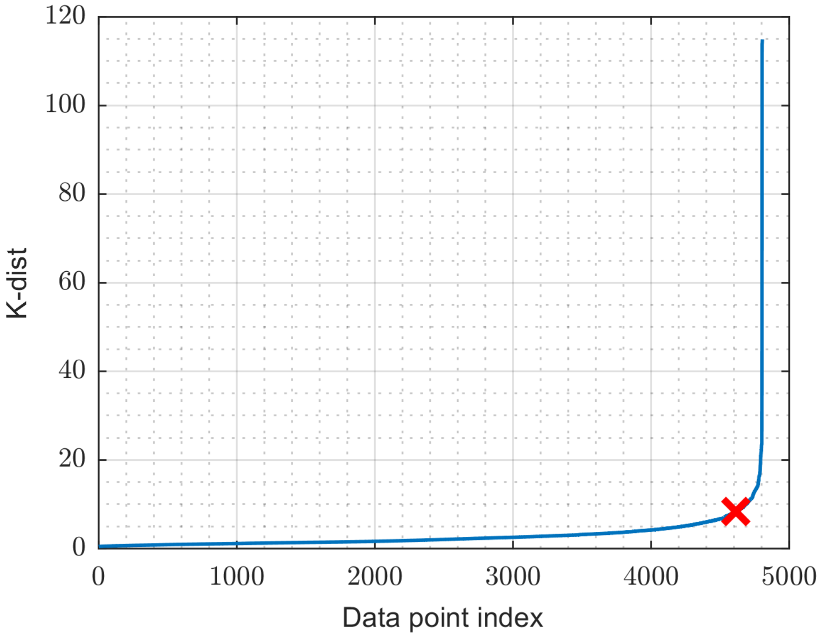


Figure 4.3: K-dist plot. Red cross: chosen ϵ.

standard deviation. We observe that aggressive driving behaviors have a higher degree of dispersion compared to normal driving behaviors, which show a higher degree of consistency on a normal distribution. Figures 4.4 and 4.5 show that aggressive driving behavior is clustered with normal driving behavior to some extent based on the difference of longitudinal and lateral acceleration. Some sample points in the figures that do not have significant differentiation may be clustered based on other kinetic features. Based on the thresholds of jerk i.e., [1.07, -1.47] m/s^3 and TTC (5 seconds), we segmented the aggressive driving data obtained, distinguishing two types of aggressive driving behaviors: one that affects driving comfort and the other with a potential crash risk. Based on this classification method, we succeeded in classifying the data into four categories: 43,761 samples representing

Feature	**Units**	**Normal**	**Aggressive**
Average Speed	(m/s)	10.67 (6.19)	10.70 (6.71)
Average Longitudinal Acceleration	(m/s^2)	0.14 (1.26)	-2.21 (2.82)
Average Lateral Acceleration	(m/s^2)	-0.13 (1.79)	-0.24 (2.38)
Average Jerk	(m/s^3)	0.01 (1.19)	-0.18 (3.21)
Max Speed	(m/s)	10.82 (6.19)	11.09 (6.70)
Max Longitudinal Acceleration	(m/s^2)	0.32 (1.27)	-1.12 (3.34)
Max Lateral Acceleration	(m/s^2)	0.02 (1.81)	0.71 (3.51)
Max Jerk	(m/s^3)	0.21 (0.91)	0.42 (3.26)
Min Speed	(m/s)	10.53 (6.19)	10.31 (6.71)
Min Longitudinal Acceleration	(m/s^2)	-0.05 (1.36)	-3.4 (3.83)
Min Lateral Acceleration	(m/s^2)	-0.29 (1.83)	-1.18 (2.47)
Min Jerk	(m/s^3)	-0.19 (1.17)	-0.76 (3.15)

Table 4.1: Statistical feature of two labels. Values in round bracket are standard deviation.

normal driving behavior (labeled 0), 1,062 samples representing general aggressive driving behavior (labeled 1), 351 samples aggressive driving behavior with potential collision risk (labeled 2), and 1,504 samples aggressive driving behavior involving comfort (labeled 3).

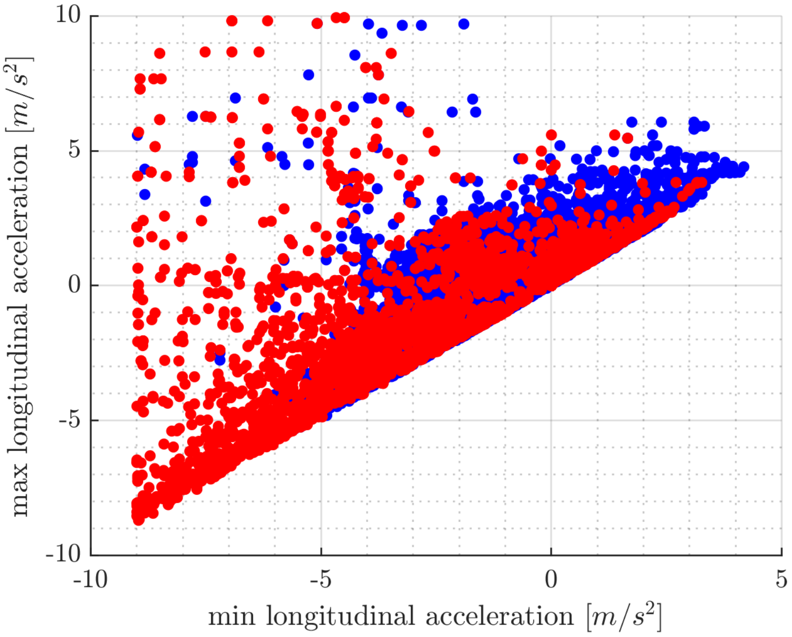


Figure 4.4: Max longitudinal acceleration vs Min longitudinal acceleration. Red dots: aggressive behavior, blue dots: normal behavior.

Taking the obtained sample features as inputs and the labels as outputs, 9237 samples were selected as the test set to verify the performance of the model. Processing the data using a sliding time window of 100 time steps and an overlap rate of 99%, and input them into a LSTM model for training. In order to select the optimal initial parameter values, we used a Bayesian optimization method to tune and optimize several key parameters including dropout rate, initial learn rate, and L2 Regularization coefficient. By using Bayesian optimization, we are able to effectively search through the parameter space and find those parameter values that maximize the model performance. The finalized initial parameters are shown in Tab. 4.2, which provide a strong starting point for our LSTM model to be able to demonstrate better learning efficiency and prediction accuracy when dealing with aggressive driving behavior recognition tasks.

To validate the effectiveness of the results achieved by the training algorithm,

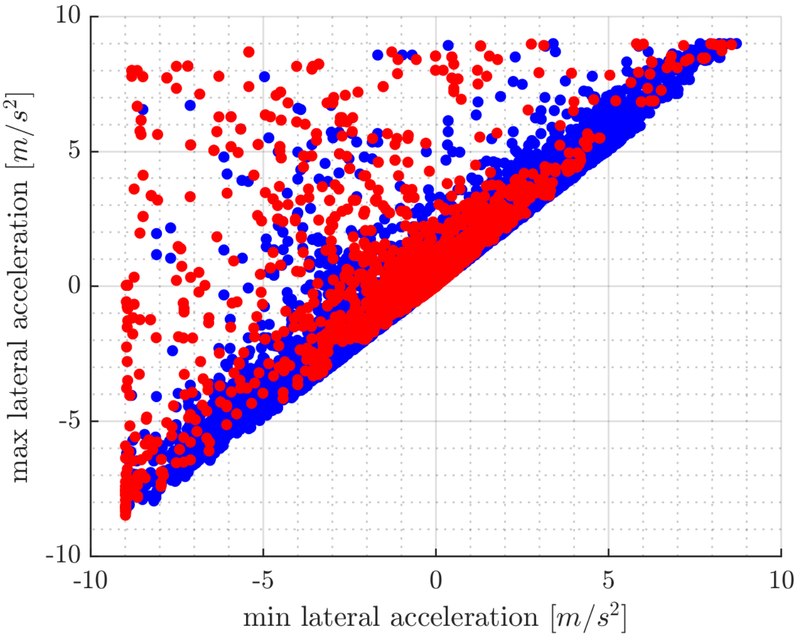


Figure 4.5: Max lateral acceleration vs Min lateral acceleration. Red dots: aggressive behavior, bule dots: normal behavior.

