Building a Database of Simulated Driver Behaviors Using the SUMO Simulator

Chapter · July 2024

DOI: 10.1007/978-3-031-66428-1_34

6 authors, including:

Badreddine Chah
University of Technology of Belfort-Montbéliard
5 PUBLICATIONS **23** CITATIONS
SEE PROFILE

Alexandre Lombard
University of Technology of Belfort-Montbéliard
34 PUBLICATIONS **203** CITATIONS
SEE PROFILE

Yazan Mualla
University of Technology of Belfort-Montbéliard
59 PUBLICATIONS **448** CITATIONS
SEE PROFILE

Anis Bkakria
Institut Mines-Télécom
27 PUBLICATIONS **107** CITATIONS
SEE PROFILE

Building a Database of Simulated Driver Behaviors Using the SUMO Simulator

Badreddine Chah[1(✉)], Alexandre Lombard[1], Yazan Mualla[1], Anis Bkakria[2], Abdeljalil Abbas-Turki[1], and Reda Yaich[2]

[1] UTBM, CIAD UMR 7533, 90010 Belfort cedex, France
Badreddine.chah@utbm.fr
[2] IRT SystemX, Palaiseau 91120, France

Abstract. Classifying the driving styles is of particular interest for enhancing road safety in smart cities. The vehicle can assist the driver by providing advice to increase awareness of potential dangers. Accordingly, dissuasive measures, such as adjusting insurance costs, can be implemented. The service is called Pay-As-You-Drive insurance (PAYD), and to address it, the paper introduces a method for constructing a database of simulated driver behaviors using the Simulation of Urban MObility Simulation of Urban MObility (SUMO) simulator. Three levels of driver behavior (slow, normal, and dangerous) are generated using the Intelligent Driver Model car-following, with parameters adjusted accordingly. The simulation takes place on the Miami city map, extracted from a real-world map, encompassing various road types and traffic signs. The control interface 'TraCI' is employed to collect data from the simulated vehicles and to trigger alerts based on violations committed by the drivers. These alerts are then used to train four machine learning (ML) models, for labeling the driving style. The four ML models are Gradient Boosted Decision Trees, K-Nearest Neighbors, Multi-layer Perceptron, and Support Vector Machines. The paper demonstrates the feasibility and accuracy of the proposed AlertDang Driver Profiling method, providing the code and dataset on GitHub. Additionally, it discusses the advantages and limitations of the simulation-based approach and suggests potential future directions.

Keywords: Simulated driving behavior dataset · Dangerous driving behavior · Sumo simulations · Connected and autonomous vehicles · Usage-based insurance · Pay-how-you-drive insurance · Car insurance · Private machine learning construction

1 Introduction

The growing demand for transportation presents various challenges related to traffic flow, pollution, energy consumption, and public health. In 2021, approximately 1.19 million deaths were attributed to road accidents, with two-thirds involving individuals aged between 19 and 59 [1]. Road accidents also stand as

K. Arai (Ed.): IntelliSys 2024, LNNS 1066, pp. 536–555, 2024.
https://doi.org/10.1007/978-3-031-66428-1_34

a leading cause of death among children and young adults aged 5 to 29[1]. With advancements in autonomous vehicles, numerous sensors have become standard in modern vehicles to alert drivers to potentially dangerous events [2,3]. However, a critical question arises concerning how the information coming from sensors can be utilized to enhance driving behavior within a legal framework that respects privacy. One effective method to prevent accidents is the implementation of systems capable of assuming control in imminent danger cases, such as emergency braking systems. Nevertheless, the driver is the one who controls the vehicle and can choose to ignore alerts or deactivate them. Another approach involves scoring driving styles to make drivers aware of the potential danger raised by their usual behaviors, similar to existing eco-driving systems. Such systems can also be utilized by insurance companies to incentivize or discourage drivers based on their driving style.

To offer insurance services such as Pay As You Drive (PAYD) insurance, drivers' personal data is needed for processing. This data has to be labeled at least into three classes: **Dangerous**, characterized by sudden acceleration, braking, and aggressive behavior on the road. **Normal**, representing average driving events. **Slow**, indicating the maintenance of a speed lower than the average. The labeled dataset will be used to train machine learning (ML) algorithms. In the case of PAYD insurance, it will be easier to identify aggressive drivers who have to pay more for their insurance. To get this data, two approaches are feasible: utilizing real data acquired from actual drivers or building data from a simulator, such as Simulation of Urban MObility (SUMO) [15]. The volume of this data is extensive and complex, making it impractical for manual manipulation by a human. This is one of the reasons why Artificial intelligence (AI) is employed to make predictions on new data based on patterns and trends identified in the training data. The aspect of protecting the confidentiality of sensitive driver data is also raised. In particular, privacy during the training of the ML model can be preserved through the use of privacy-enhancing technologies (PETs) ([4–7] for more details).

So far, three challenges have been identified:

1. Creating an environmental simulation that accurately models vehicle behavior to replicate real-world road conditions on a large scale;
2. Collecting specific data for driver behavior profiling;
3. Ensuring the reliability of the dataset for training any ML model aimed at classifying driver behavior.

The contribution of this work is twofold: First, we build a SUMO simulation to facilitate large-scale testing, which is essential for demonstrating capabilities under a variety of conditions. We aim to conduct these tests in a controlled and reproducible manner, bringing the simulation closer to real-world scenarios. To simulate real-world driver behavior, we choose the Intelligent Driver Model (IDM) as the car-following model in this research (see Sect. 3.2 for more details).

[1] https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries, accessed on 03/02/2023

The simulation is conducted on the Miami city map, extracted from a real-world map, incorporating traffic lines and various road types (such as highways, local streets, roundabouts, road signs, and so on). We use Traffic Control Interface (TraCI) [16] to access a running road traffic simulation, enabling the retrieval of values from simulated objects and the real-time manipulation of their behavior.

Second, we propose a solution that allows for the classification of driver behavior. The solution is based on warnings; for example, speeding triggers a warning. We use TraCI to collect various data points (e.g., speed, acceleration, safe distance, position, ID, etc.). Therefore, warnings are triggered after each violation committed by a driver. We propose a few warnings, but we can add others if necessary.

The main objective of the previous step is to collect a set of data of warnings on several drivers. The dataset is used to train an ML model to classify other drivers. We present an experimental step to demonstrate the feasibility of our results by applying several ML models to the obtained dataset. PAYD insurance would be an example application of this classification.

This work overcomes the limitations prevalent in existing studies. Notable advances include improved scalability, maps with a wide range of routes, the longest vehicle trajectory constraints, diversity of data collection compared to alternative methods, and reduced material and equipment costs. In addition, this SUMO-based simulation prioritizes the protection of privacy by opting for simulated data rather than actual personal information. Details are provided in Sect. 2. The code and the dataset are available on GitHub[2] and can be used for further collection of data and improving the dataset. To our knowledge, no dataset on driver behavior has been provided by a full simulator in the literature.

