RESEARCH ARTICLE

Deriving Environmental Risk Profiles for Autonomous Vehicles From Simulated Trips

JOHN ANIH [1,2]**, SARVESH KOLEKAR**[1]**, TOOSKA DARGAHI**[2,3]**, MEISAM BABAIE**[2,4]**, MOHAMAD SARAEE**[2]**, AND JACK WETHERELL**[1]

[1]Humn.ai Ltd., W6 7NL London, U.K.
[2]School of Science, Engineering and Environment, University of Salford, M5 4WT Salford, U.K.
[3]Computing and Mathematics Department, Manchester Metropolitan University, M15 6BH Manchester, U.K.
[4]School of Mechanical Engineering, University of Leeds, LS2 9JT Leeds, U.K.

Corresponding author: John Anih (john@humn.ai)

ABSTRACT The commercial adoption of Autonomous Vehicles (AVs) and the positive impact they are expected to have on traffic safety depends on appropriate insurance products due to the high potential losses. A significant proportion of these losses are expected to occur from the out-of-distribution risks which arise from situations outside the AV's training experience. Traditional vehicle insurance products (for human-driven vehicles) rely on large data sets of drivers' background and historical incidents. However, the lack of such datasets for AVs makes it imperative to exploit the ability to deploy AVs in simulated environments. In this paper, the data collected by deploying Autonomous Driving Systems (ADSs) in simulated environments is used to develop models to answer two questions: (1) *how risky a road section is for an AV to drive?* and (2) *how does the risk profile vary with different (SAE levels) of ADSs?* A simulation pipeline was built on the CARLA (Car Learning to Act): an open-source simulator for autonomous driving research. The environment was specified using parameters such as *weather*, *lighting*, *traffic density*, *traffic flow*, *no. of lanes*, etc. A metric - *risk factor* was defined as a combination of *harsh accelerations/braking*, *inverse Time to Collision*, and *inverse Time Headway* to capture the crashes and near-crashes. To assess the difference between ADSs, two ADSs: OpenPilot (Level 2/3) and Pylot (Level 4) were implemented in the simulator. The results (from data and model predictions) show that the trends in the relation between the environment *features* and *risk factor* for an AV are similar to those observed for human drivers (e.g., risk increases with *traffic flow*). The models also showed that *junctions* were a *risk hot-spot* for both ADSs. The feature importance of the model revealed that the Level 2/3 ADS is more sensitive to *no. of lanes* and the Level 4 ADS is sensitive to *traffic flow*. Such differences in feature importance provide valuable insights into the risk characteristics of different ADSs. In the future, this base model will be extended to include other features (other than the environment), e.g., take over requests, and also address the deficiencies of the current simulation data in terms of insensitivity to *weather* and *lighting*.

INDEX TERMS Autonomous vehicle, insurance, risk modelling, simulation, traffic flow, traffic density.

I. INTRODUCTION

Autonomous vehicles (AVs) are a widely anticipated driving technology, that are undergoing rapid development and

The associate editor coordinating the review of this manuscript and approving it for publication was Li Zhang.

expansion due to the recent surge in advancements in deep learning, behavioural cloning and computer vision research, leading to current and future commercial products that look to disrupt industries surrounding transport and logistics [1]. These self-driving technologies look to change the way humans commute, in particular, aiming to reduce the number

of road accidents faced by road users, which is a significant cost to public health [2], [3]. Despite these benefits, the large scale adoption of AVs has been very slow. One of the hurdles slowing down the commercialisation of AVs is their insurance. Apart from the legal aspects of *who bears the responsibility of the crash* [4], insurers do not have sufficiently good models that can predict the risk profile of an AV [5].

Traditional insurance products for human-driven vehicles use historical accident and claims data to build experience and exposure-based risk models [6]. However, these datasets are not useful for modelling the risk profile of an AV since the factors that contribute to risk in an AV are different from those that contribute to human driving [7]. For example, fatigue is a major factor for human drivers [8], [9] which is non existent for AVs. On the other hand, humans are good at generalising their driving skills to different geographical locations. This is not the case for AVs that need to be trained with data from specific regions with well defined Operational Design Domains (ODD) [10]. Hence, an alternative approach to quantify the risk profile of an AV is needed.

However, the risk profile of an AV can be complex to model since it is affected by several factors such as the environment its driving in (e.g., road type, traffic density, weather conditions), the human-machine interaction (e.g., Take Over Request procedures, timings), the cyber-security aspects, etc. In this paper an initial step towards formulating a risk profile for AVs is taken, and hence the scope of this paper has been limited to only the environment risk (road type, traffic density, weather conditions). Additionally, the approaches that are suited from an insurers perspective (who is required to insure different types of AVs, Automated Driving Systems (ADS) Level 1 -Level 5 [11]), are explored and developed.

In summary, the focus of this paper is on answering the questions: *how risky a road section is for an AV to drive*? and *how does the risk profile vary with different (SAE levels) of ADSs*?

A. LITERATURE

Several researchers have worked on quantifying the risk of AVs and these approaches found in literature could be broadly classified into two types: (1) The Fault Tree Analysis (FTA) and (2) Simulation of ADS in a virtual environment.

In the Fault Tree Analysis approach, the AV is first disassembled into its components and then a fault tree model is developed for each system (e.g., Camera, Lidar, GPS, Mechanical components) [12]. Failure probabilities to each component are then assigned using estimates from literature and by analysing publicly available datasets. This method provides a comprehensive risk analysis of the AV and provides insights for OEMs into the vulnerable subsystems that could be improved. However, it requires a detailed and systematic evaluation of each subsystem to assign failure probabilities which is not feasible for an insurer who aims to insure different AVs with a wide range of architectures and components.

The second approach of simulating AVs in a virtual environment has become popular in the recent years, especially with the advancements in simulators that render realistic environments [13], [14], [15]. For example OEMs such as Waymo, Motional, Cruise, etc. use simulations to replay the data recorded from the real world and evaluate their versions of software stack [16]. Such *what-if* [17] kind of simulations are beneficial for OEMs to investigate if a crash that actually occurred in the real world could have been avoided by their AV. Simulators have also been used to test the real-time capabilities of the AV software stack by emulating hardware [18]. However, such simulations although beneficial, do not provide a complete picture of the risk profile of an AV from an insurers perspective.

Simulating AVs in a virtual world is a powerful technique but is challenging due to the large variety and continuum of scenes that occur in naturalistic driving. To overcome this hurdle, several researchers have taken the scenario based approach to quantify risk for AVs [19], [20]. In this approach, the environment is parameterized (e.g., weather = clear, raining, cloudy, road type = highway, rural, urban). The AV algorithm is then tested in a virtual environment where all possible combinations of this parameter values are simulated. For example, De Gelder et al. [21] parameterize the environment, and calculate the exposure and severity of the AV for different scenario types (cut-inn during lane change etc.). This approach is the most relevant for the question that is being answered in this paper. However, from an insurers perspective, it is very difficult obtain the 'scenario type' when the risk is analysed in real time while the AV is driving on the road. Hence, in this paper the first question: *how risky a road section is for an AV to drive?* is answered by borrowing the *parametrization of the environment* concept from the above papers. However, we did not define specific scenario types. Instead, a large number simulations are run, assuming that all possible combinations of scenario types will occur in a given set of environments. A similar procedure was adopted by Norden et al. [22] in a study where they evaluate risk exposure of a single ADS.

Answering the second question: *how does the risk profile vary with different (SAE levels) of ADSs*, is challenging due to the fact that these ADSs are complex and rarely open-source. Additionally, it is challenging to integrate them into identical simulation environments for comparative testing. To the best of our knowledge, there are no papers that performed a one-on-one comparison of ADSs in similar settings. Hence, in this paper, we will be addressing this gap by implementing a Level 2/3 and a Level 4 ADS in a virtual environment and assessing their risk profiles.

The remainder of the paper is organised as follows:

In the Methods Section II, we describe our simulation pipeline. More specifically, how the environment is parameterized (into weather, lighting, no of lanes, traffic, etc.) and what values each parameter can assume (e.g., weather = {clear, raining, cloudy}). Next, we define the metrics used to quantify *how risky a road section was* and describe the two

different ADSs (level 2 and 4) that were simulated. In the Data Analysis Section III, we analyse the dataset generated from simulations and provide a summary of the feature (environment) distributions and label (risk factor) distributions. In the Results Section IV we train and test a model for each of the ADSs (level 2 and 4). The feature importance analysis, comparison of the predictions of the model to data from literature, and the comparison of the difference in risk profiles of the two ADSs is discussed in the discussion section. Finally, we draw conclusions and discussion from the results (Section V) and provide future steps that could help improve and simplify the process of generating risk profile for ADSs from an insurers perspective.

II. METHODS

Since the approach that we chose to quantify the risk was based on simulating AVs in a virtual world, the main task in this paper was to setup and develop a simulation pipeline (Fig. 2). The simulation pipeline consists of four stages: (1) AV simulation, (2) AV Post-processing, (3) AV Modelling, and (4) AV Application.