Hyperparameter	**Value**
LSTM number of layer	2
Number of hidden units	128
Maximum epochs	75
Batch size	64
Dropout rate	0.3
Initial learn rate	3.9463e-4
L2 Regularization	1.012e-4
Loss function	Cross-Entropy

Table 4.2: Hyperparameters of LSTM network.

we paid special attention to the optimal initial hyperparameters determined by the Bayesian optimization algorithm in the sixth iteration (see Fig. 4.6).

Figure 4.7 illustrates the trend of the training loss compared to the validation loss. It is observed that the loss value decreases rapidly at the beginning of training, indicating that the model quickly learns the features of the data in the early stages. The model tends to converge after approximately 4000 iterations.

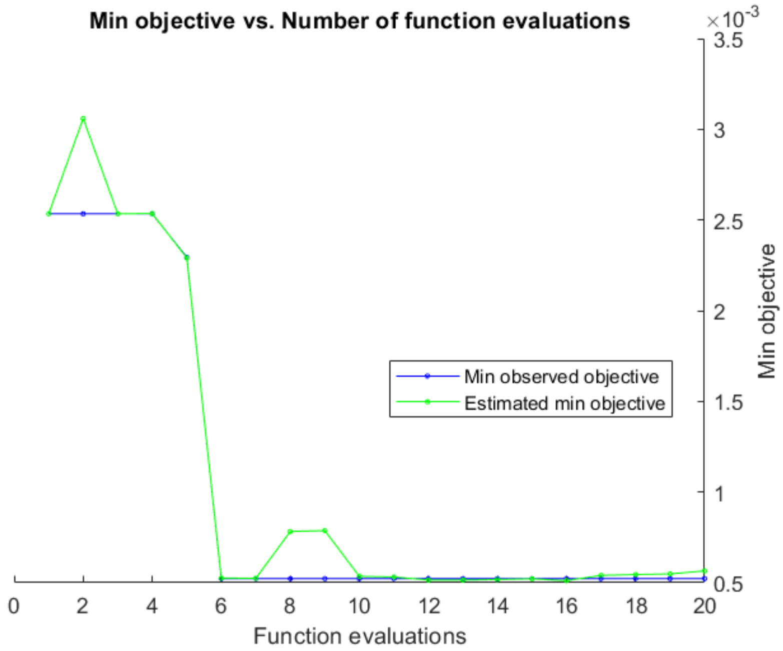


Figure 4.6: Bayesian Optimization Loss Curve.

Notably, although the blue solid line representing the training loss in the figure fluctuates (This is because we used a smaller batch size. When the batch size is small, the distribution and characteristics of the data in each batch can be very different. This variability causes the loss to fluctuate. However, this can also help the model escape from local minima and avoid overfitting. This randomness usually generalizes better to unseen data.), the overall trend is consistent with the test loss, which is represented by the red dashed line. This implies that the model does not suffer from overfitting and maintains good generalization ability.

The confusion matrix is a key tool for evaluating the performance of neural networks and is reported in Fig. 4.8. As an n x n square matrix (where n denotes the total number of categories), the confusion matrix provides a detailed comparison of the model's classification predictions with the true labels (clustered by I-DBSCAN algorithm). Elements on the diagonal of the matrix indicate the proportion of correctly categorized samples, while off-diagonal elements reflect misclassification. This detailed demonstration of categorization effectiveness not only helps us assess the overall accuracy of the model, but also allows us to identify and improve the model's under-performance on specific categories.

The algorithm predicts the data samples in four classes: *aggressive, normal*

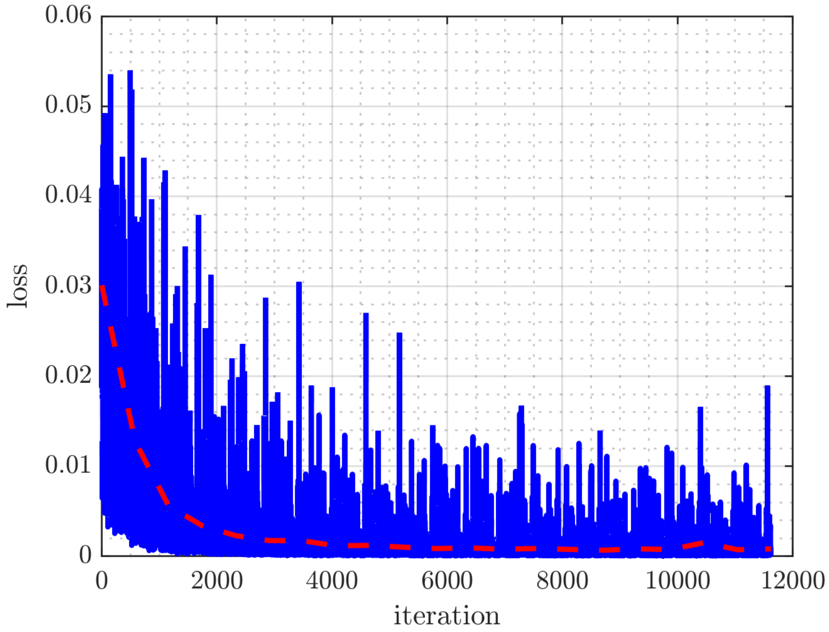
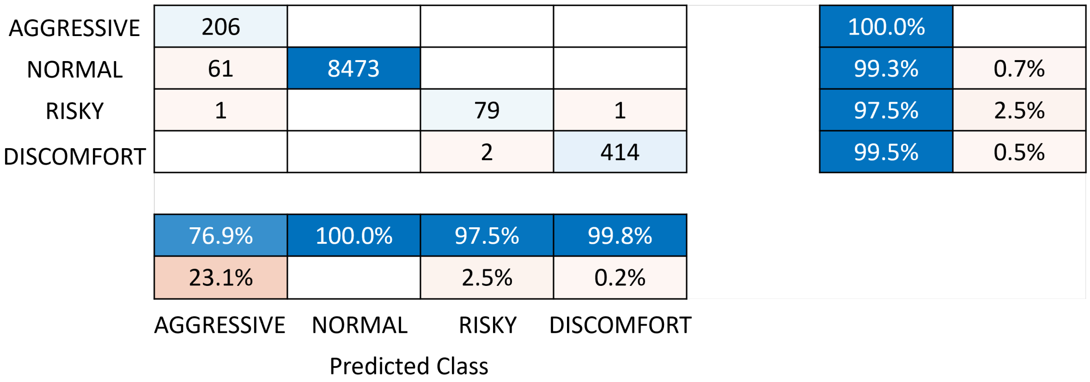


Figure 4.7: Loss curve. Blue solid line: training loss; Red dashed line: : test loss.



AGGRESSIVE	206				100.0%	
NORMAL	61	8473			99.3%	0.7%
RISKY	1		79	1	97.5%	2.5%
DISCOMFORT			2	414	99.5%	0.5%
	76.9%	100.0%	97.5%	99.8%		
	23.1%		2.5%	0.2%		
	AGGRESSIVE	NORMAL	RISKY	DISCOMFORT		

Predicted Class

Figure 4.8: Confusion matrix. Below the matrix: precision for each label, right side of the matrix: recall for each label.