The paper is organized as follows: Sect. 2 presents the related work and the existing limitations. We answer how our work overcomes the limitations. In Sect. 3, we introduce the important SUMO simulation component: Miami City maps and the car-following model. We then present the impact of the simulation parameters on driver behavior. In Sect. 4, we present the driver profiling method based on warning, called AlertDang. Section 5 includes an analysis of the data collected from the SUMO simulation and an application of ML classification to the warning datasets. We conclude our paper in Sect. 6 by summarizing and discussing the benefits and possible improvements of our work.

2 Related Work

Several approaches have been proposed for acquiring authentic data from real driving scenarios and using it to identify driver behavior. They differ from one another: There are those processed from in-vehicle Controller Area Network (CAN) data [8,9] and those based on devices present in cars [11], such as smartphones. Some others are based on a simulation [12–14]. Table 1 resumes the studies that focused on profiling and classifying drivers through data analysis.

[2] https://github.com/badr007-01/Building-a-Database-of-Simulated-Driver-Behaviors-Using-the-SUMO-Simulator.

Table 1. Comparison of the related work with our proposed work

Research	Collected dataset	Features	Classification algorithms
[8]	In-vehicle's CAN network data	Mechanical features from automotive parts such as engine, fuel, and transmission	Decision Tree, Random Forest, KNN, and MLP
[9]	In-vehicle's CAN network data	Acceleration, brake, steering wheel, vehicle speed, engine speed, gear shift, yaw rate, shaft angular velocity, engine torque, fuel consumption, throttle position, turn signal	SVM, Random Forest, Naive Bayes, and KNN
[11]	Smartphone data	Acceleration (X, Y, and Z axis) and the Gyroscope (X, Y, and Z axis)	Random Forest, XGBoost, Dense NN, LSTM, CNN LSTM, and ConvLSTM
[12]	Driving simulation data	Acceleration, steering wheel	HMM
[13]	Driving simulation data	Acceleration, brake, and steering wheel	HMM
[14]	Driving simulation data	Vehicle speed, following distance from vehicle ahead, accelerator pedal pressure, and brake pedal pressure	Gaussian Mixture Model (GMM)
Our Work	Driving simulation data	Distance driven, Acceleration, Speed, Allowed Speed, Gap, and 5 different warning	GBDT, KNN, MLP, and SVM

The first type of approach experimentally investigates the possibility of identifying individuals using snippets of sensor data from their natural driving behavior. More specifically, the authors recorded data from on-board sensors on the CAN of a typical modern vehicle. For instance, the authors in [9] recorded the data streams from 16 sensors broadcast on the car's internal computer network. They are doing the same for 15 drivers (seven women, eight men). The drivers completed (1) three laps of a closed circuit consisting of parking and avoidance maneuvers, and (2) around 50 km of open road driving. From this experiment, they derived multiple data: brake pedal position, steering wheel angle, longitudinal acceleration, turning speed, driving speed, current gear, acceleration pedal position, engine speed, maximum engine torque, fuel consumption rate, and throttle position. The authors of [8] present driving data from a recent Kia Motors Corporation model in South Korea. 10 drivers took part in the experiments on four routes in Seoul. The route is made up of three types of urban lanes, motorized lanes and parking spaces, with a total length of 23 km. The

data is recorded every one second during driving. The authors collected 51 features through the OBD-II, which has a total of 94,401 records recorded every second with a size of 16.7Mb in total. The data collected for both papers utilize ML algorithms for profiling, including MLP, SVM, Random Forest, Naive Bayes, and k-nearest Neighbor (KNN) algorithms. Some devices such as smartphones can be used to collect the data. but this time the features will be different. The authors of study [11] collect the data using a Samsung Galaxy S10 smartphone in a vehicle (Dacia Sandero 1.4 MPI). The features taken into consideration for driver profiling are the Acceleration (X, Y, and Z axis) and the Gyroscope (X, Y, and Z axis).

Another possibility is to rely on mixed reality to build a driver behavior dataset. The "Sim Racing" simulator can collect data on driving behavior. In study [12], the authors run an experiment virtual road scenario and the environment was designed based on the AutoSim driving simulator. The simulated road extended 14 km long, including 12 straight lines, five horizontal curves, four obstacle zones and three parking bays. The participating drivers are a group of men and women who have been able to drive in a variety of circumstances. The authors indicate the type of behavior to be adopted by each participant. It is used to label the data collected for each driver. The dataset proposed is based on the following characteristics: accelerator, steering wheel, brake, clutch, and speed. In a parallel study [13], a real-time graphic driving simulator-for collecting and modeling human driving behaviors. The dataset is derived from seven drivers and then processed through a hidden Markov model (HMM) for training human behavior models. The authors propose the dataset with features including acceleration, brake, and steering wheel data. The research aimed to analyze the characteristics of individual drivers. Subsequently, the authors of study [14] propose driver identification methods based on driving behavior signals. Similarly, the authors are using a graphic driving simulator-for collecting and modeling human driving behaviors. It generated the simulation data and compared the identification performance using different models. The dataset is based on only three participants driving on a two-lane freeway with an actual Japanese highway.

To compare the above works, we present the advantages that each approach offers over the others and their limitations. **A:** Real-world road experiment-based approaches present some advantages. The approach provides real data from natural behaviors. Also, it can reveal unexpected patterns or scenarios that might not be taken into account in specific simulations. However, it presents several limitations:

- **No scalability:** The number of tests and distances covered is limited.
- **Restricted map options:** The variety of trajectories is limited, lacking diversity in road types such as local streets, highways, state roads, boulevards, avenues, and more. as well as diversity in areas, including residential areas, business districts, industrial zones, etc.
- **Dangerous driver accuracy:** Testing extreme, dangerous, or unlikely scenarios can pose difficulties.

- **Labeled dataset:** the driver behavior dataset that presents two papers [8,9] is not labeled.
- **Profiling Method:** In the presence of a huge number of samples within the dataset, classification algorithms pose challenges for ML models. Some may become computationally expensive, while others can lead to overfitting and under-fitting problems.
- **Investing Time and Money:** The costs associated with real-world trials can result in significant time and high expenses for implementation.
- **Privacy:** The use of sensitive driver data may raise privacy concerns and should comply with data protection regulations, such as the General Data Protection Regulation (GDPR) [10] in the European Union.

B: The Sim Racing simulator-based approach [12–14] can offer a solution to address some of these challenges. Money and time are saved when the simulation can be carried out on-site, enabling the data to be retrieved quickly. In addition, the flexibility precisely extends to extreme scenarios. Moreover, the dataset is labeled in the way the authors ask drivers to drive differently. Nevertheless, there are additional challenges that require attention.