A. AV SIMULATION

The main component of the simulation is the simulator itself. In this study, we chose to use CARLA [14] due to its flexibility, realism (in rendering visuals and physics), and the availability of open-source support for integrating with ADSs such as OpenPilot (Level 2/3) [23] and Pylot (Level 4 ADS) [24]. The simulation involved two steps: (1) Simulating the environment, and (2) Simulating the Automated Driving Systems (ADSs).

1) SIMULATE THE ENVIRONMENT

To answer the first research question: *how risky a road section is for an AV to drive?*, the approach was to parameterize the environment and then generate the combinations of the different parameter values to define a large number of simulation scenes (Fig. 1). The parameters and their corresponding values are shown in Table 1 (details in Supplementary material Section I).

TABLE 1. Parameters defining the CARLA simulation.

Parameters	Values
Lighting	{morning, daylight, night}
Weather	{clear, cloudy, raining, fog}
Number of pedestrians	{10, 20, 30}
Number of vehicles (traffic)	{50, 100, 150, 200, 250, 300}
Traffic Aggression	{0=calm, 1=normal, 2=aggressive}
*Road geometry*a	{1-124}

2) SIMULATE THE AUTOMATED DRIVING SYSTEMS (ADSs)

To answer the second research question: *how does the risk profile vary with different (SAE levels) of ADSs*, two ADSs

aSpecifies a combination of a CARLA map and a start point realising different road types (highway, intersections, curvy rural roads).

were integrated into the CARLA simulation environment. Their behaviours tested in different scenarios and assessed for risky behaviour.

1) OpenPilot [23] from Comma.ai (Level 2/3 ADS).
2) Pylot [24] from the Erdos Project (Level 4 ADS).

The above two ADSs were chosen because they are open-source and provide us with the two different ADS levels we are interested in. It was also important that these ADS were being widely used in (on-road) vehicles and simulators. For example, OpenPilot can be installed in most modern vehicles and driven on-road, while Pylot being a powerful modular ADS has been widely used in AV research. The advantage of open-source ADSs is that several researchers can evaluate these systems. For example, Chen et al. [25] re-implemented OpenPilot and evaluated its capabilities as a level 2 ADS using several on-road and simulated datasets.

OpenPilot being a level 2/3 ADS, has limitations compared to Pylot which is a Level 4 ADS. Hence, directly comparing them would result in the trivial conclusion that a Level 4 ADS is safer than the Level 2/3 ADS. Since this paper is aimed at assessing the effect of the environment on the ADSs, we assume that the ADSs will be used while respecting their Operation Design Domain (ODD). Violations of the ODD are not dealt in this paper, and will be a separate module in the final insurance product. The ODD for Pylot was not restrictive (as per their documentation) and all the road geometries (1-124) could be used. On the other hand, OpenPilot (version 0.8.6) had 2 main restrictions which needed to be addressed: (1) It could not modulate the target speed according to the speed limit (Comma.ai have now added this feature is later versions). Hence, we implemented a module that acquired the speed limit from CARLA and set the target speed for OpenPilot thus mimicking a driver in charge. (2) OpenPilot was not intended to be used in sharp corners. Hence, all the road geometries where the OpenPilot crashed on its own accord in clear weather, daylight, without any traffic or pedestrians on the road were discarded for OpenPilot (details in Supplementary material Section II-A).

Each simulation was intended to last 150 seconds and the output of the AV Simulation was a raw time series dataset which included the position, velocity, bounding boxes of every actor (vehicle and pedestrian) in the scene (Fig. 2). The traffic interacting with the ADSs was spawned around the AV and was controlled by the default *Autopilot* provided in CARLA.

B. AV POST-PROCESSING

Practically, we intend to train a model and use it for predicting a risk score while an AV is driving in the *real-world*. The data from the *real-world* is usually obtained from traffic and weather APIs (e.g., OSM). Hence, in the AV Post-processing stage we convert the raw output of the AV Simulation stage into data of a similar format that can be obtained from the APIs. But, before we could post-process the data, data cleaning was performed which involved discarding the simulations
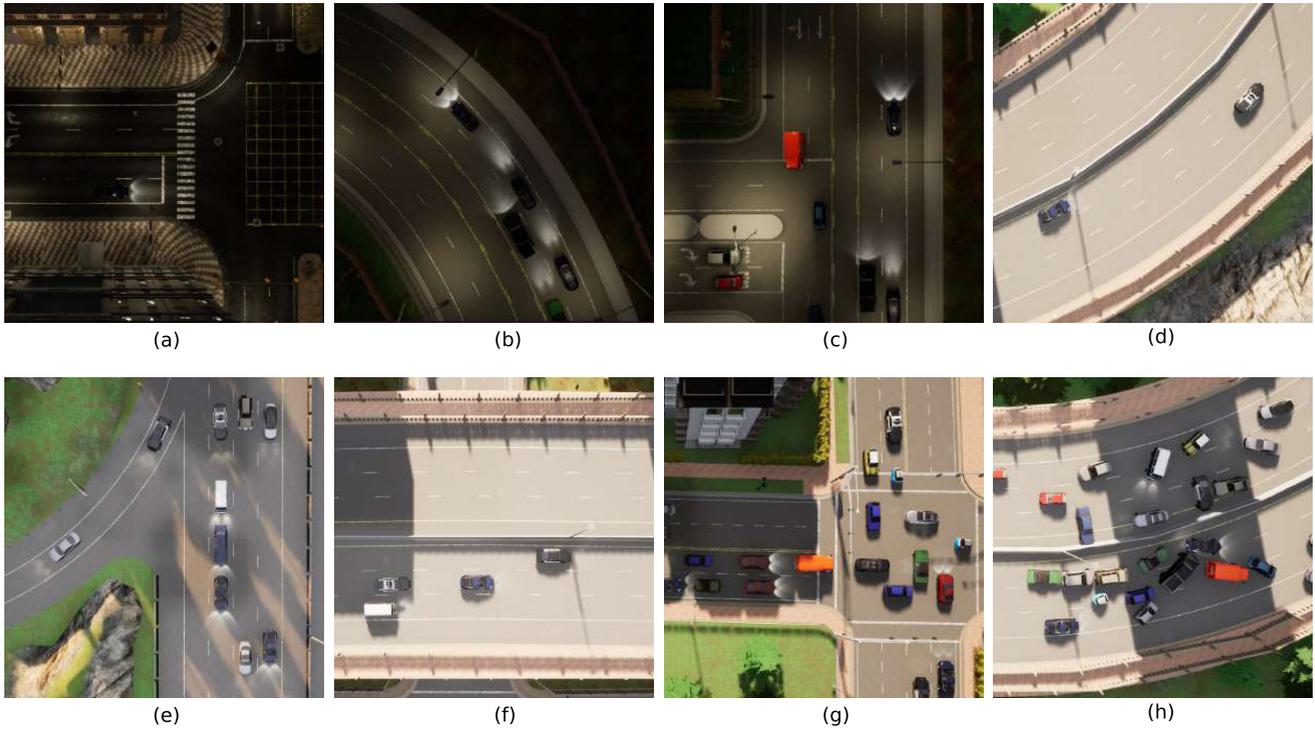
FIGURE 1. Traffic simulations: Screenshots of the simulations in different road, traffic and weather conditions. (a) Urban intersection at night with sparse traffic and hence low interactions. (b) Curved section of a highway at night with traffic surrounding the AV, leading to high interactions. (c) A T-junction merging onto a highway at night presents a challenging situation for the AV. (d) Curved section of highway in daylight with sparse traffic is one of the easier scenarios for the AV. (e) An off-ramp on a highway section in daylight presents challenges to the AV since vehicles are switching lanes to get off the highway. (f) A straight section of a highway with surrounding vehicles. (g) An intersection in a urban area with medium-to-dense traffic conditions. (h) Some simulations failed due to computational overload and were discarded from the dataset.

that ended before the intended 150 seconds (often indicated some problem with computational resources).

1) POST-PROCESSED ENVIRONMENT METRICS (FEATURES)

The parameters related to weather and lighting can be directly obtained from weather APIs and hence, did not need any post-processing. Parameters related to vehicles, pedestrians, and the road geometry were post-processed to obtain the metrics listed in Table 2:

TABLE 2. Post-processed metrics.

Post-processed metrics	Description
No. of lanes	No. of lanes on the current road section
Road curvature	Curvature of the road at current location
Traffic density	No. of vehicles per kilometer on the current road section
Traffic average speed	Mean speed of all the vehicles on the current road section
Walker density	No. of pedestrians per kilometer on the current road section
Is junction (True/False)	If a junction was present on the current road section

All the above metrics that are mentioned (*features*) define the environment, and as can be seen from the definition of the metrics above, a crucial part of the post-processing was the *current road section*. The current road section was obtained

from the CARLA *World object*. However, in many cases, especially in curves, the road sections defined by CARLA are small and hence were aggregated together to form a road section with a minimum length of 50 m (details in Supplementary material Section III).