(normal driving), risky, and *discomfort.* The algorithm demonstrates excellent performance in identifying normal, risky, and uncomfortable driving behaviors, with both precision and recall exceeding 97%. However, some miss classifications occur in general aggressive driving events, likely due to significant imbalances among different categories of samples. Since the identification of risky and uncomfortable behaviors is based on single feature threshold limits of TTC and

jerk, these classifications are largely unaffected by data imbalance. Overall, the model performs exceptionally well, as the majority of samples are correctly classified.

In Tab. 4.3, we compared the results of LSTM with RF. Due to the imbalance in the samples, we chose to evaluate the performance of the models using the F-score. It was found that LSTM outperforms RF across all labels. This is because LSTM has the capability to automatically extract useful features from raw time series data, whereas RF often faces limitations when dealing with high-dimensional data.

	RF	**LSTM**
F-score for aggressive	0.836	0.869
F-score for normal	0.991	0.996
F-score for risky	0.775	0.975
F-score for discomfort	0.847	0.994

Table 4.3: Performance comparison of RF and LSTM

Figure 4.9 illustrates how the algorithm detects and identifies aggressive driving behaviors by feeding the features.

The LSTM model outputs four potential predictions labeled as 0, 1, 2, and 3. By observing the lateral acceleration, it is noticed that around time 1.6 s, there is a sudden change in lateral acceleration (which could be attributed to abrupt maneuvers such as sharp turns or lane changes). The neural network's prediction shifts from 0 to 1, indicating the algorithm successfully identifies aggressive driving behavior. As a result, this methodology can be used to alert the driver if he deviates from his normal driving style.

4.2 Efficiency and aggressive driving recognition model

We collected 8 hours of driving data from three drivers. According to the preprocessing steps described in the methodology, 2,794,882 samples were divided into 139,767 MP, each MP with a time interval of 0.1 seconds, length of 0.3 seconds and includes 110 statistical features.

Further we classified all the MP into three groups based on their speed distribution using a K-means clustering algorithm with 3 clusters: low-speed, medium-speed, and high-speed driving (see Figure 4.10).

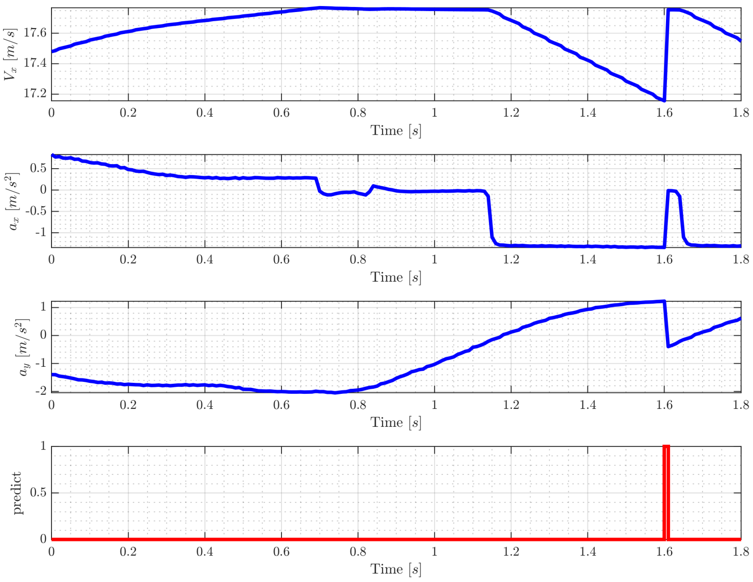


Figure 4.9: Neural network input and output. The first row: longitudinal velocity signal, the second row: longitudinal acceleration signal, the third row: lateral acceleration signal, the fourth row: aggressive driving behavior prediction. 1 represents aggressive behavior.

Each group was subdivided into five clusters, with classifications as follows: straight-line driving for steering angles less than 0.17 radians (10 degrees), slight steering for angles between 0.17 and 0.79 radians (10 to 45 degrees), and significant steering for angles greater than 0.79 radians (45 degrees). After segmenting each steering operation combination, we used a TTC threshold of 5 seconds to further divide the 15 EDB into 30 EDB. 15 EDB with a TTC of less than 5 seconds were directly labeled as aggressive driving behavior, other 15 EDB with TTC greater than 5 seconds are used for the I-DBSCAN algorithm (the EDB mentioned later in I-DBSCAN refers to them).

To avoid the curse of dimensionality, it is essential to extract key features from the 110 features of each MP. By applying the PCA method and considering 90% of the cumulative variance in the data, the input features were reduced to 17 key features. Fig. 4.11 displays the results of the PCA.

Before clustering the processed data, two critical hyperparameters for the DBSCAN algorithm had to be defined: minPts and radius (ϵ). minPts was set to twice

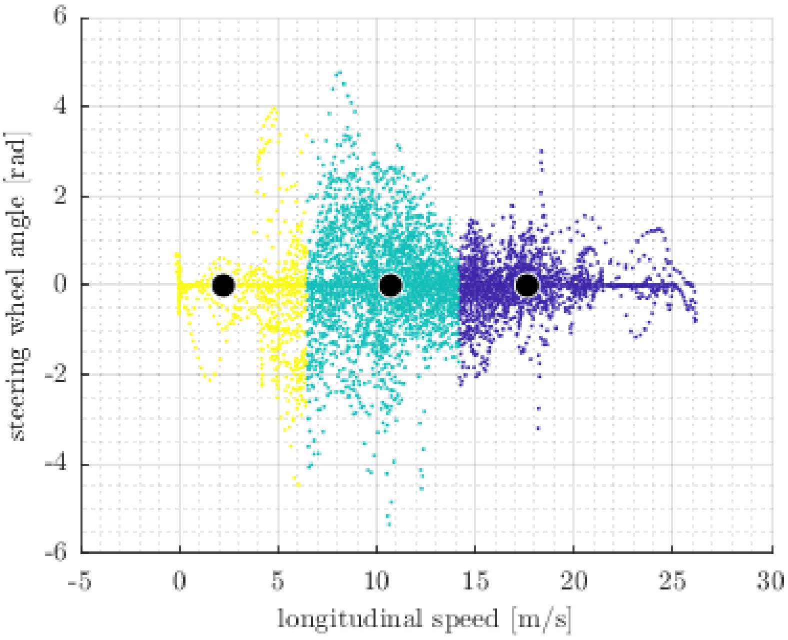


Figure 4.10: K-means clustering of speed for efficiency recognition. Black dot: centroid, yellow: low-speed data point, cyan: medium-speed data points, purple: high-speed data points.

the number of principal components, i.e., 34 and a suitable ϵ range was chosen between 2.4 and 12.54 by means of Elbow diagrams (see Fig. 4.12).

After completing all preprocessing steps, the data was input into the I-DBSCAN algorithm, where all EDB was run through at least three iterations after finding clusters containing more than 80% of the EDB data, the purpose of this step is to stopping algorithm after discovering clusters covering 80% of the EDB data. This indicates that the algorithm successfully classified normal driving behaviors as the main clusters and aggressive driving behaviors as noise. After combining EDB with TTC less than 5s, 89,005 MP were labeled as normal driving behaviors, marked with a 0, and 50,762 MP were labeled as aggressive driving behaviors, marked with a 1. To understand the differences between normal and aggressive driving behaviors, we extracted and compared some statistical features (such as averages, mean of maximum, and minimum values) from the EDB (see Tab. 4.4).