Our work tries to tick all the limits set out above (see Sect. 2). The limitations presented by the related work have been overcome. Firstly, the constraints associated with investing time and money in real-life testing have been alleviated. Our proposed SUMO simulation reduces the time and expense involved. Additionally, the question of scalability was addressed, allowing the range of tests and distances covered by vehicles to be extended. The map of Miami has been chosen to include various trajectories and zones, offering a comprehensive representation of road types and locations. It is now easier to accurately assess dangerous driver behavior. Furthermore, the datasets collected as part of this study have been labeled, providing valuable information for training ML models. Additionally, the profiling method based on the warning ensures the efficient process of numerous samples without incurring computational costs or falling prey to over- or under-fitting problems. The experiments described in the previous section show that the dataset gives favorable results, although it could be further improved to achieve greater accuracy. Lastly, simulated data, which doesn't contain sensitive information, can be utilized during ML algorithm training. This allows us to build robust algorithms without exposing sensitive data directly. Subsequently, the model can be further enhanced by applying PETs when incorporating real driver data ([4–7] for more details). Table 2 shows comparisons of the limitations addressed by the related work and the proposed approach.

3 SUMO-Based Driver Behavior Simulation Framework

The SUMO simulator is an open-source driving simulator. The reasons we chose this simulator are twofold: First, it allows the modeling of intermodal traffic systems, including road vehicles, public transport, and pedestrians. Second, it has multiple flexible tools that handle tasks such as visualization, network import,

Table 2. Comparison of the limitations addressed by the related work and the proposed approach

Research	Scalability	Reduce Time and Money	Restricted map options	Dangerous driver accuracy	Profiling Method	Privacy
Our Work	✓	✓	✓	✓	✓	✓
[8]	✗	✗	✗	✗	✗	✗
[9]	✗	✗	✗	✗	✗	✗
[11]	✗	✗	✗	✗	✗	✗
[12]	✓	✓	✗	✗	✗	✗
[13]	✓	✓	✗	✗	✗	✗
[14]	✓	✓	✗	✗	✗	✗

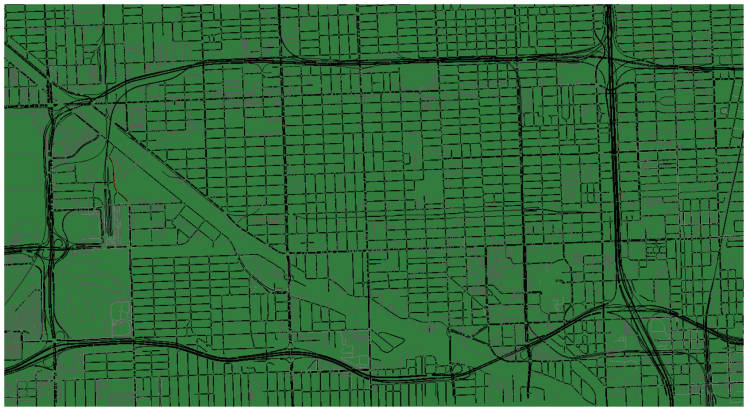
and the ability to add extensions such as TraCI for the access and control of simulation in real time. In this section, the simulation framework will be outlined. First, an overview of the SUMO network is presented, followed by the presentation of the chosen car-following model. Second, the analyses of the impact of certain parameters on the behavior of the AV in our simulation.

3.1 Simulation Environment Parameters

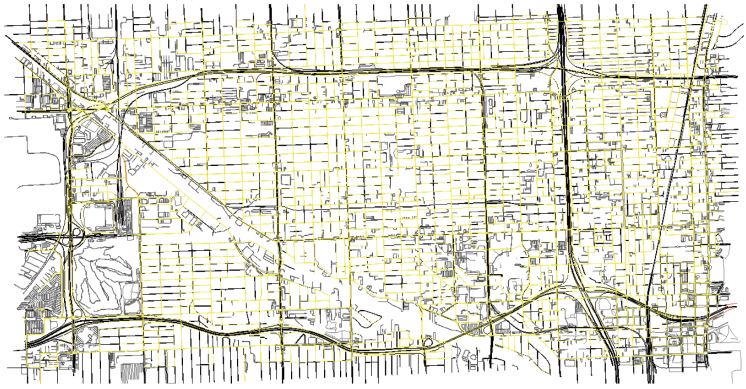
To obtain real-life road networks, we use SUMO's network importer a tool called "NETCONVERT"[3] which can read networks from OpenStreetMap [33]. OpenStreetMap is an open geographic database updated and maintained by a community of volunteers. Therefore, we get Miami City from the OpenStreetMap to import into our simulation, using NETCONVERT (see Fig. 1). The imported network includes several elements that can be useful for the continuity of this work. It comprises various transportation infrastructures, such as road signalization, various types of roads, waterways, bike paths, walkways, railways, and trams. Additionally, it integrates public services such as parking facilities, gas stations, and others, further enriching the dataset. The map to be chosen needs to have the following features: it should provide an accurate visual representation of the road network, distinguishing between major roads, arterial roads, and local streets. It also should present a diversity of road types, including Local Streets, Highways, State Roads, Boulevards, Avenues, and more. Traffic lights are typically indicated at major intersections, offering a visual representation of the simulation. The map should further showcase the diversity of areas, encompassing residential zones, commercial districts, industrial areas, and more. Thus, the map of Miami city has been selected to meet these requirements (see Fig. 1a). In general, the city is evident in its ability to convey road hierarchies, traffic flows, and the strategic location of traffic lights, making it easier to observe vehicle behavior during the simulation.

After obtaining the basic elements for the simulation, two additional components are necessary for its execution. First, generating a set of random trips for

[3] https://sumo.dlr.de/docs/netconvert.html.



(a) Imported network from OpenStreetMap



(b) Random possible trips

Fig. 1. Miami city in SUMO: real-life road networks

a given Miami network. SUMO provides a tool[4] for this purpose by uniformly selecting source and destination edges at random (see Fig. 1b). Consequently, initial parameters need to be established, including the desired number of vehicles, the minimum straight-line distance (in meters) between the start and end edges of a trip, the interval time for the trip, and so forth. In this work, two separate simulations were conducted: 2400 vehicles (800 for each driver's behavior), and 5400 vehicles (1800 for each driver's behavior). Second, choosing the optimal car-following model enables the regulation of the driver behavior, with the optimum parameters. The subsequent section will provide a detailed solution to this specific case.

[4] https://sumo.dlr.de/docs/Tools/Trip.html.