2) POST-PROCESSED RISK METRICS (LABELS)

Since our task was to quantify the risk an AV finds itself exposed to, we post-processed the raw AV simulation data to derive several metrics that literature suggests correlate with risk.

The first metric was the total (lateral and longitudinal) acceleration that the AV experienced. The aim was to capture harsh accelerations, braking, cornering and crash events. Hence a signal named *harsh accelerations* was derived from all the total accelerations >5 m/s^2 (values <5 m/s^2 are set $= 0$).

However, not all risks translate to high values of accelerations. For example, if an oncoming car whizzes past the AV on a narrow road, it does not reflect in the accelerations of the AV but is a risky scenario that needs to be captured. For this reason, we also calculated the Time To Collision (TTC) metric [26], [27] (equation (1)) and has been used to quantify risk exposure of AVs in different situations [28]. Put simply, it is the time before a collision will occur if the AV and the
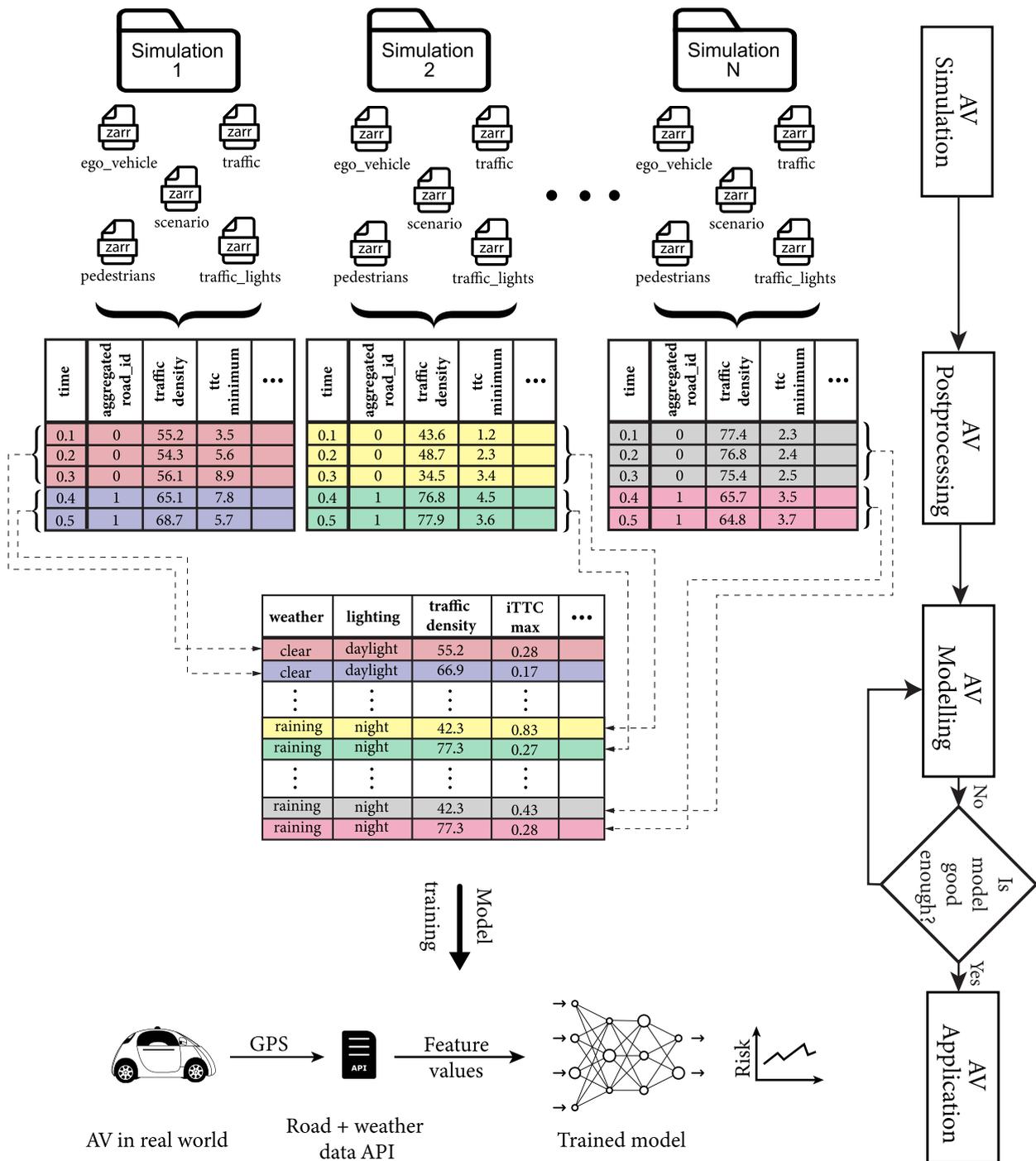
FIGURE 2. Data processing pipeline. The top row shows the 4 steps in the pipeline. (1) AV Simulation: The different ADSs are simulated in different environments and generate one folder per simulation. Each folder contains five '.zarr' files which contain raw data about the scenario (e.g., weather, lighting), ego_vehicle (e.g., speed, acceleration), all the pedestrians (e.g., position, bounding box), all the traffic vehicles (e.g., position, bounding box), and all the traffic_lights (e.g., traffic light state, position). (2) AV Post-processing: In this stage one DataFrame (saved as '.parquet') is created from the raw data (five '.zarr') files. The columns of the DataFrame consist of metrics such as *traffic density* and each row represents one time step. (3) AV Modelling: Since the goal of our modelling is to predict 'how risky a road section is for an AV to drive?', we need to aggregate the time-stamped rows from the '.parquet' file over an entire road section. The aggregation of each column is detailed in section II-C. Hence, each row in the aggregated DataFrame (saved as '.csv') represents a unique road section. The aggregated DataFrame is used for the train, validate, and test data spilt for the ML model. (4) Once we have the trained model, it is deployed in Humn.ai's cloud infrastructure, that connects with the AV on the road. It gathers the GPS location of the vehicle and queries several APIs to collect the values for the features (e.g., *traffic density*, *traffic average speed*) at the current GPS location and feeds it into the model. The model then makes a prediction about *how risky the current road section is*.

other vehicle continue moving at the same velocity.

$$\text{TTC} = \frac{\text{Relative distance}}{\text{Relative velocity}} \quad (1)$$

However, the TTC metric has a flaw, i.e. it does not work well when the vehicles are driving at the same velocity (relative velocity = 0). Hence, if a vehicle is tailgating the ego AV or if the AV was tailgating another vehicle (keeping a small relative distance) but following at the same speed, TTC would be infinite (suggesting zero risk). This problem is addressed by the Time Head Way (THW) metric [29], [30] given by equation (2)

$$\text{THW} = \frac{\text{Relative distance}}{\text{Velocity of ego vehicle}} \quad (2)$$

Since the inverse values of TTC and THW correlate with risk [31], a single metric that quantified the risk an AV is exposed to was calculated using the three metrics: harsh acceleration, inverse TTC (iTTC = 1/TTC), and inverse THW (iTHW = 1/THW).

A second iteration of data cleaning (details in Supplementary material Section IV) was performed after the post-processing, where:

1) Traffic density was capped to 250 vehicles/km.
2) All road sections with length < 50 m were discarded.
3) Number of lanes were capped at 4.

In summary, the output of the AV Post-processing stage is a time series dataset with metrics such as iTTC, iTHW, harsh acceleration, which quantify risk, as well as metrics such as *traffic density*, and *traffic average speed* which can be obtained via external APIs and define the environment the AV is travelling in Fig. 2.

C. AV MODELLING

In this stage the post-processed data was aggregated and classical machine learning models were trained. Since the aim is to to predict *how risky a **road section** is for an AV to drive?*, each data point in the dataset needs to correspond to a single road section. Hence, the time series data obtained from the AV Post-processing stage was aggregated over each road section (Fig. 2).

The metrics which remain constant through out a simulation (weather, lighting) do not need any aggregation. *Number of lanes* was aggregated as the closest integer to the (weighted by length) average number of lanes on the road section. The mean value was used as the aggregated metric for *road curvature, traffic density, traffic average speed*, and *walker density*. If any part of a road section was a junction, the *is junction* feature was aggregated to *True*.

The metrics that quantify the risk (iTTC, iTHW, harsh accelerations) were aggregated as the mean value over each road section. However, the road sections in our dataset had different lengths (>50 m) and a longer road section would have a larger probability of having a risky event. Hence, the mean values of iTTC, iTHW, and harsh acceleration were scaled by the length of the road section.

Since the total risk was a combination of the three risk metrics (iTTC, iTHW, and harsh acceleration), the final *risk factor* (label) was a summation of the three metrics. A simple summation sufficed since the three metrics had values in similar range (Table. 3).

TABLE 3. The three components of risk factor.