Standard deviations are displayed in parentheses. It can be observed that normal driving behaviors, due to their smaller standard deviations, exhibit higher consistency in a normal distribution; conversely, aggressive driving behaviors show greater

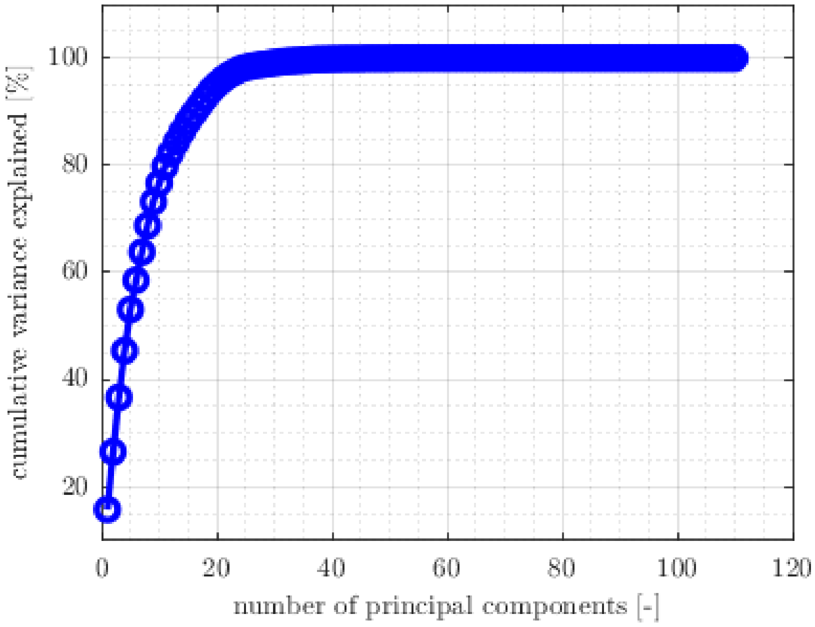


Figure 4.11: Principal component analysis for efficiency recognition.

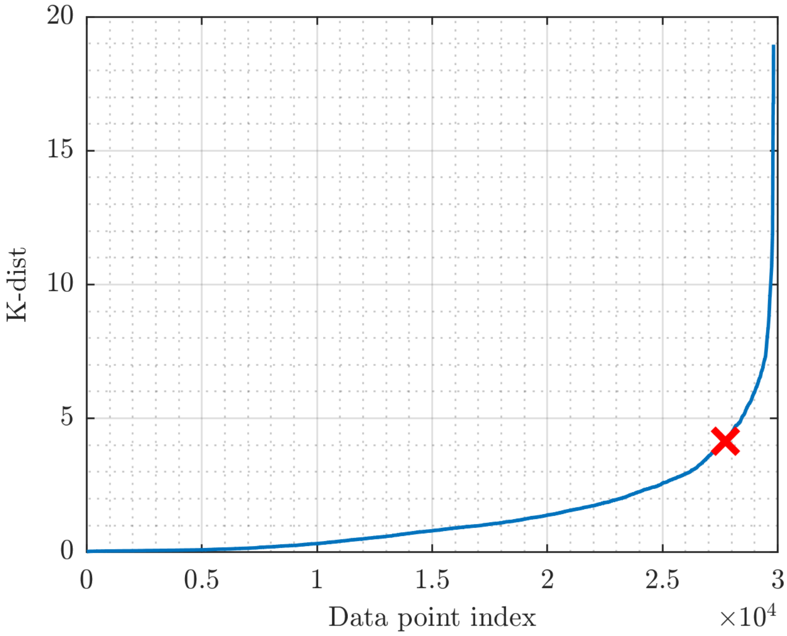


Figure 4.12: K-dist plot for efficiency recognition. Red cross: chosen ϵ.

dispersion due to larger standard deviations. This indicates that the algorithm successfully separated the outliers. By comparing the longitudinal acceleration of normal and aggressive driving (see Fig. 4.13), it is found that the action of slamming on the brakes is more likely to be identified as aggressive driving, as most red MP are concentrated in negative longitudinal acceleration. From Fig. 4.13, it is evident that some MP are not distinctly separated, primarily because they are clustered based on other input features, such as lateral acceleration or steering speed.

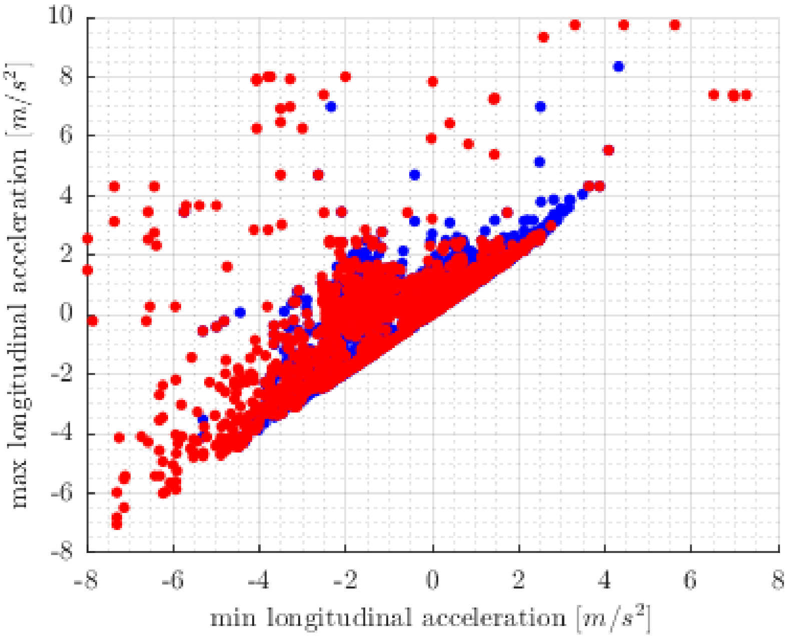


Figure 4.13: Max longitudinal acceleration vs Min longitudinal acceleration for efficiency recognition. Red dots: aggressive behavior, blue dots: normal behavior.

After successfully isolating aggressive driving behaviors from 15 EDB, we used 25th percentile of motor efficiency within the normal driving cluster as a threshold to further divide normal driving behaviors into two types: one associated with high efficiency driving behavior, and the other classified as inefficient driving behavior. The box plot (see Fig. 4.14) shows that the average operational efficiency for high efficiency driving is 94%, while the mean for inefficient driving behaviors is centered around 87%. Ultimately, this study successfully categorized the original unlabeled data into three distinct groups: high efficiency (labeled 0), aggressive (labeled 1), and inefficient driving behavior (labeled 2).

This study utilized 111,913 MP to train and optimize the neural network model,

Feature	Units	Normal	Aggressive
Average Speed	(m/s)	10.67 (5.31)	12.06 (5.74)
Average Longitudinal Acceleration	(m/s^2)	0.07 (1.3)	-0.27 (1.63)
Average Lateral Acceleration	(m/s^2)	-0.21 (1.84)	0.07 (2.26)
Average Steering Speed	(rad/s)	-0.2 (1.02)	-0.005 (1.16)
Max Speed	(m/s)	10.85 (5.33)	12.22 (5.64)
Max Longitudinal Acceleration	(m/s^2)	0.22 (1.29)	0.09 (1.71)
Max Lateral Acceleration	(m/s^2)	-0.03 (1.84)	0.18 (2.29)
Max Steering Speed	(rad/s)	-0.11 (1.03)	0.014 (1.14)
Min Speed	(m/s)	10.54 (5.33)	11.91 (5.53)
Min Longitudinal Acceleration	(m/s^2)	-0.09 (1.36)	-0.61 (1.81)
Min Lateral Acceleration	(m/s^2)	-0.40 (1.86)	-0.03 (2.25)
Min Steering Speed	(rad/s)	-0.29 (1.01)	-0.025 (1.18)

Table 4.4: Statistical feature of two labels for efficiency recognition. Values in round bracket are standard deviation.

while the remaining 27,854 MP were used as a test set to assess the model's performance. Before inputting features into the Bayesian optimization LSTM model, we reshaped the data using a sliding time window with a step size of 150 and an overlap rate of 99%, which helps the model learn the temporal information contained in the data. To achieve the best model, Bayesian optimization was employed to select appropriate initial hyperparameters. After 10 rounds of iteration, the initial hyperparameters were finalized as shown in Tab. 4.5.