3.2 Car-Following Model

SUMO provides the capability to simulate various car-following models. These models refer to the way one vehicle follows another on a road. Some commonly used car-following models in SUMO include the Extended Intelligent Driver Model (EIDM) [19], Krauss Model [17], Adaptive Cruise Control (ACC) [26], etc. A comparison of these car-following models is presented in study [25]. In our simulations, we chose the EIDM [19] as a car-following model. It is based on many known model extensions of the IDM by Treiber and Kesting. The reason EIDM has been chosen is that it produces practical accelerations almost independently of the integration method, and has been continuously improved since its introduction. Compared to other models, EDIM is the model that focuses mainly on the realistic representation of human acceleration patterns, distinguishing it from other models. For instance, the Krauss model operates collision-free simulations, making it impossible to simulate dangerous behavior.

$$a_{ACC} = \begin{cases} a_{IDM} & \text{if } a_{IDM} \geq a_{CAH} \\ (1 - c_{ACC}) \times a_{IDM} & \\ + c_{ACC} \left[a_{CAH} + b \times \tanh\left(\frac{a_{IDM} - a_{CAH}}{b} \right) \right] & \text{otherwise} \end{cases} \tag{1}$$

In general, the IDM comprises two main equations [18]: the desired gap $s^{\times}_{n-1}$ and the acceleration a_{IDM}. It includes five important parameters: the desired time headway (T), maximum acceleration (a_{max}), desired deceleration (b), minimum gap (s_0), and acceleration exponent (δ). The EIDM model presented in [19], can not operate as a stand-alone model, it is used as an extension of the IDM. It serves as a basis for determining how the drivers' behaviors will be controlled at a later stage. The most important improvement of the EIDM compared to the IDM is the calculation of the acceleration a_{ACC}. It is based on a combination of two equations as shown in Eq. (1). The first one is the basic IDM acceleration a_{IDM}, and the second one is an operation called the Constant Acceleration Heuristic (CAH) a_{CAH}. The CAH acceleration takes the Leader acceleration into account, unlike the IDM equation. The acceleration a_{ACC} uses the coolness parameter noted c_{ACC}, with values between 0 and 1. c_{ACC} describes how "cool" a driver reacts when gaps are reduced. The Enhanced IDM improves the lane-changing behavior of the original IDM. The authors present other equations in their work (see [19] for more details). By understanding the model's operation, it becomes easy to determine a specific behavior level by manipulating precise parameters. We analyze this in the following section.

3.3 Impact of Sumo and Car Following Parameters on Driver Behavior

One of the objectives of our work is to control the behavior of vehicles. For this purpose, we have identified certain parameters of the EIDM model that we can manipulate: acceleration (**Accel**), deceleration (**Decel**), minimum gap

(**MinGap**), and maximum speed (**Speed$_{\mathbf{max}}$**). In addition, SUMO also presents its own parameters that frame the vehicles according to the used model. The important SUMO parameters that can be manipulated in our work are: Reaction time **Tau** represents the time taken for reactions before braking, responding to light signals, and acknowledging priorities at junctions before driving off. Custom speed factor **SpeedFactor** is the multiplier for lane speed limits and desired Max Speed, pushing the vehicle to surpass the speed limit.

For a stable and collision-free driving experience, we kept the following SUMO parameters (see Table 3): $Accel$ ="1.5m/s^2", $Decel$ = "4.00m/s^2", Tau ="1.00", and $MinGap$ ="2.00m", $SpeedFactor$ ="1.00", and $Speed_{max}$ Road max speed. We generate a dataset of the three different behaviors based on the same parameters described above. These elements are detailed in the following section.

4 AlertDang Driver Profiling Method

Nowadays, there are several technologies designed to assist drivers with the safe operation of a vehicle. The well-known technologies are Automatic Emergency Braking (AEB) [29], Advanced Driver Assistance Systems (ADAS) [28], Lane-Keeping Assist (LKA) [27], Blind Spot Detection (BSD) [30], Collision Avoidance Systems [31], Driver Monitoring Systems (DMS) [32], and Pedestrian Detection. These technologies deliver warnings in various ways. It employs different types of alerts such as visual warnings, audible warnings, haptic warnings, head-up displays, and in some cases, taking control of the vehicle. In this work, we use these warnings to determine the driver's behavior. We call it the **AlertDang** Driver Profiling method. This method focuses exclusively on warnings related to unsafe driving [20]. In other words, we consider only the signals related to aggressive behavior. For instance, the blind spot alert (meaning that another vehicle is in the blind spot of the driver) should not be used to identify the driver's behavior.

Since we do not have these technologies in the SUMO simulator. We generate five warnings that are inspired by the existing technology:

1. **The Speed limit respect "SpeedRespect":** It verifies whether the vehicle respects the maximum speed limit of the traffic lane (see [22]). We compare the vehicle's speed with the lane maximum speed indicated on the Miami map.
2. **The Front Safe Distance "SafeDist":** Where we verify the inter-vehicular distance (see [21]). The $SafeDist$ must be less than 2.5 m.
3. **SecureDist**: It is used to ensure that the appropriate safety distance on time is maintained with the vehicle ahead (see [21]). It is calculated by dividing the inter-vehicular distance by the speed of the vehicle to the rear. The $SecureDist$ must be less than 1.5 s.
4. **The Time To Collision(TTC) "TTCRespect":** It is the time remaining before a collision between a vehicle following and a vehicle in front (see [23]).

The TTC is calculated by dividing the inter-vehicle distance by the inter-vehicle speed. TTC is triggered if it is less than two seconds.

5. **The emergency Brake "EmergencyBrake":** It refers to measuring sudden braking (see [24]). The alert is triggered when the deceleration exceeds the expected value.

Bear in mind that from now on, these five warnings will be used to profile drivers, though additional warnings may be added. As the classic method for profiling, we assume that the more warnings a driver has, the more dangerous he is. However, the decision must be made based on the position and the distance driven. If we compare an AV with other vehicles passing from the same position or with different distances driven, the driver can have 20 warnings without being dangerous. This method can be improved by considering the number of warnings as a function of position and distance driven.

Table 3. The behavior type parameter in our SUMO simulation

Behavior Type	minGap(m)	accel(m/s^2)	decel(m/s^2)	tau	speedFactor
Per default (By SUMO)	2.00	1.50	4.00	1.00	1.00
Dangerous	1.00	4.00	3.00	0.7	1.20
Normal	2.50	2.00	3.00	2.00	1.20
Slow	2.5	1.00	3.00	3.00	0.90

Based on the PAYD needs, we choose these three levels of behavior based on the need for the PAYD service presented in Sect. 1. These behaviors are compared within a group of vehicles. To obtain the best representation of these behaviors, we choose the following parameters in Table 3. In the rest of this work, we have considered the following three classes of behavior:

- **Dangerous:** Presents a high number of warnings proportional to the driven distance.
- **Normal:** Indicates the average number of warnings for the vehicle within the group.
- **Slow:** Indicates a low number of warnings relative to the driven distance.