	Harsh accelerations	**iTTC**	**iTHW**
min	0.0000	0.0000	0.0000
max	0.2127	0.1165	0.4592
mean	0.0001	0.0035	0.0064
std	0.0033	0.0042	0.0116

Additionally, the feature *traffic flow* (equation 3) replaced the *traffic average speed* feature, since literature [32] suggests that *traffic flow* is highly correlated with accident rates equation (3).

$$\text{Traffic flow} = \text{Traffic density} \times \text{Traffic average speed} \quad (3)$$

The data was prepared for machine learning by (1) splitting the aggregated dataset into train, validation, and test datasets, (2) one-hot encoding the categorical features (weather, lighting), and (3) normalizing the continuous features (traffic density, traffic average speed, road curvature, walker density, no. of lanes).

TABLE 4. Number of data points for in the datasets.

ADS	**Train**	**Validation**	**Test**
OpenPilot	16383	4681	2340
Pylot	2643	755	377

Finally, for each ADS we had a train, validation and test dataset with the number of data points shown in Table 4. The training, structure and predictions of the models are discussed in the Results section, later in this paper. The AV application stage shown in Fig. 2 is not discussed in this paper, but involves the practical implementations (data engineering) to gather at from real world data via external APIs to generate an online risk estimate for the AV.

III. DATA ANALYSIS

In this section, the aggregated data which is the output of aggregation step in the AV Modelling stage is analysed and the distributions of environment parameters (Features in Table 5) and risk factor (Label in Table 5) for both ADSs are presented.

A. FEATURE AND LABEL DISTRIBUTIONS

The distributions of the environment parameters (features) and the risk factor (label) along with its components (*harsh accelerations*, iTTC, iTHW) for the two ADSs are provided in Fig. 3. The first two columns of Fig. 3 show the features and the third column shows the label.

1) FEATURE (ENVIRONMENT) DISTRIBUTIONS

Since the two ADSs were simulated in environments that were closest to the environments we expect them to be used
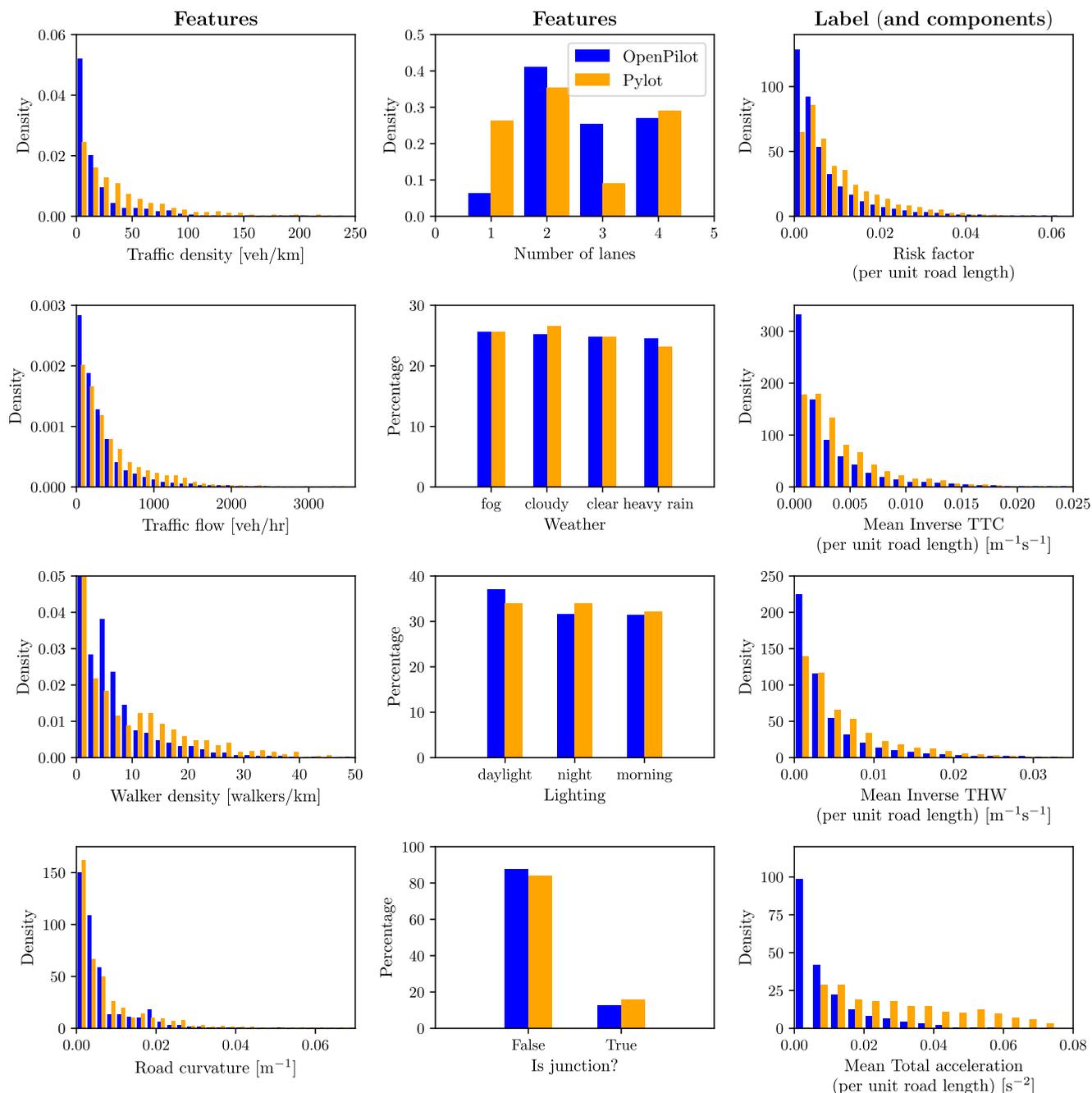
FIGURE 3. Data analysis summary: The distributions of the (environment) features (first two columns) and (risk factor) label (third column) are shown. The label (and its components) are represented in a *per unit road section length* format.

TABLE 5. Features and label for machine learning.

Features	*Continuous*	Traffic density, Traffic flow, Walker density, Road curvature, Number of lanes
	Categorical	Weather, Lighting, Is junction
Label		Risk Factor (= iTTC + iTHW + harsh acceleration)

in the real-world, differences can be observed in the distributions of the different features. OpenPilot being a Level 2/3 ADS was simulated mainly on the highways where the

no. of lanes > 1. Pylot, on the other hand, being a Level 4 ADS was simulated on both highways and urban settings. This difference can be clearly seen in the *no. of lanes* plot in Fig. 3 where the percentage of road sections with 1 lane for OpenPilot is low compared to that for Pylot. On the other hand, Pylot experienced fewer number of 3 lane highways compared to OpenPilot. This is also reflected in other plots (*traffic density*, *traffic flow*, *walker density*) in Fig. 3 where the density of higher values is more in Pylot compared to OpenPilot, indicating that Pylot was exposed to more dense scenarios in terms of the traffic.

The features that remain constant throughout a simulation (*weather* and *lighting*) were designed to be uniformly distributed (25% for each weather type and 33% for each lighting type). The minor deviations from the uniform distribution occur due to the simulations that were excluded in the data cleaning process. With regards to the *junctions*; as expected, the number of road sections without a junction (OpenPilot=87.6%, Pylot=84.0%) are far fewer compared to the ones with a junction (OpenPilot=12.4%, Pylot=15.9%).

2) LABEL (RISK FACTOR) DISTRIBUTIONS

The *risk factor* plot (Fig. 3 third column) shows that the risk for Pylot has higher density at higher values, as compared to OpenPilot. This is hypothesized to be also an affect of the differences in environment distributions. The components: iTTC and iTHW show similar trends in distribution as the *risk factor*. The *total acceleration* distributions show the most amount of difference between Pylot and OpenPilot. This is mainly because Pylot being simulated mainly in dense traffic scenarios (e.g., on urban roads) has more interaction with the traffic which results in higher number of (harsh) accelerations.

B. VALIDATING THE SIMULATION ENVIRONMENT

One of the essential components of the simulation was the traffic that interacted with the ADSs. Since, AVs will (at least initially) be sharing the roads with human drivers, it becomes vital that the simulated traffic exhibits patterns that are observed in human driving data (at the very least) on a macroscopic level. Figure 4 shows the macroscopic patterns generated by the traffic simulated in our scenarios.

The top row of Fig. 4 shows the scatter plots of *traffic average speed* VS *traffic density* for OpenPilot, Pylot, and from data measured on a highway from literature [33]. The measured data (human driving) shows that the *traffic average speed* decreases as the *traffic density* increases. This is because at low *traffic densities* vehicles can move at free flow speed, whereas at high *traffic densities* there is almost a 'grid lock' situation which results in low *traffic average speed*. The simulated traffic for OpenPilot shows a similar trend. However, in addition to the points that indicte the inverse relation between *traffic average speed* and *traffic density*, the plot is populated with points that are 'below the curve'. This is because the measured data from [33] is from highway sections, whereas the data from our simulations includes several factors such as traffic lights, junctions, etc. which lead to points that have a lower *traffic average speed*. For example, even when *traffic density* is zero, a red traffic light will result in zero *traffic average speed*. The simulated traffic for Pylot as shows the inverse relationship, however Pylot was simulated in more complex scenarios compared to OpenPilot, the higher values of *traffic average speed* are missing.