The results are displayed in Fig. 4.15. This matrix is in a 3 x 3 format, where the 3 labels are aggressive, low, and high-efficiency driving behaviors. It intuitively

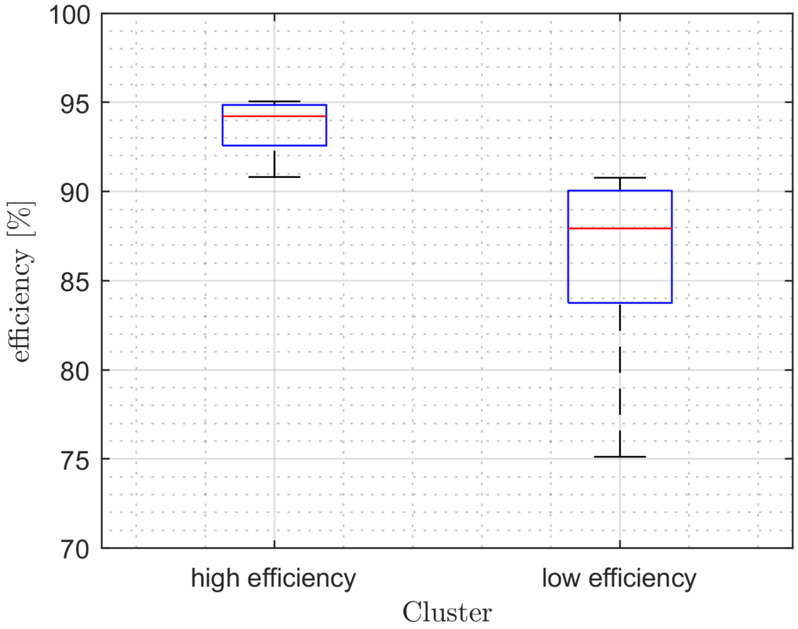


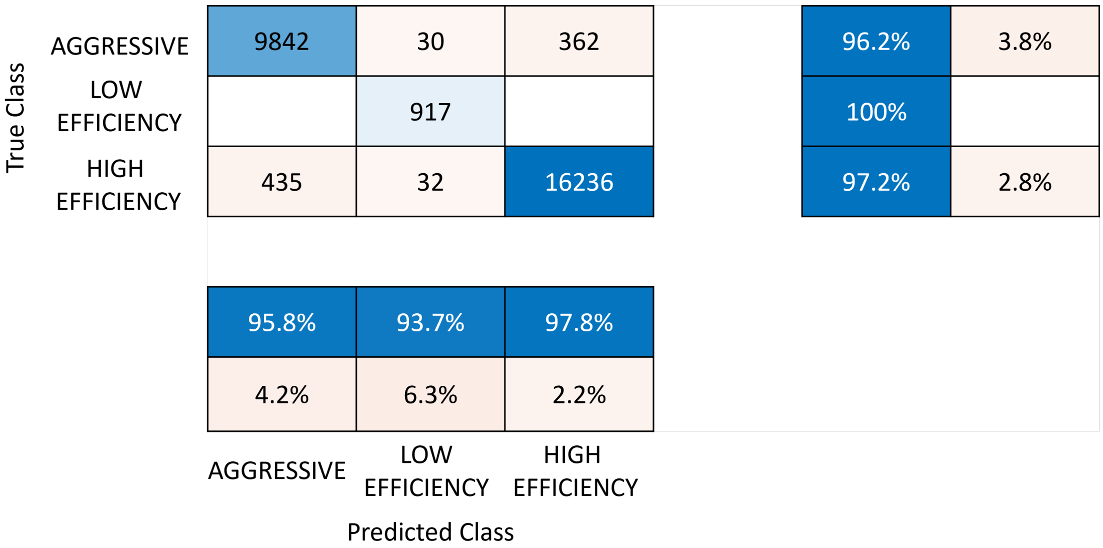
Figure 4.14: High efficiency vs low efficiency. Red dots: Outliers, red horizontal line: mean efficiency.

Hyperparameters	**Value**
LSTM number of layers	2
Number of hidden units	128
Maximum epochs	75
Batch size	128
Dropout rate	0.3
Initial learn rate	4.6663e-4
L2 Regularization	1.0343e-4
Loss function	Cross-Entropy

Table 4.5: Hyperparameters of LSTM network for efficiency recognition.

compares the model's classification predictions with the actual category labels defined by TTC threshold limits and I-DBSCAN algorithm.

By analyzing the confusion matrix, we can see that the model exhibits extremely high accuracy in predicting both high efficiency and aggressive driving behaviors, with precision and recall exceeding 95% for both. However, in predicting inefficient



True Class	AGGRESSIVE	LOW EFFICIENCY	HIGH EFFICIENCY		
AGGRESSIVE	9842	30	362	96.2%	3.8%
LOW EFFICIENCY		917		100%	
HIGH EFFICIENCY	435	32	16236	97.2%	2.8%
	95.8%	93.7%	97.8%		
	4.2%	6.3%	2.2%		

Figure 4.15: Confusion matrix for efficiency recognition. Below the matrix: precision for each label, right side of the matrix: recall for each label.

driving behaviors, performance slightly lower than the other two categories due to smaller number of samples in this category, but its precision and recall rates still exceed 90%. Overall, the model performs excellently in predicting these three types of driving behaviors, with the vast majority of samples being correctly classified. Due to the imbalance in data distribution, using the F-score is an excellent metric for evaluating model performance. Tab. 4.6 shows the F-scores of the RF and LSTM models and it is evident that both models perform exceptionally well across the three categories, with the RF model achieving F-scores exceeding 98% in all cases.

	RF	**LSTM**
F-score for aggressive	0.987	0.960
F-score for Low efficiency	0.991	0.967
F-score for High efficiency	0.992	0.975

Table 4.6: Performance comparison of RF and LSTM for efficiency recognition.

Fig. 4.16 shows the performance of the driving patterns recognition algorithm by feeding features. From Fig. 4.16, it is observed that between 1.2 seconds and 1.7 seconds, the motor efficiency suddenly drops. This behavior is detected by the model, causing the prediction to change from 0 to 2. Additionally, between 3.4

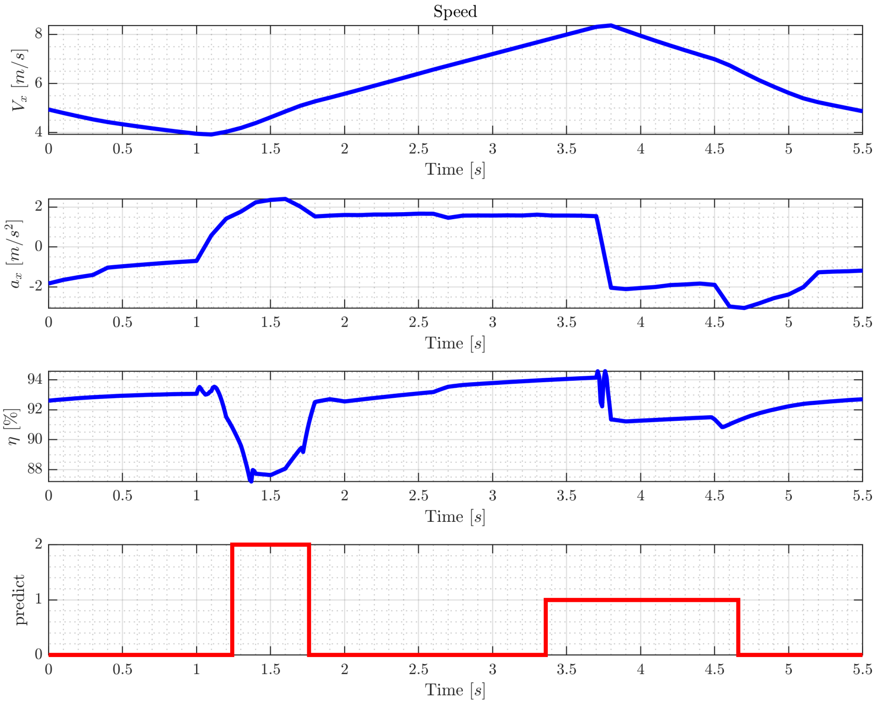


Figure 4.16: Neural network input and output for efficiency recognition. The first row: longitudinal velocity signal, the second row: longitudinal acceleration signal, the third row: motor efficiency signal, the fourth row: driving behavior prediction. 0 reprents high behavior, 1 represents aggressive behavior. 2 represents low efficiency behavior.

seconds and 4.6 seconds, the longitudinal acceleration drops sharply. This behavior is also captured by the model, resulting in the prediction change from 0 to 1.