The parameters outlined in Sect. 3.3 directly influence the activation of our alerts. Through our experimentation and analysis, the following effects are observed:

- Tau considerably impacts the safety distance, which is logical because if a driver's reaction time is short. It takes time for them to decelerate, which declares the safe distance warning.
- $MinGap$ represents the desired distance a driver aims to maintain from the vehicle in front. This has a substantial impact on the values of $SecureDist$ and $SafeDist$ alerts. If the driver closely follows the vehicle, it is evident that they may disregard the alerts.

- Both *Accel* and *Decel* have a substantial impact on $SafeDist$ and a noticeable influence on other warnings such as $TTCRespect$, $SecureDist$, $EmergencyBrake$, and $SpeedRespect$.
- *Decel* and *accel* represent the combination that has a high impact on the $EmergencyBrake$ warning.
- *Accel* and $speed_{max}$ significantly affect the $SpeedRespect$ warning
- The impact of $speedFactor$ has a substantial impact on all warnings, including TTC warnings.

5 Experiments

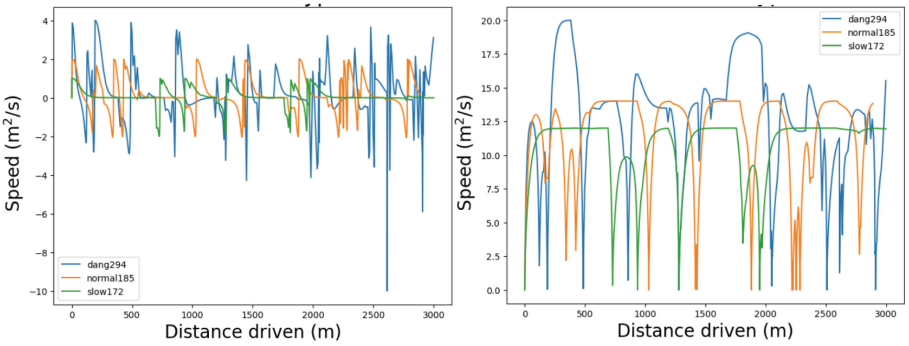
In our work, two simulations are performed: First, 2400 vehicles are launched in the simulation (800 for each driver's behavior), and we collected 916,601 samples: 363,112 slow samples, 298,449 normal samples, and 255,040 dangerous samples. Second, launching 5400 vehicles (1800 for each driver's behavior), we collected 2,101,756 samples: 786,642 slow samples, 692,044 normal samples, and 623,070 dangerous samples. In each of these two simulations, we collected two Datasets: The **In-Vehicle Dataset** and the **Warnings Dataset** (four Datasets are available). In this section, we present an analysis of these datasets. Then, we use only the warning datasets for ML classification. Keep in mind that the warning dataset collected in **Simulation 2** will be used to train the ML algorithms, and that the dataset from **Simulation 1** will be used for testing.

5.1 Dataset Analysis

In this section, we explore two datasets collected from our simulation. The first dataset comprises In-Vehicle Data and the second dataset includes warning signals. Based on these datasets, we present how the data is distributed and how the warning values differ between the driving behaviors collected.

In-Vehicle Dataset. The first dataset comprises In-Vehicle Data collected while running the vehicle in the SUMO simulator. For the moment, we are only collecting the elements we need: Label, Vehicle ID, Distance driven, Acceleration, Speed, Allowed Speed, and Gap. Bear in mind that we can collect many other In-Vehicle Data. We recommend that the reader check the SUMO documentation to know all the elements that can be collected.

To illustrate how the data is distributed, we present the following graphs. Figure 2 presents how each type of driver behaves during driving. Three vehicles are chosen randomly, with one vehicle from each type of driving behavior. The graph represents a part of the trajectory of these vehicles. The '*dang294*' highlighted in blue represents a driver exhibiting dangerous behavior. The '*normal185*' highlighted in orange, corresponds to normal driving behavior. Lastly, the '*slow172*' presented in green, indicates a driver with a slower and more cautious driving.



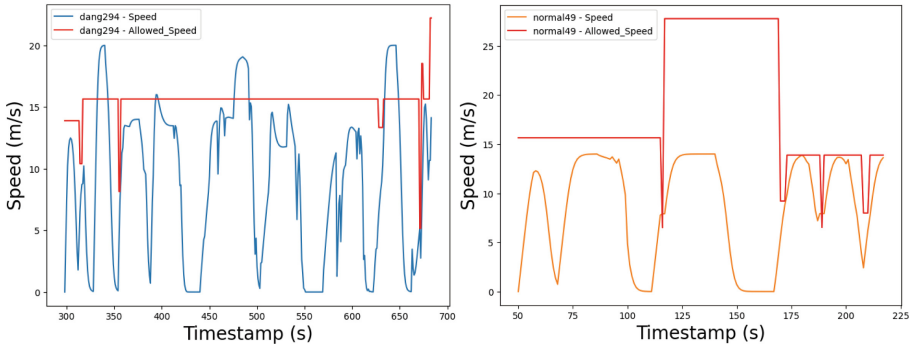
(a) The *Acc* of each type of behavior **(b)** The speed of each type of behavior

Fig. 2. The driver's behavior on the road

In Fig. 2a, sudden braking is observed from the vehicle *dang294*, where the acceleration drops to -10m/s^2 at one instance and -6m/s^2 at another (normal braking is -4m/s^2). This suggests that the vehicle behaves aggressively in one way or another. For instance, by analyzing the acceleration and deceleration curves for sharp peaks and valleys. We notice sudden spikes or dips in the curve which indicate aggressive behavior, like hard acceleration or harsh braking. In addition, while comparing the frequent and abrupt shifts between acceleration and deceleration, vehicle *slow172* exhibits the lowest peak-to-valley interval, whereas vehicle *dang294* has the largest peak-to-valley interval. This indicates that the vehicle *dang294* rapidly alternates between speeding and slowing down.

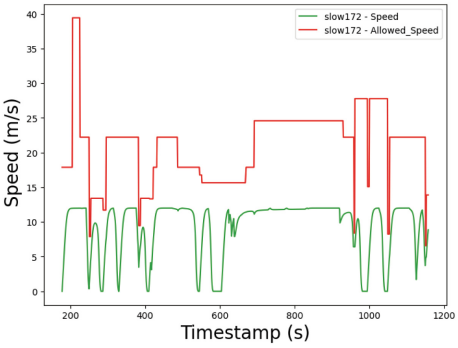
In Fig. 2b, we observe that, for the same period, the three vehicles exhibit different speed peaks. Vehicle *slow172* has the lowest peal, while *dang294* has the highest speed. This indicates an excessive speed from vehicle *dang294*, between the two moments when the vehicle stops. Concerning the speed limit, Fig. 3 shows that the vehicle *dang294* exceeds the speed limit on several occasions (see Fig. 3a). It also shows that the driver of the vehicle *slow172* is very cautious (see Fig. 3c), in contrast to the usual behavior of drivers of the vehicle *normal49* (see Fig. 3b).