The bottom row of Fig. 4 shows the scatter plots of *traffic flow* VS *traffic density* for OpenPilot, Pylot, and from data measured on a highway from literature [33]. The measured data (human driving) shows that the *traffic flow* increases

TABLE 6. Parameters for XGBoost regressor model.

Parameter	XGBoost regressor	
	OpenPilot	**Pylot**
`n_estimators`	35	100
`objective`	`reg:squarederror`	`reg:squarederror`
`max_depth`	7	7
others	*default*	

linearly and then decreases as the *traffic density* increases. A similar trend is seen in the OpenPilot and Pylot plots. Essentially, this is due the fact that when *traffic density* is low, even if all the vehicles are going fast (high *traffic average speed*), the *traffic flow* is low. The *traffic density* reaches an optimum point where it doesn't hinder the *traffic average speed* and leads to the highest *traffic flow*. Beyond this optimum *traffic density*, it hinders the flow of traffic and eventually ends up in a 'grid-lock' at very high density.

Hence, from Fig. 4 it can be concluded that the simulated traffic behaved in a manner to represent human-driven vehicles interacting with the ADSs at the macroscopic level.

IV. RESULTS

In this section, we will discuss the classical machine learning models that were trained on the aggregated data (Section II-C). We have also implemented a deep learning model on the same dataset but is beyond the scope of this paper and will be discussed in a future publication.

The label (*risk factor*) that needed to be predicted by the model was a continuous signal, hence, regression models were explored. A grid search was performed to select the best model structure and the hyper parameters. The results led us to select a XGBoost Regressor model for both the ADSs, with the parameters as shown in Table 6.

The *MSE Loss* was used as a metric to compare the performance of the models with respect to a Baseline model. The Baseline model was the mean value of the *risk factor* (label) from the train dataset of the respective ADS. The predictions of the baseline models for the *risk factor* for OpenPilot and Pylot resulted in *MSE Loss* of 0.0002562 and 0.0000866, respectively on the test datasets. The XGBoost Regressor models yielded improvements of 25.0% and 52.4% with *MSE Loss* of 0.0001928 and 0.0000412 for OpenPilot and Pylot, respectively.

The ranking power of the models was evaluated using the lift curves. Figure 5 shows the lift curves for the two ADSs. It can be seen that the model for the Pylot is closer to the 'target lift' (calculated using the test data) as compared to the OpenPilot model. However, the OpenPilot model has a higher lift value.

A. RISK FACTOR VS FEATURES

The *MSE Loss* and *Lift curves* provide a comparative understanding of the model performance. However, to get a more complete picture of the model performance, the model's predictions were compared to the data by plotting the *risk*
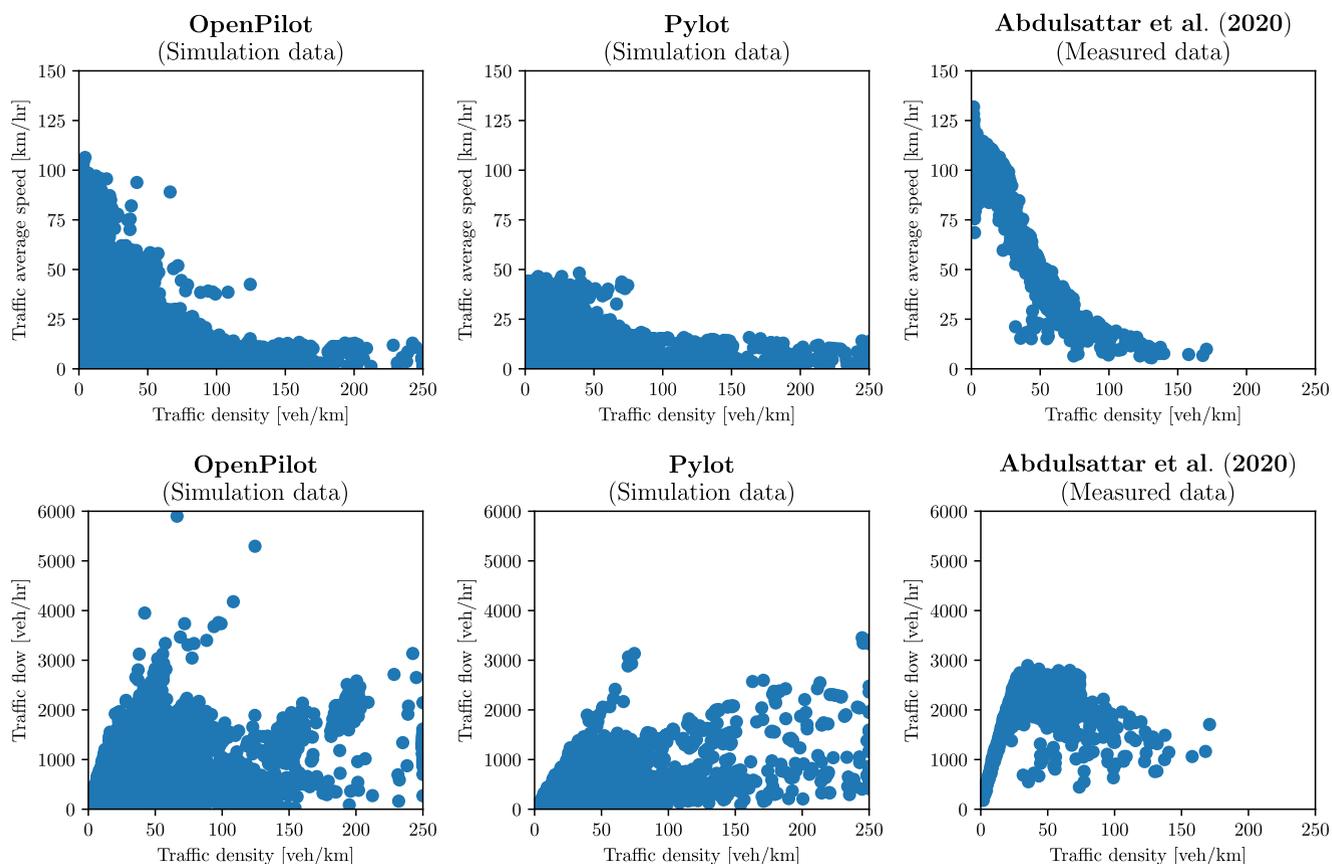
FIGURE 4. Simulator validation: The top row shows the relationship between *traffic average speed* and *traffic density* and the bottom row shows the relationship between *traffic flow* and *traffic density*. The first two columns have the data for the two ADSs from our simulations and are compared to the real-world measured data from literature [33] in the third column.
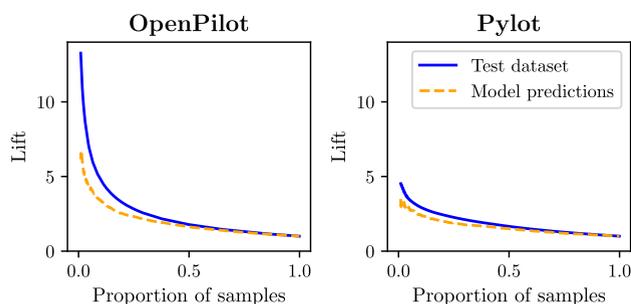
FIGURE 5. Lift curves: The dotted lines indicate the lift curve calculated by using the (XGBoost Regressor) model predictions on the test dataset. The solid line indicates the 'target' (ideal) lift curve, which is calculated using the test dataset.

factor vs the *features* in Fig. 7. This, we think, provides a more palatable way to understand the performance of the model. For example, from Fig. 7 it can be seen that the trends in the solid lines (green: train data, yellow: validation data, and red: test data) are captured well by the dotted lines of the XGBosst Regressor model predictions. Additionally, we also observe the (univariate) relationship between the different *features* and *risk factor*. For every plot the *Mean risk* is plotted on the y axis, which is the mean of the *risk factor* (label) for the interval in which the corresponding

feature (x axis) falls in. For continuous features `pandas.qcut` was used to discretize the features into equal sized buckets based on sample quantiles. For categorical features, the *Mean risk* corresponding to each feature value is plotted.

1) MEAN RISK VS TRAFFIC DENSITY

From Fig. 7 (a)-(b), it can be seen that the *mean risk* initially increases with *traffic density* and then plateaus, beyond 50 vehicles/km. This trend is captured well by the model predictions as well. This trend can be understood by the fact that there is an increased interaction of the AV with surrounding vehicles as the *traffic density* increases. Higher the interaction, higher the probability of a risky event occurring. But beyond a certain *traffic density*, this probability may not change significantly.