Chapter 5

Conclusion

This thesis proposes a methodology to cluster the driving behavior using I-DBSCAN and predict using Bayesian optimization-based LSTM neural network and RF algorithm. Data were gathered through a driving simulator within an environment constructed in SCANeR^{TM}Studio.

It involves dividing the dataset into multiple monitoring periods and calculating statistical features for each period. The data are then segmented into Elementary Driving Behaviors and clustered at least three times using the I-DBSCAN algorithm for each EDB. Clusters representing 90% or 80% of the EDB data are labeled as normal driving behavior, with the remainder marked as aggressive driving behavior. Aggressive driving behaviors are further classified into those affecting comfort and safety based on Jerk and Time to Collision thresholds for discomfort and risk driving recognition. On the other hand, motor efficiency thresholds are used in efficiency and aggressive driving recognition. Then the labeled dataset is used to train an LSTM neural network and RF in the Matlab environment, and their performances are compared.

The results show that LSTM achieves better performance for discomfort and risk driving recognition model, with F-scores of 0.869, 0.975, and 0.994 for aggressive driving behavior, risky driving behavior, and discomfort driving behavior, respectively. For efficiency and aggressive driving recognition model. F-scores of RF are 0.987, 0.991, and 0.992 for aggressive driving behavior, low-efficiency driving behavior, and high-efficiency driving behavior, respectively. These results indicates that the algorithm has acceptable accuracy in identifying these driving behaviors.

Despite the algorithm's notable results, some areas require improvement, such as the data imbalance from three drivers affecting the model's performance. Expanding the dataset size, incorporating more drivers, and increasing the sample size of aggressive driving behaviors could further enhance model performance. The model has not yet been tested in a real-world environment, so its reliability in actual conditions remains to be verified.

Acknowledgements

Time flies like an arrow, and in the blink of an eye, three years of my academic journey have come to an end. Looking back on my three years of graduate studies, I have benefited immensely. Every bit of progress I have made is inseparable from the earnest teachings of my professors, the enthusiastic help of my classmates and friends, and the silent support of my family. Thank you all!

This thesis was completed under the meticulous guidance of Prof. Angelo Bonfitto and Prof. Shailesh Sudhakara Hegde. From selecting the topic, determining the experimental plan, theoretical analysis, data processing, to the writing and finalization of the thesis, they provided me with careful instruction and selfless assistance. The entire process of completing this thesis is permeated with their dedication and hard work.

During my three years as a graduate student, under the guidance of various professors, my theoretical knowledge was enhanced. I learned how to apply my knowledge to practical engineering, re-evaluating previously learned concepts from a practical perspective. The teachers have shown great concern and guidance in all aspects of my study, life, and conduct. From them, I have learned not only professional skills and knowledge but also composure in the face of difficulties and perseverance in work. Their open-mindedness, rigorous academic attitude, and diligent work ethic will serve as a lifelong example for me.

As I complete this thesis, I would like to express my deepest gratitude to Prof. Angelo Bonfitto, Prof. Shailesh Sudhakara Hegde, Feiyu Ou, Wei Li, and Xiangdong Wang. Without your assistance, this thesis would never have been completed. To my classmates and all my friends, your care and help have been invaluable; without them, I do not know how I would have navigated university life. Thank you to my peers! And to my family, especially my parents, Zhongli Ma and Lu Chen, for over twenty years of selfless dedication and meticulous upbringing, which have brought me to where I am today. Every bit of my achievement is a result of your hard work and effort. No matter where I go, your teachings will forever be etched in my heart.

Hao Chen
June 2024 at Politecnico di Torino

Bibliography

[1] *Global status report on road safety 2023*. en. URL: https://www.who.int/publications-detail-redirect/9789240086517 (visited on 06/11/2024) (cit. on p. 1).

[2] *AAA Foundation for Traffic Safety*. en-US. Sept. 2017. URL: https://aaafoundation.org/,%20https://aaafoundation.org/ (visited on 06/11/2024) (cit. on p. 1).

[3] «Aggressive Driving Research Update 2009». en. In: (2009) (cit. on p. 1).

[4] Jack N. Barkenbus. «Eco-driving: An overlooked climate change initiative». en. In: *Energy Policy* 38.2 (Feb. 2010), pp. 762–769. ISSN: 03014215. DOI: 10.1016/j.enpol.2009.10.021. URL: https://linkinghub.elsevier.com/retrieve/pii/S0301421509007484 (visited on 05/26/2024) (cit. on p. 2).

[5] Mike Knowles, Helen Scott, and David Baglee. «The effect of driving style on electric vehicle performance, economy and perception». en. In: *International Journal of Electric and Hybrid Vehicles* 4.3 (2012), p. 228. ISSN: 1751-4088, 1751-4096. DOI: 10.1504/IJEHV.2012.050492. URL: http://www.inderscience.com/link.php?id=50492 (visited on 05/26/2024) (cit. on p. 2).

[6] Alex Donkers, Dujuan Yang, and Miloš Viktorović. «Influence of driving style, infrastructure, weather and traffic on electric vehicle performance». en. In: *Transportation Research Part D: Transport and Environment* 88 (Nov. 2020), p. 102569. ISSN: 13619209. DOI: 10.1016/j.trd.2020.102569. URL: https://linkinghub.elsevier.com/retrieve/pii/S1361920920307562 (visited on 05/26/2024) (cit. on p. 2).

[7] MarkVollrath, Susanne Schleicher, and Christhard Gelau. «The influence of Cruise Control and Adaptive Cruise Control on driving behaviour – A driving simulator study». en. In: *Accident Analysis & Prevention* 43.3 (May 2011), pp. 1134–1139. ISSN: 00014575. DOI: 10.1016/j.aap.2010.12.023. URL: https://linkinghub.elsevier.com/retrieve/pii/S0001457510004008 (visited on 03/09/2024) (cit. on p. 3).

[8] Panos Konstantopoulos, Peter Chapman, and David Crundall. «Driver's visual attention as a function of driving experience and visibility. Using a driving simulator to explore drivers' eye movements in day, night and rain driving». en. In: *Accident Analysis & Prevention* 42.3 (May 2010), pp. 827–834. ISSN: 00014575. DOI: 10.1016/j.aap.2009.09.022. URL: https://linkinghub.elsevier.com/retrieve/pii/S0001457509002607 (visited on 03/09/2024) (cit. on p. 3).

[9] Giorgio Previati et al. *Cooperative Connected and Automated Mobility in a Roundabout*. English. SAE Technical Paper 2024-01-2002. ISSN: 0148-7191, 2688-3627. Warrendale, PA: SAE International, Apr. 2024. URL: https://www.sae.org/publications/technical-papers/content/2024-01-2002/ (visited on 03/12/2024) (cit. on p. 3).

[10] Omurcan Kumtepe, Gozde Bozdagi Akar, and Enes Yuncu. «Driver aggressiveness detection via multisensory data fusion». en. In: *EURASIP Journal on Image and Video Processing* 2016.1 (Dec. 2016), p. 5. ISSN: 1687-5281. DOI: 10.1186/s13640-016-0106-9. URL: https://jivp-eurasipjournals.springeropen.com/articles/10.1186/s13640-016-0106-9 (visited on 03/08/2024) (cit. on p. 3).