Warnings Dataset. The second dataset includes warning data (see Sect. 4 for more details), illustrating the various alerts triggered during the real-time simulation. By computing the average alert for each label. In Fig. 4, we present how vehicles are increasingly alerted when they are driving. The more dangerous the vehicles, the more alerts they receive. They can receive more than 3000 alerts, which is normal because a warning does not necessarily imply a collision. In Simulation 2, 78 collisions were detected among 1800 vehicles, all associated with the dangerous vehicle. Slow vehicles also receive alerts, as shown in Fig. 4b. They mainly receive the TTC alert because of the low *Tau* parameter, which is designed to model the minimum waiting time desired by the driver (in seconds). They take the time to decide whether to accelerate or brake.



(a) Dangerous behavior

(b) Normal behavior



(c) Slow behavior

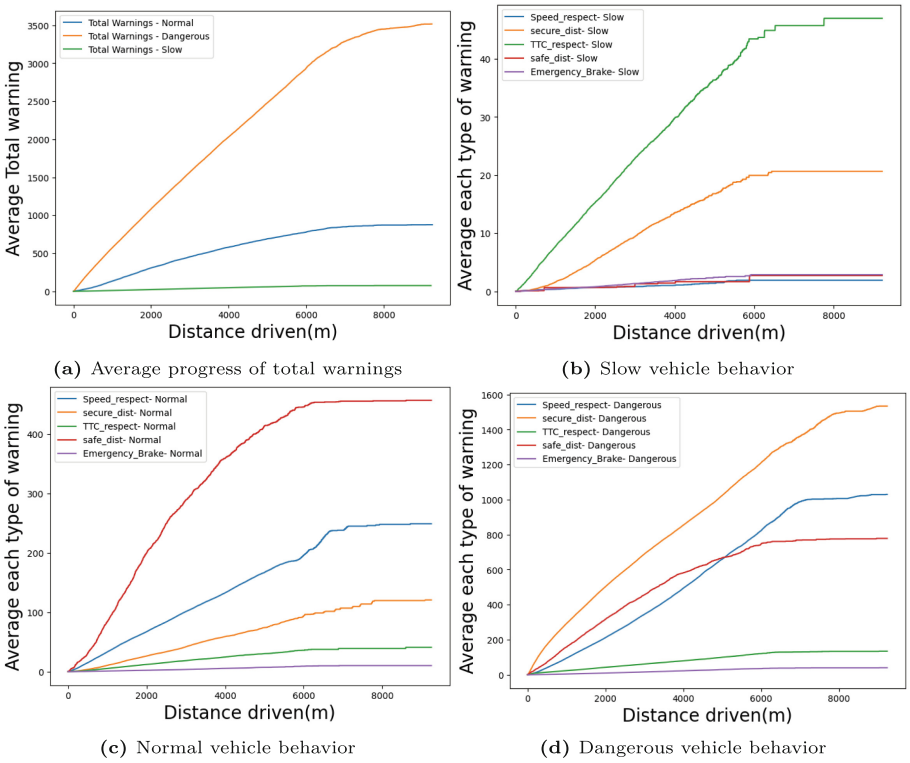
Fig. 3. Behavior of each type of driver in terms of the speed limit, depending on the route taken

5.2 ML Classification on the Warning Dataset

The motivation for the ML classification in the paper is that we want to assess the accuracy of the models trained and tested on our datasets.

Only the **Warning Dataset** will be used for classification. The purpose of this classification algorithm is to identify driver behavior, such as in the case of the PAYD service. To classify drivers accurately, we posit that a dangerous driver cannot be aggressive all the time. Therefore, we propose accumulating warnings for each X-timestamp. For example, we choose the total accumulated warnings for each vehicle ID every 50 timestamps instead of every timestamp. From now, two datasets are extracted from the warnings dataset: **"DS 1"** and **"DS 2"** which refers to a dataset with warnings for 1 timestamp, and a dataset with warnings for 50 timestamps, respectively.

Both datasets share identical features, encompassing Label, Distance_driven, Speed_respect, Secure_dist, TTC_respect, Safe_dist, Emergency_Brake, and Total. To reiterate, this study comprises two simulations: firstly, with 5400 vehicles launched (1800 for each driver's behavior), and secondly, with 2400 vehicles



(a) Average progress of total warnings

(b) Slow vehicle behavior

(c) Normal vehicle behavior

(d) Dangerous vehicle behavior

Fig. 4. Average progress of vehicle warnings based on distance traveled

launched (800 for each driver's behavior). The intent is to utilize one for training ML models and the other for testing ML models. To indicate the sample size, the simulation with 5400 vehicles launched produced 2,101,756 samples, while the simulation with 2400 vehicles launched produced 916 601 samples. The accumulation operation presented above is applied to the dataset collected from both simulations. After transformation to accumulate warnings for each 50 timestamps, simulation 1 produced 47 629 samples and the second 20 472 samples. In the following, we briefly describe the ML algorithm chosen and then present the different results obtained for the two types of datasets.

ML Algorithms Description. In this phase, we consider the algorithms that show good performance in our tests, e.g., the Gradient Boosting Decision Trees (GBDT), K nearest neighbors algorithm (KNN), multi-layer perceptron (MLP), and SVM classifier. In the following, we briefly describe these ML algorithms that we are using, beginning with GBDT, which is an ensemble learning technique that builds a strong predictive model by combining multiple weak models, typically decision trees. It operates sequentially, with each new tree correcting errors made by the previous ones. Gradient boosting is used to minimize a loss

function, making it well-suited for both regression and classification tasks. Moving on to KNN, is a simple and versatile classification algorithm that classifies data points based on the majority class of their nearest neighbors. It works by measuring the distance between a data point and its neighbors, using a user-defined value of 'k' to determine how many neighbors to consider. Additionally, MLP is a type of artificial neural network with multiple layers of nodes (neurons). It is trained using a process called backpropagation, where the network learns from the error in its predictions and adjusts the weights of connections between neurons. Lastly, the SVM is a supervised ML algorithm used for classification and regression tasks. It works by finding a hyperplane that best separates data points of different classes in a high-dimensional space. In other words, SVM aims to maximize the margin between classes, making it robust to outliers and effective in high-dimensional spaces.

5.3 Classification Result

The ML algorithms used in our experiment are imported from the Scikit-learn library coded in Python. We use the default setting presented by the library, apart from the one defined in this subsection.

Table 4. Accuracy of ML algorithms. Training on the simulation 1 dataset (5400 vehicles) and testing on the simulation 2 dataset (2400 vehicles). DS 1 refers to a dataset with warnings for 1 timestamp. DS 2 refers to a dataset with total warnings for every 20 timestamps.