It is also interesting to note that the magnitudes of *mean risk* are similar for OpenPilot and Pylot. This maybe due to the fact that, although they are very different ADSs, they were simulated in their respective ODDs.

2) MEAN RISK VS TRAFFIC FLOW

From Fig. 7 (c)-(d), it can be seen that the *mean risk* increases with *traffic flow* for both the ADSs. This trend is similar to
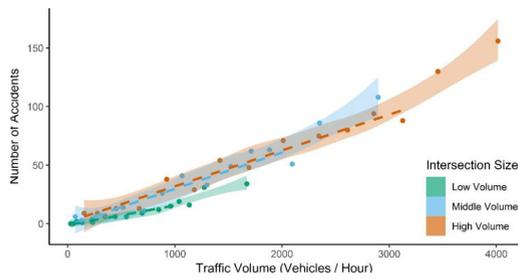
FIGURE 6. No. of accidents VS *traffic flow*: The data from literature [32] with human driven vehicles shows the positive correlation between number of accidents and *traffic flow*.

that found in literature [32] with human drivers, where the accidents increase as the *traffic flow* increases.

Traffic flow being the product of *traffic density* and *traffic speed*, combines the two 'opposing' features (high *traffic density* leads to low *traffic average speed*). This combined with the fact that it has a high correlation with the *mean risk* makes it a valuable feature and is also reflected in the top 4 of the feature importance plots (Fig. 7 (q)-(r)).

3) MEAN RISK VS WALKER DENSITY

From Fig. 7 (e)-(f), it can be seen that the *mean risk* initially decreases with *walker density* increase; reaches a minimum, and then increases, while plateauing at *walker densities*>15 pedestrians/km. We expected the *mean risk* to increase monotonously with *walker density*, however, the 'dip' seen in the data (and model predictions) is hypothesized to be due to the other confounding features. For example, low pedestrian densities are usually observed on highway sections where the crashes are usually at higher speeds leading to higher severity. Another point to be noted is that the values of *walker density* may seem low (e.g., 50 pedestrians/km), but pedestrians were usually concentrated at cross-walks and the rest of the road was empty. The pedestrians in the simulator were not allowed to *jaywalk*.

4) MEAN RISK VS ROAD CURVATURE

From Fig. 7 (g)-(h), it can be seen that the *mean risk* initially decreases with *road curvature* increase; reaches a minimum, and then increases, while plateauing at *road curvature*>0.015 m^{-1}. This is similar to the trend observed for *walker density*. Again, we expected the *mean risk* to increase monotonously with *road curvature*. The 'dip' is (again) hypothesized to be due to confounding feature. For example, straight roads (low *road curvature*) are generally highways which tend to have high severity crashes.

5) MEAN RISK VS NUMBER OF LANES

From Fig. 7 (j), it can be seen that the *mean risk* increases with *no. of lanes* for Pylot. However, for OpenPilot (Fig. 7 (i)) the *mean risk* 'dips' at number of lanes = 3 (a highway). The feature importance plot for OpenPilot (Fig. 7 (q)) also shows that *no. of lanes* feature is very important. In contrast,

for Pylot, the *no. of lanes* feature is not very important. This, we think, is due to the fact that OpenPilot (Level 2/3 system) which is mainly driven on highways and Pylot (Level 4 system) is mainly driven in urban areas.

6) MEAN RISK VS WEATHER

From Fig. 7 (k)-(l), it can be seen that the *mean risk* was not affected by *weather*. This, we think, is a limitation of our simulation data, since it is very evident from literature [34], [35] that weather conditions are a significant factor in the functioning of an AV. We hypothesize that our simulations cold not capture the effect of weather due to limitations in simulating the sensors (lidars, radars, etc.) and low noise in the sensors (as compared to the ones used on real-world AVs). Another possibility is that the affects of weather were not strong enough. This will be investigated in a future paper.

7) MEAN RISK VS LIGHTING

From Fig. 7 (m)-(n), it can be seen that the *mean risk* was not affected by *lighting*. Similar to *weather*, we think, is a limitation of our simulation data, since it is very evident from literature [34], [36] that lighting conditions are a significant factor in the safe operations of an AV. This (similar to weather) is attributed to the limitations of our sensors which may be providing the perception module 'better' data that what be possible in the real world.

8) MEAN RISK VS IS JUNCTION

From Fig. 7 (o)-(p), it can be seen that the *mean risk* was high in a junction compared to when the road section did not include a junction. This is logical since the number of interactions the AV has with traffic at a junction are considerably higher than while driving on a straight road. The high importance of the *is junction* feature is also reflected in the feature importance plots for both, OpenPilot and Pylot.

9) FEATURE IMPORTANCE

Figures 7 (q)-(r) show the feature importance for OpenPilot and Pylot, respectively. *Is junction* is the most important feature for both ADSs and is logical since, larger the number interactions, higher the risk for an AV. The second most important feature for OpenPilot is the *no. of lanes* and is probably because OpenPilot is mainly meant to drive on highways. Pylot on the other hand has *traffic flow* as its second most important feature. These features are followed by the remaining continuous features of *traffic density*, *road curvature*, and *walker density*. The categorical features of *weather* and *lighting* do not seem to have much affect on the predictions of the model.

In summary, it can be said that the trends shown by the data are captured well by the XGBosst Regressor models, for both OpenPilot and Pylot. The trends in *mean risk* vs *features* agree with traditional knowledge for human driven vehicles. However, *weather* and *lighting* parameters do not seem to
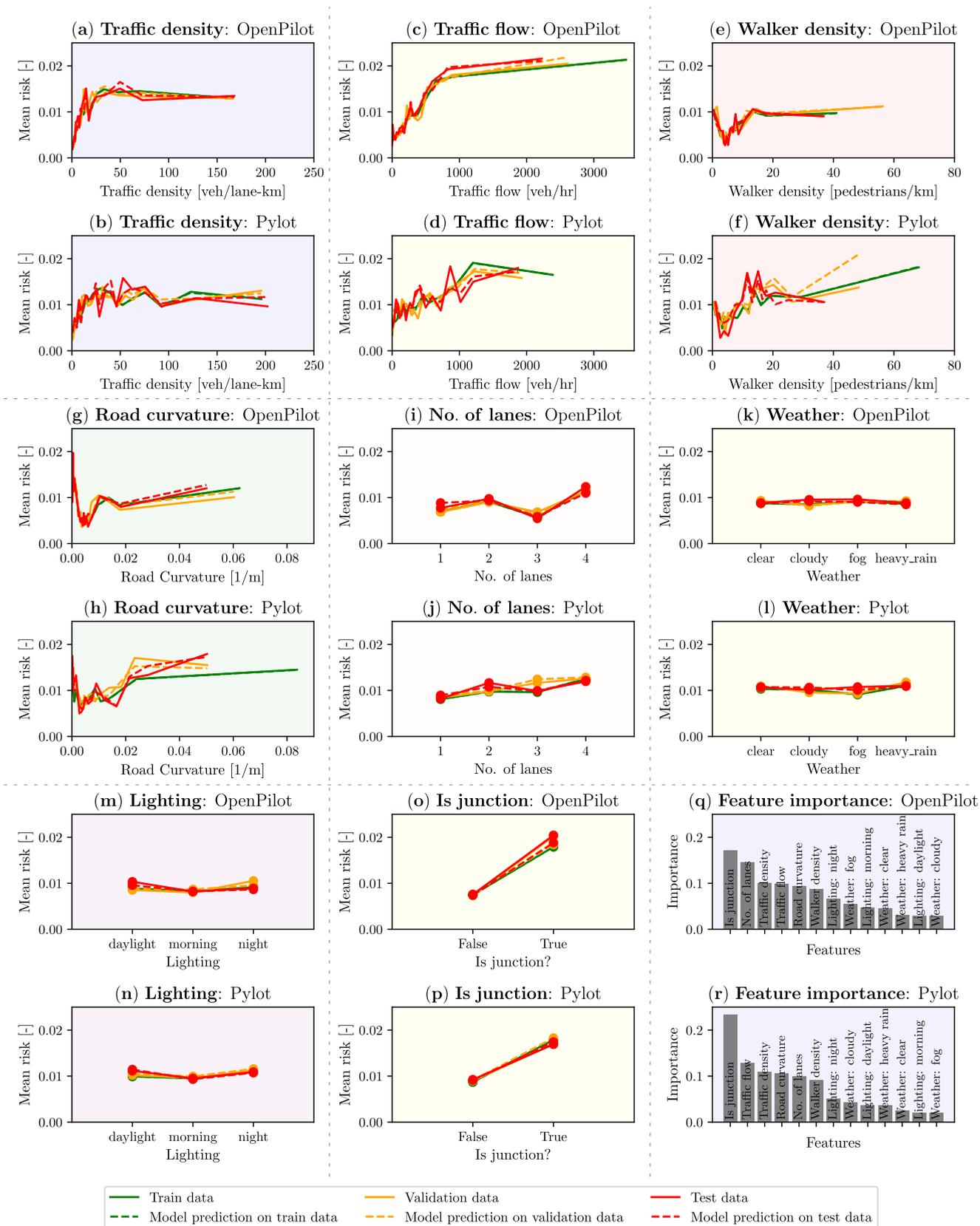
FIGURE 7. Mean risk VS features: The figure contains 9 groups of 2 subplots each. The 9 groups are from the 8 features and the 1 feature importance plot. The *Mean risk* on the y axis is the mean of the *risk factor* (label) for the corresponding interval in which the feature (x axis) falls in. For continuous features `pandas.qcut` was used to discretize the features into equal sized buckets based on sample quantiles. For categorical features, the *Mean risk* corresponding to each feature value is plotted.

affect the behaviour of the AV, which is due to the limitations of our simulation.