[11] Kwan Lee, Hyo Yoon, Jong Song, and Kang Park. «Convolutional Neural Network-Based Classification of Driver's Emotion during Aggressive and Smooth Driving Using Multi-Modal Camera Sensors». en. In: *Sensors* 18.4 (Mar. 2018), p. 957. ISSN: 1424-8220. DOI: 10.3390/s18040957. URL: http://www.mdpi.com/1424-8220/18/4/957 (visited on 03/08/2024) (cit. on p. 3).

[12] Arash Jahangiri, Vincent J. Berardi, and Sahar Ghanipoor Machiani. «Application of Real Field Connected Vehicle Data for Aggressive Driving Identification on Horizontal Curves». In: *IEEE Transactions on Intelligent Transportation Systems* 19.7 (July 2018), pp. 2316–2324. ISSN: 1524-9050, 1558-0016. DOI: 10.1109/TITS.2017.2768527. URL: https://ieeexplore.ieee.org/document/8169689/ (visited on 03/08/2024) (cit. on p. 3).

[13] Yiwen Zhou, Fengxiang Guo, Simin Wu, Wenyao He, Xuefei Xiong, Zheng Chen, and Dingan Ni. «Safety and Economic Evaluations of Electric Public Buses Based on Driving Behavior». en. In: *Sustainability* 14.17 (Aug. 2022), p. 10772. ISSN: 2071-1050. DOI: 10.3390/su141710772. URL: https://www.mdpi.com/2071-1050/14/17/10772 (visited on 05/08/2024) (cit. on pp. 3, 5).

[14] Philipp Themann, Julian Bock, and Lutz Eckstein. «Optimisation of energy efficiency based on average driving behaviour and driver's preferences for automated driving». en. In: *IET Intelligent Transport Systems* 9.1 (Feb. 2015),

pp. 50–58. ISSN: 1751-9578, 1751-9578. DOI: 10.1049/iet-its.2013.0121. URL: https://onlinelibrary.wiley.com/doi/10.1049/iet-its.2013.0121 (visited on 05/26/2024) (cit. on p. 3).

[15] Valentina Gatteschi, Alberto Cannavo, Fabrizio Lamberti, Lia Morra, and Paolo Montuschi. «Comparing Algorithms for Aggressive Driving Event Detection Based on Vehicle Motion Data». In: *IEEE Transactions on Vehicular Technology* 71.1 (Jan. 2022), pp. 53–68. ISSN: 0018-9545, 1939-9359. DOI: 10.1109/TVT.2021.3122197. URL: https://ieeexplore.ieee.org/document/9585660/ (visited on 03/08/2024) (cit. on p. 4).

[16] Karam Darwish and Majd Ali. «Driving Behaviors Recognition Using Deep Neural Networks». en. In: *Embedded Selforganising Systems* (July 2023), 9–12 Pages. DOI: 10.14464/ESS.V10I5.592. URL: https://www.bibliothek.tu-chemnitz.de/ojs/index.php/cs/article/view/592 (visited on 03/08/2024) (cit. on p. 4).

[17] D. Deva Hema and K. Ashok Kumar. «Hyperparameter optimization of LSTM based Driver's Aggressive Behavior Prediction Model». In: *2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS)*. Coimbatore, India: IEEE, Mar. 2021, pp. 752–757. ISBN: 978-1-72819-537-7. DOI: 10.1109/ICAIS50930.2021.9396047. URL: https://ieeexplore.ieee.org/document/9396047/ (visited on 03/12/2024) (cit. on pp. 4, 32).

[18] Korosh Vatanparvar, Sina Faezi, Igor Burago, Marco Levorato, and Mohammad Abdullah Al Faruque. «Extended Range Electric Vehicle With Driving Behavior Estimation in Energy Management». In: *IEEE Transactions on Smart Grid* 10.3 (May 2019), pp. 2959–2968. ISSN: 1949-3053, 1949-3061. DOI: 10.1109/TSG.2018.2815689. URL: https://ieeexplore.ieee.org/document/8315470/ (visited on 05/26/2024) (cit. on p. 4).

[19] Laura Ferrarotti et al. «Autonomous and Human-Driven Vehicles Interacting in a Roundabout: A Quantitative and Qualitative Evaluation». In: *IEEE Access* 12 (2024), pp. 32693–32705. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2024.3370469. URL: https://ieeexplore.ieee.org/document/10445226/ (visited on 03/12/2024) (cit. on p. 4).

[20] Dinesh Cyril Selvaraj, Shailesh Hegde, Nicola Amati, Francesco Deflorio, and Carla Fabiana Chiasserini. «An ML-Aided Reinforcement Learning Approach for Challenging Vehicle Maneuvers». In: *IEEE Transactions on Intelligent Vehicles* 8.2 (Feb. 2023), pp. 1686–1698. ISSN: 2379-8904, 2379-8858. DOI: 10.1109/TIV.2022.3224656. URL: https://ieeexplore.ieee.org/document/9963702/ (visited on 03/12/2024) (cit. on p. 4).

[21] Ruoqi Wang, Liangyao Yu, and Yong Huang. «A Collision Avoidance Strategy Based on Inevitable Collision State». In: Sept. 2022, pp. 2022–01–1170. DOI: 10.4271/2022-01-1170. URL: https://www.sae.org/content/2022-01-1170/ (visited on 03/12/2024) (cit. on p. 4).

[22] B. G. Simons-Morton, Z. Zhang, J. C. Jackson, and P. S. Albert. «Do Elevated Gravitational-Force Events While Driving Predict Crashes and Near Crashes?» en. In: *American Journal of Epidemiology* 175.10 (May 2012), pp. 1075–1079. ISSN: 0002-9262, 1476-6256. DOI: 10.1093/aje/kwr440. URL: https://academic.oup.com/aje/article-lookup/doi/10.1093/aje/kwr440 (visited on 03/08/2024) (cit. on p. 4).

[23] Jinzhen Wang, Yiming Cheng, and Liangyao Yu. «Racing Driver Modeling Based on Driving Behavior». In: *Volume 1: 23rd International Conference on Advanced Vehicle Technologies (AVT)*. Virtual, Online: American Society of Mechanical Engineers, Aug. 2021, V001T01A011. ISBN: 978-0-7918-8536-9. DOI: 10.1115/DETC2021-71113. URL: https://asmedigitalcollection.asme.org/IDETC-CIE/proceedings/IDETC-CIE2021/85369/V001T01A011/1128343 (visited on 03/12/2024) (cit. on p. 4).

[24] Laura Eboli, Gabriella Mazzulla, and Giuseppe Pungillo. «Combining speed and acceleration to define car users' safe or unsafe driving behaviour». en. In: *Transportation Research Part C: Emerging Technologies* 68 (July 2016), pp. 113–125. ISSN: 0968090X. DOI: 10.1016/j.trc.2016.04.002. URL: https://linkinghub.elsevier.com/retrieve/pii/S0968090X16300067 (visited on 03/08/2024) (cit. on p. 4).

[25] Anusha Adavikottu and Nagendra R Velaga. «Analysis of speed reductions and crash risk of aggressive drivers during emergent pre-crash scenarios at unsignalized intersections». en. In: *Accident Analysis & Prevention* 187 (July 2023), p. 107088. ISSN: 00014575. DOI: 10.1016/j.aap.2023.107088. URL: https://linkinghub.elsevier.com/retrieve/pii/S0001457523001355 (visited on 03/08/2024) (cit. on pp. 4, 31).

[26] Y. Bai. «Research on driving energy-saving technology». In: *Chang'an University: Xi'an, China* (2011) (cit. on p. 5).