Dataset type	GBDT		KNN		MLP		SVM	
	Training	Testing	Training	Testing	Training	Testing	Training	Testing
DS 1	89%	80%	89%	74%	88%	80%	82%	74%
DS 2	88%	81%	89%	74%	84%	80%	79%	73%

After running these algorithms on the collected Warnings Dataset from the SUMO simulation, we present the accuracy scores on the training and testing datasets (see Table 4). Additionally, we present the confusion matrix to evaluate the model's performance further, showcasing an instance in a predicted value while the column represents the actual value (see the matrix in Fig. 5). The diagonals represent correct predictions, while off-diagonal elements represent misclassifications. The input parameters and results are described as follows:

– The GBDT is initialized with specific parameters, including the number of estimators = 10, the learning rate = 1.0, and the maximum depth = 3. For DS 1, the accuracy scores on the training and testing datasets are 89% and 80%, respectively. The prediction is illustrated by the confusion matrix in Fig. 5a. As for DS 2, it achieves an accuracy of 88% on the training dataset and 81% on the test dataset. The prediction is illustrated by the confusion matrix in Fig. 5a and b.

- The KNN classifier is instantiated and trained with k=3 and L1 distance (Manhattan distance). The accuracy scores on both the training and testing datasets 1 are 89% and 74%, respectively. Additionally, a confusion matrix is presented in5c. DS 2 presents an accuracy of 89% on the training dataset and 74% on the test dataset. The confusion matrices are in Fig. 5c, d.
- The MLP Classifier is set up with specific parameters, including the structure of the hidden layers (4, 8, 4), the maximum number of iterations = 100, the activation function for the hidden layer 'relu', the solver for weight optimization 'adam', and the learning rate schedule for weight updates equal to 0.001. The accuracy scores on both the training and testing datasets are 88% and 80%, respectively. For DS 2, the accuracy scores on the training and testing datasets are 84% and 80%, respectively. The confusion matrix for both datasets is presented in Fig. 5e and f.
- Since we have numerous samples. We choose a Stochastic Gradient Descent (SGD) Classifier with an SVM. The implementation includes feature scaling through StandardScaler and utilizes a pipeline for seamless processing. For DS 1, the accuracy scores for both training and testing data are 82% and 74%, respectively. As for DS 2, it presents an accuracy of 79% on the training dataset and 73% on the test dataset. The confusion matrices are presented in Fig. 5g and h.

$$\begin{bmatrix} 353992 & 9059 & 61 \\ 126354 & 170333 & 1762 \\ 5702 & 39923 & 209415 \end{bmatrix}$$

(a) GBDT- DS 1

$$\begin{bmatrix} 7762 & 208 & 1 \\ 2642 & 3991 & 61 \\ 91 & 813 & 4903 \end{bmatrix}$$

(b) GBDT- DS 2

$$\begin{bmatrix} 281230 & 81165 & 717 \\ 106491 & 185912 & 6046 \\ 4800 & 37555 & 212685 \end{bmatrix}$$

(c) KNN- DS 1

$$\begin{bmatrix} 6458 & 1464 & 49 \\ 2487 & 3970 & 237 \\ 139 & 747 & 4921 \end{bmatrix}$$

(d) KNN- DS 2

$$\begin{bmatrix} 353669 & 9395 & 48 \\ 128673 & 169263 & 513 \\ 4881 & 38629 & 211530 \end{bmatrix}$$

(e) MLP-DS 1

$$\begin{bmatrix} 7683 & 288 & 0 \\ 2966 & 3708 & 20 \\ 146 & 868 & 4793 \end{bmatrix}$$

(f) MLP-DS 2

$$\begin{bmatrix} 281230 & 81165 & 717 \\ 106491 & 185912 & 6046 \\ 4800 & 37555 & 212685 \end{bmatrix}$$

(g) SVM-DS 1

$$\begin{bmatrix} 7700 & 270 & 1 \\ 2732 & 3962 & 0 \\ 168 & 1046 & 4593 \end{bmatrix}$$

(h) SVM-DS 2

Fig. 5. The confusion matrix

For all the predictions obtained by these algorithms (see Figs. 4 and 5), the accuracy and the confusion matrices reveal certain information about the input databases. GBDT achieves high accuracy, especially on DS 1, indicating good

performance. KNN has moderate accuracy, struggling more on DS 2. MLP performs well with high accuracy. SVM shows a similar performance to GBDT and KNN. However, the predictions for normal drivers are always very close to those for slow drivers. This slight convergence is explained by the fact that slow drivers also have warnings in our simulation.

6 Conclusion

In summary, we have introduced a novel SUMO simulation and a dataset designed to train an ML model to classify drivers. This approach can reduce the need to rely on real datasets for training an ML model. These data may contain sensitive information on drivers. This work focuses on three main challenges: First, creating an environmental simulation that accurately models vehicle behavior to replicate real-world road conditions on a large scale. Second, collecting specific data for driver behavior profiling. Third, ensuring the reliability of the dataset for training any ML model aimed at classifying driver behavior. Our emphasis lies in constructing a SUMO simulation to facilitate large-scale tests essential for demonstrating capabilities across diverse conditions. We aim to conduct these tests in a controlled and reproducible manner, bringing the simulation closer to real-world scenarios. In addition, the proposal of a means to separate the three levels of behavior, slow, normal, and dangerous. The dataset presented in this article is readily available for use in ML algorithms or neural networks. Replicating the experiments is simple, using the code provided, which can be accessed on the dedicated GitHub page.

The behavior style is determined mathematically by adjusting the parameters of the IDM model. This eliminates the need to rely on subjective assessments of aggressive driving by each participant in the "Sim Race" experiment. Indeed, such assessments may vary from one individual to another. Additionally, we utilize expanded map options featuring diverse road types, along with a scalable number of drivers and distances covered. Furthermore, we introduce an original Driver Profiling method known as AlertDang. We cover all the issues outlined above. To our knowledge, no dataset on driver behavior has been provided by a full simulator in the literature.

For future work in this direction, our SUMO simulation can be improved in many ways. For instance, We plan to introduce a default behavior for drivers who adopt aggressive behavior for only part of the trajectory, instead of the entire trajectory. Additionally, we can add other types of warnings to improve the datasets. Furthermore, the data set will be used to classify drivers in the PAYD insurance use case. This service is based on the assumption that the more dangerous the driver is on the road, the more he or she will have to pay. For this use case, sensitive driver data will be processed, and we aim to preserve the privacy of these drivers while maintaining PAYD services. We look forward to merging federated learning with PETs such as differential privacy and homomorphic encryption. Thus, the datasets provided by this work will help us train ML algorithms in a privacy-friendly manner.