V. DISCUSSION

The aim of this paper was to answer two questions:

- How risky a road section is for an AV to drive?
- How does the risk profile vary with different (SAE levels) of ADSs?

To answer these questions, we developed a simulation pipeline as discussed in Section II-A and simulated 2 ADSs, namely: OpenPilot (a Level 2/3 system) and Pylot (a Level 4 system). Each simulation was 150 seconds long with different (CARLA) Towns and starting points defining the different scenes. Relevant signals and metrics were derived from the recorded raw data (position, speed, acceleration). Signals such as *traffic density*, *traffic flow*, *road curvature*, *weather* etc. were deived and used as features due to their relevance in affecting human driven vehicles and availability via traffic and weather APIs (in terms of practical application).

A *risk factor* metric was formulated from *inverse Time to Collision* (iTTC), *inverse Time Headway* (iTHW), and the *harsh accelerations* from the AV. This metric would not only allow us to capture crashes and high accelerations events, but also time critical events that do not reflect as *harsh accelerations*.

The data was then cleaned and prepared for training separate models for the two ADSs. Since the label (*risk factor*) is a continuous variable, a Regressor (XGBoost) was chosen for the model structure. The results (Fig. 7) show that the models were able to capture the trends in relation between the *mean risk* and the different *features*. Additionally, the trends could be explained using the data observed in human driven vehicles. These model predictions provided us with the answer to the first question i.e., *How risky a road section is for an AV to drive?*

The second question of *How does the risk profile vary with different (SAE levels) of ADSs?* is most suitably answered by the feature importance plots (Fig. 7 (q)-(r)). Junctions being an 'interaction' hot-spot have the highest importance for both the ADSs. However, after this the differences can be seen. OpenPilot being a Level 2/3 system, which is mainly intended to be driven on highways, has *no. of lanes* as the second most important feature. Whereas, for Pylot being a Level 4 system, which can be driven on highways and urban areas, has *traffic flow* as the second most important feature. It is also interesting to see that the importance for *Is junction* and *no. of lanes* are closer as compared to the importance for *Is junction* and *traffic flow* for Pylot. Pylot has a higher sensitivity to junctions compared to OpenPilot. Such differences are valuable for understanding 'risk potential' of different road features depending on the type of the ADS.

As mentioned earlier, one of the limitations of the simulation data was that it was unable to capture the effect of *weather* and *lighting*. This is hypothesized to be either due to the weather and lighting not being simulated effectively or the sensors being 'unrealistically' perfect. This raises the need for methods and metrics by which the sensors in the simulation can be calibrated to match the sensors used on the real vehicle. Metrics that can assess the performance and quality of sensors can be a valuable tool for such calibration [37]. This will be investigated in an upcoming paper. Another possible reason for the insensitivity of the *risk factor* to *weather* and *lighting* could be the non-dependence of traffic agents' behaviour on these factors. For example, the probability of a tail-gator rear ending an AV is higher on a wet road compared to a dry road. In future iterations, these effects will be enabled and tested for.

Despite these limitations, the simulation data and models provide valuable insights for a macroscopic model of risk for AVs from an insurers perspective. With more digitization and the concept of *digital clones* picking up momentum, more relevant features will be available in the future, that could help improve the predictive qualities of risk models.

VI. CONCLUSION

The work in this paper assesses the affect of the environment on an AV in a simulated environment. The results, with their coherence with trends found in literature lead us to conclude that parameterizing the environment using features (*traffic density*, *traffic flow*, *weather*, etc.) provides a computationally tractable way for simulating Autonomous Vehicles. In terms of the first question: *How risky a road section is for an AV to drive?* we can conclude that a road section with a *junction* and high *traffic flow*, high *traffic* and *walker* density are risky for an AV. These results are similar to the results found in literature for human drivers. When answering the second question: *How does the risk profile vary with different (SAE levels) of ADSs?*, we conclude that level 2/3 and level 4 ADSs show similarities in terms of the result that for both ADSs *junctions* are a risk hot-spot. However, the differences appear when we look at the feature importance, where a level 2/3 ADS is more sensitive to *no. of lanes* in comparison to a level 4 ADS. These results and conclusions provide a strong base to develop more complex models in the future that include other factors such as takeover requests, cyber security, and connected car technology.

CONFLICTS OF INTEREST

There are no conflicts to declare.

ACKNOWLEDGMENT

REFERENCES

[1] P. A. Hancock, T. Kajaks, J. K. Caird, M. H. Chignell, S. Mizobuchi, P. C. Burns, J. Feng, G. R. Fernie, M. Lavallière, I. Y. Noy, D. A. Redelmeier, and B. H. Vrkljan, "Challenges to human drivers in increasingly automated vehicles," *Hum. Factors*, vol. 62, no. 2, pp. 310–328, Mar. 2020.

[2] D. Bissell, T. Birtchnell, A. Elliott, and E. L. Hsu, "Autonomous automobilities: The social impacts of driverless vehicles," *Current Sociol.*, vol. 68, no. 1, pp. 116–134, Jan. 2020.

[3] M. O'Kelly, A. Sinha, H. Namkoong, J. Duchi, and R. Tedrake, "Scalable end-to-end autonomous vehicle testing via rare-event simulation," in *Proc. 32nd Int. Conf. Neural Inf. Process. Syst.* Red Hook, NY, USA: Curran Associates, 2018, pp. 9849–9860.

[4] J. M. Anderson, K. Nidhi, K. D. Stanley, P. Sorensen, C. Samaras, and O. A. Oluwatola, *Autonomous Vehicle Technology: A Guide for Policymakers*. Santa Monica, CA, USA: Rand Corporation, 2014.

[5] G. Yeomans, "Autonomous vehicles: Handing over control-opportunities and risks for insurance," *Lloyd's*, vol. 18, Mar. 2014.

[6] S. Abdelhadi, K. Elbahnasy, and M. Abdelsalam, "A proposed model to predict auto insurance claims using machine learning techniques," *J. Theor. Appl. Inf. Technol.*, vol. 98, no. 22, pp. 1–10, 2020.

[7] N. Gatzert and K. Osterrieder, "The future of mobility and its impact on the automobile insurance industry," *Risk Manage. Insurance Rev.*, vol. 23, no. 1, pp. 31–51, Mar. 2020.

[8] H. Summala and T. Mikkola, "Fatal accidents among car and truck drivers: Effects of fatigue, age, and alcohol consumption," *Hum. Factors*, vol. 36, no. 2, pp. 315–326, Jun. 1994.

[9] A. P. Smith, "A U.K. Survey of driving behaviour, fatigue, risk taking and road traffic accidents," *BMJ Open*, vol. 6, no. 8, Aug. 2016, Art. no. e011461.

[10] P. Koopman and F. Fratrik, "How many operational design domains, objects, and events?" in *Proc. Safeai@ AAAI*, 2019, pp. 1–4.

[11] *SAE Levels of Driving Automation Refined for Clarity and International Audience*. Accessed: Oct. 26, 2022. [Online]. Available: https://www.sae.org/blog/sae-j3016-update

[12] P. Bhavsar, P. Das, M. Paugh, K. Dey, and M. Chowdhury, "Risk analysis of autonomous vehicles in mixed traffic streams," *Transp. Res. Rec.*, vol. 2625, no. 1, pp. 51–61, 2017.

[13] S. Shah, D. Dey, C. Lovett, and A. Kapoor, "AirSim: High-fidelity visual and physical simulation for autonomous vehicles," in *Proc. 11th Int. Conf. Field Service Robot.*, Zurich, Switzerland. Springer, 2018, pp. 621–635. [Online]. Available: https://dblp.org/db/conf/fsr/index.html

[14] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proc. 1st Annu. Conf. Robot Learn.*, 2017, pp. 1–16.

[15] G. Chance, A. Ghobrial, K. McAreavey, S. Lemaignan, T. Pipe, and K. Eder, "On determinism of game engines used for simulation-based autonomous vehicle verification," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 11, pp. 20538–20552, Nov. 2022.

[16] J. M. Scanlon, K. D. Kusano, T. Daniel, C. Alderson, A. Ogle, and T. Victor, "Waymo simulated driving behavior in reconstructed fatal crashes within an autonomous vehicle operating domain," *Accident Anal. Prevention*, vol. 163, Dec. 2021, Art. no. 106454.