[27] Sara Luciani, Angelo Bonfitto, Nicola Amati, and Andrea Tonoli. «Comfort-Oriented Design of Model Predictive Control in Assisted and Autonomous Driving». In: *Volume 4: 22nd International Conference on Advanced Vehicle Technologies (AVT)*. Virtual, Online: American Society of Mechanical Engineers, Aug. 2020, V004T04A008. ISBN: 978-0-7918-8393-8. DOI: 10.1115/DETC2020-22418. URL: https://asmedigitalcollection.asme.org/IDETC-CIE/proceedings/IDETC-CIE2020/83938/Virtual,%20Online/1089870 (visited on 03/12/2024) (cit. on p. 5).

[28] Sara Luciani, Angelo Bonfitto, Nicola Amati, and Andrea Tonoli. «Model predictive control for comfort optimization in assisted and driverless vehicles». en. In: *Advances in Mechanical Engineering* 12.11 (Nov. 2020), p. 168781402097453. ISSN: 1687-8140, 1687-8140. DOI: 10.1177/1687814020974532. URL: http://journals.sagepub.com/doi/10.1177/1687814020974532 (visited on 03/12/2024) (cit. on p. 5).

[29] *AVSimulation, «SCANeR™studio's documentation».* (Cit. on pp. 9, 10, 15, 16).

[30] Charles Marks, Arash Jahangiri, and Sahar Ghanipoor Machiani. «Iterative DBSCAN (I-DBSCAN) to Identify Aggressive Driving Behaviors within Unlabeled Real-World Driving Data». In: *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. Auckland, New Zealand: IEEE, Oct. 2019, pp. 2324–2329. ISBN: 978-1-5386-7024-8. DOI: 10.1109/ITSC.2019.8917162. URL: https://ieeexplore.ieee.org/document/8917162/ (visited on 03/08/2024) (cit. on pp. 24, 30).

[31] Alessandro Mauro Danusso. «Aggressive driving detection through Iterative DB-SCAN labeling and supervised pattern recognition». Book Title: Aggressive driving detection through Iterative DB-SCAN labeling and supervised pattern recognition. PhD thesis. Politecnico di Torino, 2022 (cit. on pp. 24, 26, 30).

[32] Arash Jahangiri, Sahar Ghanipoor Machiani, and Vahid Balali. «Big Data Exploration to Examine Aggressive Driving Behavior in the Era of Smart Cities». en. In: *Data Analytics for Smart Cities*. Ed. by Amir Alavi and William G. Buttlar. 1st ed. Boca Raton, Florida : CRC Press, Taylor & Francis Group, [2019]: Auerbach Publications, Oct. 2018, pp. 163–182. ISBN: 978-0-429-43498-3. DOI: 10.1201/9780429434983-7. URL: https://www.taylorfrancis.com/books/9780429786631/chapters/10.1201/9780429434983-7 (visited on 03/08/2024) (cit. on p. 26).

[33] Chris Schwarz. «On computing time-to-collision for automation scenarios». en. In: *Transportation Research Part F: Traffic Psychology and Behaviour* 27 (Nov. 2014), pp. 283–294. ISSN: 13698478. DOI: 10.1016/j.trf.2014.06.015. URL: https://linkinghub.elsevier.com/retrieve/pii/S1369847814000898 (visited on 03/08/2024) (cit. on pp. 26, 31).

[34] *Unsupervised Learning: K-Means Clustering \textbar by Diego Unzueta \textbar Towards Data Science.* Mar. 2024. URL: https://towardsdatascience.com/unsupervised-learning-k-means-clustering-6fd72393573c?gi=e2f9f4562669 (visited on 03/08/2024) (cit. on p. 27).

[35] Martin Ester, Hans-Peter Kriegel, and Xiaowei Xu. «A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise». en. In: () (cit. on p. 28).

[36] Jörg Sander, Martin Ester, Hans-Peter Kriegel, and Xiaowei Xu. «Density-Based Clustering in Spatial Databases: The Algorithm GDBSCAN and its Applications». en. In: () (cit. on p. 30).

[37] Erich Schubert, Jörg Sander, Martin Ester, Hans Peter Kriegel, and Xiaowei Xu. «DBSCAN Revisited, Revisited: Why and How You Should (Still) Use DBSCAN». en. In: *ACM Transactions on Database Systems* 42.3 (Sept. 2017), pp. 1–21. ISSN: 0362-5915, 1557-4644. DOI: 10.1145/3068335. URL: https://dl.acm.org/doi/10.1145/3068335 (visited on 03/08/2024) (cit. on p. 30).

[38] Kevin Beyer, Jonathan Goldstein, Raghu Ramakrishnan, and Uri Shaft. «When Is "Nearest Neighbor" Meaningful?» In: *Database Theory — ICDT'99*. Ed. by Catriel Beeri, Peter Buneman, Gerhard Goos, Juris Hartmanis, and Jan Van Leeuwen. Vol. 1540. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999, pp. 217–235. ISBN: 978-3-540-65452-0 978-3-540-49257-3. DOI: 10.1007/3-540-49257-7_15. URL: http://link.springer.com/10.1007/3-540-49257-7_15 (visited on 03/08/2024) (cit. on p. 30).

[39] Fred Feng, Shan Bao, James R. Sayer, Carol Flannagan, Michael Manser, and Robert Wunderlich. «Can vehicle longitudinal jerk be used to identify aggressive drivers? An examination using naturalistic driving data». en. In: *Accident Analysis & Prevention* 104 (July 2017), pp. 125–136. ISSN: 00014575. DOI: 10.1016/j.aap.2017.04.012. URL: https://linkinghub.elsevier.com/retrieve/pii/S0001457517301409 (visited on 03/08/2024) (cit. on p. 31).

[40] Wenjing Zhao, Siyuan Gong, Dezong Zhao, Fenglin Liu, N.N. Sze, and Helai Huang. «Effects of collision warning characteristics on driving behaviors and safety in connected vehicle environments». en. In: *Accident Analysis & Prevention* 186 (June 2023), p. 107053. ISSN: 00014575. DOI: 10.1016/j.aap.2023.107053. URL: https://linkinghub.elsevier.com/retrieve/pii/S0001457523001008 (visited on 03/08/2024) (cit. on p. 31).

[41] Bryan Lim and Stefan Zohren. «Time-series forecasting with deep learning: a survey». en. In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 379.2194 (Apr. 2021), p. 20200209. ISSN: 1364-503X, 1471-2962. DOI: 10.1098/rsta.2020.0209. URL: https://royalsocietypublishing.org/doi/10.1098/rsta.2020.0209 (visited on 03/09/2024) (cit. on p. 32).

[42] Leo Breiman. «Random Forests». en. In: *Machine Learning* 45.1 (Oct. 2001), pp. 5–32. ISSN: 1573-0565. DOI: `10.1023/A:1010933404324`. URL: `https://doi.org/10.1023/A:1010933404324` (visited on 05/25/2024) (cit. on p. 34).

[43] V. Rodriguez-Galiano, M. Sanchez-Castillo, M. Chica-Olmo, and M. Chica-Rivas. «Machine learning predictive models for mineral prospectivity: An evaluation of neural networks, random forest, regression trees and support vector machines». en. In: *Ore Geology Reviews* 71 (Dec. 2015), pp. 804–818. ISSN: 01691368. DOI: `10.1016/j.oregeorev.2015.01.001`. URL: `https://linkinghub.elsevier.com/retrieve/pii/S0169136815000037` (visited on 05/25/2024) (cit. on p. 34).

[44] Ahmad Abdulla, George Baryannis, and Ibrahim Badi. «An integrated machine learning and MARCOS method for supplier evaluation and selection». en. In: *Decision Analytics Journal* 9 (Dec. 2023), p. 100342. ISSN: 27726622. DOI: `10.1016/j.dajour.2023.100342`. URL: `https://linkinghub.elsevier.com/retrieve/pii/S2772662223001820` (visited on 07/04/2024) (cit. on p. 34).