References

1. Global Status Report on Road Safety 2023. World Health Organization, Geneva (2023). License: CC BY-NC-SA 3.0 IGO
2. Chah, B., Lombard, A., Bkakria, A., Yaich, R., Abbas-Turki, A., Galland, S.: Privacy threat analysis for connected and autonomous vehicles. Procedia Comput. Sci. **210**, 36–44 (2022)
3. Chah, B., Lombard, A., Bkakriab, A., Yaich, R., Abbas-Turkia, A.: Exploring Privacy Threats in Connected and Autonomous Vehicles: An Analysis (2023)
4. Li, Q., Wu, Z., Wen, Z., He, B.: Privacy-preserving gradient boosting decision trees. In: Proceedings of the AAAI Conference on Artificial Intelligence (2020)
5. Zhao, L., Ni, L., Hu, S., Chen, Y., Zhou, P., Xiao, F., Wu, L.: Inprivate digging: enabling tree-based distributed data mining with differential privacy. In: IEEE INFOCOM 2018-IEEE Conference on Computer Communications (2018)
6. Chah, B., Lombard, A., Bkakria, A., Abbas-Turki, A., Yaich, R.: H3PC: enhanced security and privacy-preserving platoon construction based on fully homomorphic encryption. In: 26th IEEE International Conference on Intelligent Transportation Systems (ITSC) 2023
7. Paul, J., Annamalai, M.S.M.S., Ming, W., Al Badawi, A., Veeravalli, B., Aung, K.M.M.: Privacy-preserving collective learning with homomorphic encryption. IEEE Access **9**, 132084–132096 (2021)
8. Kwak, B.I., Woo, J., Kim, H.K.: Know your master: driver profiling-based anti-theft method. In: 2016 14th Annual Conference on Privacy, Security and Trust (PST), pp. 211-218. IEEE (2016)
9. Enev, M., Takakuwa, A., Koscher, K., Kohno, T.: Automobile driver fingerprinting. Proc. Priv. Enhancing Technol. **2016**(1), 34–50 (2016)
10. EU General Data Protection Regulation (2016). http://www.eugdpr.org/
11. Cojocaru, I., Popescu, P.S.: Building a driving behavior dataset. In RoCHI, pp. 101–107 (2022)
12. Zhang, X., Zhao, X., Rong, J.: A study of individual characteristics of driving behavior based on hidden Markov model. Sens. Transducers **167**(3) (2014)
13. Meng, X., Lee, K.K., Xu, Y.: Human driving behavior recognition based on hidden Markov models. In: 2006 IEEE International Conference on Robotics and Biomimetics, pp. 274–279. IEEE (2006)
14. Wakita, T., Ozawa, K., Miyajima, C., Igarashi, K., Itou, K., Takeda, K., Itakura, F.: Driver identification using driving behavior signals. IEICE Trans. Inf. Syst. **89**(3), 1188–1194 (2006)
15. Lopez, P.A., Behrisch, M., Bieker-Walz, L., Erdmann, J., Flötteröd, Y. P., Hilbrich, R., Wießner, E.: Microscopic traffic simulation using SUMO. In: 2018 21st ITSC IEEE (2018)
16. Wegener, A., Piorkowski, M., et al.: Traci: an interface for coupling road traffic and network simulators. In: Proceedings of the 11th Communications and Networking Simulation Symposium, 2008
17. Krauss, S.: Modélisation microscopique du flux de trafic : étude de la dynamique des véhicules sans collision (1998)
18. Treiber, M., Hennecke, A., Helbing, D.: Congested traffic states in empirical observations and microscopic simulations. Phys. Rev. E **62**(2), 1805 (2000)
19. Salles, D., Kaufmann, S., Reuss, H.C.: Extending the intelligent driver model in SUMO and verifying the drive off trajectories with aerial measurements. In: SUMO Conference Proceedings, vol. 1, pp. 1–25 (2020)

20. Global Status Report on Road Safety 2023. World Health Organization 2023. https://www.who.int/publications/i/item/9789240086517
21. Michael, P.G., Leeming, F.C., Dwyer, W.O.: Headway on urban streets: observational data and an intervention to decrease tailgating. Transport. Res. F: Traffic Psychol. Behav. **3**(2), 55–64 (2000)
22. Høye, A.: Speeding and impaired driving in fatal crashes-Results from in-depth investigations. Traffic Inj. Prev. **21**(7), 425–430 (2020)
23. Wali, B., Khattak, A.J., Karnowski, T.: The relationship between driving volatility in time to collision and crash-injury severity in a naturalistic driving environment. Anal. Methods Accid. Res. **28**, 100136 (2020)
24. Tan, H., Zhao, F., Hao, H., Liu, Z., Amer, A.A., Babiker, H.: Automatic emergency braking (AEB) system impact on fatality and injury reduction in China. Int. J. Environ. Res. Public Health **17**(3), 917 (2020)
25. Schrader, M., Al Abdraboh, M., Bittle, J.: Comparing measured driver behavior distributions to results from car-following models using SUMO and real-world vehicle trajectories from radar: SUMO default versus radar-measured CF model parameters. In: SUMO Conference Proceedings (2023)
26. Serafin, C.: Driver Preferences and Usability of Adjustable Distance Controls for an Adaptive Cruise Control (ACC) System. Technical Report, National Highway Traffic Safety Administration (1996)
27. Utriainen, R., Pöllänen, M., Liimatainen, H.: The safety potential of lane keeping assistance and possible actions to improve the potential. IEEE Trans. Intell. Veh. **5**(4), 556–564 (2020)
28. Ziebinski, A., Cupek, R., Grzechca, D., Chruszczyk, L.: Review of advanced driver assistance systems (ADAS). In: AIP Conference Proceedings, vol. 1906, No. 1. AIP Publishing (2017)
29. Coelingh, E., Eidehall, A., Bengtsson, M.: Collision warning with full auto brake and pedestrian detection-a practical example of automatic emergency braking. In: 13th International IEEE Conference on Intelligent Transportation Systems, pp. 155–160. IEEE (2010)
30. Kwon, D., Park, S., Baek, S., Malaiya, R.K., Yoon, G., Ryu, J.T.: A study on development of the blind spot detection system for the IoT-based smart connected car. In: ICCE (2018)
31. ongcai, Z., Hongwei, X., Kexin, Y.: Autonomous collision avoidance system in a multi-ship environment based on proximal policy optimization method. Ocean Eng. **272**, 113779 (2023)
32. Mohammed, K., Abdelhafid, M., Kamal, K., Ismail, N., Ilias, A.: Intelligent driver monitoring system: an Internet of Things-based system for tracking and identifying the driving behavior. Comput. Stan. Interfaces **84**, 103704 (2023)
33. OpenStreetMap website. http://www.openstreetmap.org/