[17] M. Schwall, T. Daniel, T. Victor, F. Favaro, and H. Hohnhold, "Waymo public road safety performance data," 2020, *arXiv:2011.00038*.

[18] M. Hoppe, J. C. Kirchhof, E. Kusmenko, C. Y. Lee, and B. Rumpe, "Agent-based autonomous vehicle simulation with hardware emulation in the loop," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2022, pp. 16–21.

[19] G. Li, Y. Yang, T. Zhang, X. Qu, D. Cao, B. Cheng, and K. Li, "Risk assessment based collision avoidance decision-making for autonomous vehicles in multi-scenarios," *Transp. Res. C, Emerg. Technol.*, vol. 122, Jan. 2021, Art. no. 102820.

[20] C. Sun, Z. Deng, W. Chu, S. Li, and D. Cao, "Acclimatizing the operational design domain for autonomous driving systems," *IEEE Intell. Transp. Syst. Mag.*, vol. 14, no. 2, pp. 10–24, May 2021.

[21] E. de Gelder, A. K. Saberi, and H. Elrofai, "A method for scenario risk quantification for automated driving systems," in *Proc. 26th Int. Tech. Conf. Enhanced Saf. Vehicles (ESV)*, 2019, pp. 1–11.

[22] J. Norden, M. O'Kelly, and A. Sinha, "Efficient black-box assessment of autonomous vehicle safety," 2019, *arXiv:1912.03618*.

[23] *OpenPilot Integrated With CARLA*. Accessed: Oct. 15, 2022. [Online]. Available: https://github.com/commaai/openpilot/wiki/CARLA

[24] I. Gog, S. Kalra, P. Schafhalter, M. A. Wright, J. E. Gonzalez, and I. Stoica, "Pylot: A modular platform for exploring latency-accuracy tradeoffs in autonomous vehicles," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2021, pp. 8806–8813.

[25] L. Chen, T. Tang, Z. Cai, Y. Li, P. Wu, H. Li, J. Shi, J. Yan, and Y. Qiao, "Level 2 autonomous driving on a single device: Diving into the devils of openpilot," 2022, *arXiv:2206.08176*.

[26] E. R. Hoffmann and R. G. Mortimer, "Drivers' estimates of time to collision," *Accident Anal. Prevention*, vol. 26, no. 4, pp. 511–520, 1994.

[27] R. Van Der Horst and J. Hogema, "Time-to-collision and collision avoidance systems," Verkeerskundig Studiecentrum, Rijksuniversiteit Groningen, Tech. Rep., 1994.

[28] S. Lefèvre, D. Vasquez, and C. Laugier, "A survey on motion prediction and risk assessment for intelligent vehicles," *ROBOMECH J.*, vol. 1, no. 1, pp. 1–14, 2014.

[29] W. V. Winsum and A. Heino, "Choice of time-headway in car-following and the role of time-to-collision information in braking," *Ergonomics*, vol. 39, no. 4, pp. 579–592, Mar. 1996.

[30] D. Swaroop and K. R. Rajagopal, "A review of constant time headway policy for automatic vehicle following," in *Proc. IEEE Intell. Transp. Syst. (ITSC)*, Aug. 2001, pp. 65–69.

[31] V. E. Balas and M. M. Balas, "Driver assisting by inverse time to collision," in *Proc. World Autom. Congr.*, Jul. 2006, pp. 1–6.

[32] A. E. Retallack and B. Ostendorf, "Relationship between traffic volume and accident frequency at intersections," *Int. J. Environ. Res. Public Health*, vol. 17, no. 4, p. 1393, Feb. 2020.

[33] H. Abdulsattar, A. Mostafizi, M. R. K. Siam, and H. Wang, "Measuring the impacts of connected vehicles on travel time reliability in a work zone environment: An agent-based approach," *J. Intell. Transp. Syst.*, vol. 24, no. 5, pp. 421–436, Sep. 2020.

[34] J. Vargas, S. Alsweiss, O. Toker, R. Razdan, and J. Santos, "An overview of autonomous vehicles sensors and their vulnerability to weather conditions," *Sensors*, vol. 21, no. 16, p. 5397, Aug. 2021.

[35] S. Zang, M. Ding, D. Smith, P. Tyler, T. Rakotoarivelo, and M. A. Kaafar, "The impact of adverse weather conditions on autonomous vehicles: How rain, snow, fog, and hail affect the performance of a self-driving car," *IEEE Veh. Technol. Mag.*, vol. 14, no. 2, pp. 103–111, Jun. 2019.

[36] X. Zhang, B. Story, and D. Rajan, "Night time vehicle detection and tracking by fusing sensor cues from autonomous vehicles," in *Proc. IEEE 91st Veh. Technol. Conf. (VTC-Spring)*, May 2020, pp. 1–7.

[37] A. Warchoł, "The concept of LiDAR data quality assessment in the context of BIM modeling," *Int. Arch. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. 42, pp. 61–66, Sep. 2019.

JOHN ANIH received the M.E. degree in automation and information technology (IT) from the Cologne University of Applied Sciences (TH Koln), Germany. He is currently a Data Scientist and a Software Developer with Humn.ai Ltd., in partnership with the University of Salford through the U.K. Research and Innovation (UKRI). He has over four years of professional experience designing software applications for AI and data science.

SARVESH KOLEKAR received the B.Tech. degree in mechanical engineering from NIT Surathkal, India, in 2012, and the M.Sc. and Ph.D. degrees in mechanical engineering from the Delft University of Technology, The Netherlands, in 2016 and 2021, respectively. He was a Visiting Researcher with Nissan, Atsugi, Japan, in 2019, and the CBL Laboratory, University of Cambridge, in 2014. He is currently a part of the Data Science Team, Humn.ai. His research interests include human driver models, risk models, sensorimotor control, and vehicle dynamics.

TOOSKA DARGAHI received the Ph.D. degree in computer engineering (network security) from the Tehran Science and Research Branch, Azad University, Iran, in 2014. She was a Visiting Researcher with the University of Padua, Italy, in 2014, and a Research Fellow with the University of Rome Tor Vergata, Italy, from 2015 and 2017. She is currently a Senior Lecturer in cyber security with Manchester Metropolitan University (Manchester Met), U.K. Before joining Manchester Met, she was a Lecturer in cyber security, and the Programme Leader for M.Sc. in cyber security with the University of Salford. She has an established track record in the field of security and privacy in distributed systems, including the Internet of Things (IoT) and smart cities. Her recent work includes research on the privacy challenges of IoT devices and intelligent transportation systems. She has served as an organizing committee member for several IEEE/ACM conferences and workshops, a reviewer for several high-ranked journals, a guest editor for several special issues in reputable journals, and an Associate Editor of the book titled *Cyber Threat Intelligence* (Springer).

MEISAM BABAIE received the Ph.D. degree in mechanical engineering from the Queensland University of Technology (QUT), Australia, in 2014. He is an experienced mechanical engineer and scientist, with an employment history in both industry and academia. He worked in different industries such as automotive manufacturing, disposable products manufacturing, and cement industry and international academic institutions and laboratories such as the Biofuel Engine Laboratory Facility, Australia; Gunma University, Japan; and the Automotive and Autonomous Vehicle Technology Laboratory, U.K. He is currently a Lecturer with the School of Mechanical Engineering, University of Leeds, U.K. His research interests encompass different areas of modern automotive engineering including connected and autonomous vehicles. He has an established track record in the field, and his research outcome has been presented at reputed conferences, industrial reports, invited talks, and prestigious journals of the field.

MOHAMAD (MO) SARAEE received the Ph.D. degree in computer science from the University of Manchester. He holds the Chair in data science, and he is the Programme Leader of the M.Sc. in data science and the M.Sc. in IoT with data science courses with the University of Salford-Manchester. He leads a Data Science Research Group that focuses on machine and deep learning, data and text mining, NLP, big data, and medical informatics. His research addressed multidisciplinary, cross-school topics with transformative impact, benefiting local communities, and dedication to action through real-world application of research including developing and integrating innovative data mining approaches to improve human health, in collaboration with both Salford City Council and the NHS. He has secured and led funded projects with an income total of over $1.7 m. He is the Editor-in-Chief of *International Journal of Web Research* and a reviewer for several high-tier international journals.

JACK WETHERELL received the Ph.D. degree in theoretical physics from the University of York and the York Center, Quantum Technologies. He is currently a Machine Learning Data Scientist with Humn.ai. His role is to develop machine-learning models to quantify and predict the environmental and behavioral risks generated by fleets of autonomous vehicles. He deploys many types of real-world automated driving systems in state-of-the-art simulators in a range of diverse scenarios, to generate risk profiles of the vehicles and learn how these couple to the environmental effects. This allows him to train machine-learning models that can couple the environmental features of a vehicle to its driving events that produce risk and are unique to self-driving systems.

· · ·