

An Automated Machine Learning (AutoML) Method of Risk Prediction for Decision-Making of Autonomous Vehicles

Xiupeng Shi[✉], Yiik Diew Wong, Chen Chai, *Member, IEEE*, and Michael Zhi-Feng Li

Abstract—This study presents a domain-specific automated machine learning (AutoML) for risk prediction and behaviour assessment, which can be used in the behavioural decision-making and motion trajectory planning of autonomous vehicles (AVs). The AutoML enables end-to-end machine learning from vehicle movement and sensing data to detailed risk levels and corresponding behaviour characteristics, which integrates three main components of: unsupervised risk identification by surrogate risk indicators and big data clustering, feature learning based on XGBoost, and model auto-tuning by Bayesian optimisation. Then, the functions and performance of AutoML are evaluated based on NGSIM data, with assumptions of various sensing configurations or data acquisition conditions. AutoML achieves satisfactory results of behaviour-based risk prediction, which has a predictive power of 91.7% overall accuracy for four risk levels, and about 95% accuracy for safe-risk distinction. Bayesian optimisation guides the self-learning of AutoML to get the optimised feature subsets and hyperparameter values. The identification of key features not only produces better performance with fewer computation costs, but also provides data-driven insights about AV design, such as sensor configurations and sensor data mining, from risk decision-making perspectives. The application potentials of AutoML in AVs are discussed.

Index Terms—Automated machine learning, risk prediction, autonomous vehicles, sensor configurations.

I. INTRODUCTION

AUTONOMOUS vehicles (AVs) have the potentials to dramatically reduce vehicle collisions associated with driver errors and negligence. However, driving scenarios and risk conditions are hugely complex, and the biggest challenge facing AVs is how to operationalise the required high level of safety and reliability in driving [1]. With maturity in sensor perception and vehicle control, safe decision-making and risk assessment mechanisms are becoming increasingly important towards promoting AV development.

Manuscript received December 5, 2019; revised March 28, 2020; accepted June 3, 2020. This work was supported in part by the National Natural Science Foundation of China under Grant 61803283 and in part by the National Key Research and Development Program of China under Grant 2018YFB1600500. The Associate Editor for this article was C. Guo. (*Corresponding author: Chen Chai*)

Xiupeng Shi and Yiik Diew Wong are with the School of Civil and Environmental Engineering, Nanyang Technological University, Singapore 639798 (e-mail: xpshi@ntu.edu.sg; cydwong@ntu.edu.sg).

Chen Chai is with the College of Transportation Engineering, Tongji University, Shanghai 201804, China (e-mail: chaichen@tongji.edu.cn).

Michael Zhi-Feng Li is with the Nanyang Business School, Nanyang Technological University, Singapore 639798 (e-mail: zfli@ntu.edu.sg).

Digital Object Identifier 10.1109/TITS.2020.3002419

The decision-making mechanisms and technical solutions of self-driving are complex. Briefly, typical models include rule-based systems, end-to-end deep learning, and their hybrids. Rule-based systems (e.g., finite state machine) pre-define thousands of behaviour rules based on expert knowledge, which have certain interpretability and transparency on behaviour-risk reasoning [2]. To consider all possible driving scenes, the rules are incredibly complex, which essentially limits the prospects of further improvement. End-to-end deep learning is data-driven and scalable, in which the actions (e.g., steering, speed) are directly learned from sensed scenes (e.g., lidar point cloud, camera video) [3]. Based on well-tuned models and high-quality big data, deep learning is expected to achieve high performance and be adopted widely, such as LSTM (long short-term memory) [4]. But such scene-to-action learning is more like the intuitions of human drivers, and is less straightforward to inspect and control quantitatively (e.g., when assessing risk levels). Besides, motion planning is also related to the future behaviour intentions and trajectories of surrounding vehicles (e.g., within next 8s in Apollo), which encounters high behaviour uncertainty [5], [6]. Herein, interpretable learning from behaviours to risk levels can benefit existing systems, which adds decisive advantages, such as causal reasoning of behaviour-to-risk, modular division of sensing and decision-making, etc. It is important that behaviour-risk decision-making can be inspected, such as the responsibilities and faults-checking of individual components [7]. Reliable risk prediction can enhance the confidence level of decision-making in uncertainty, and early identification of potential risk conditions can empower AVs with advanced intelligence.

To build an interpretable and successful machine learning system for risk prediction is inherently challenging. Achieving a state-of-the-art performance depends not only on the fundamental power of the core algorithms, but also on careful data processing, useful feature engineering, optimised hyperparameter tuning, and well-configured end-to-end pipeline [8], [9]. Decision-tree based ensemble learning algorithms have demonstrated the advantages in interpretability and performance, such as XGBoost [10]. Moreover, there are greater technical and practical challenges to build machine learning used for risk decision-making in self-driving. Firstly, it is not straightforward to obtain the ground-truth data labels about risk levels, which is the basic to guide behaviour-risk learning. Feature engineering is an important aspect to

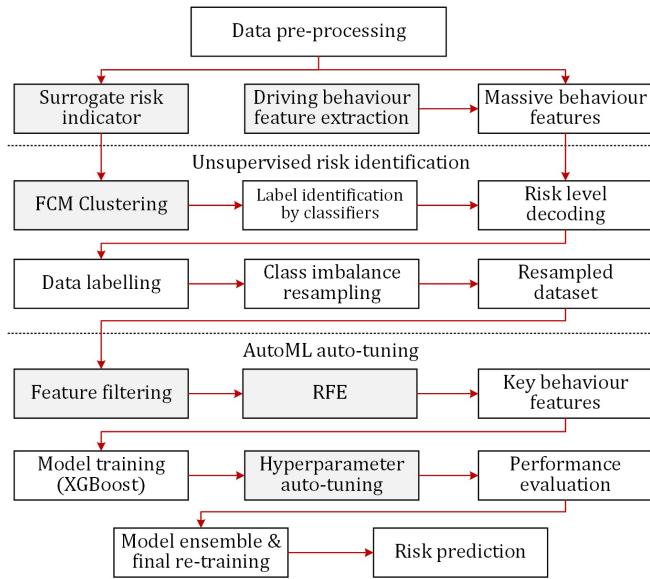


Fig. 1. Risk prediction AutoML framework.

improve interpretability and predictive ability, which is the process of using domain knowledge to extract and select implicit information from sensing data, as the input of learning algorithms [11], [12]. The performance of a given model also leverages on hyper-parameter tuning, which usually requires manually trying out the best one from all possible settings and values [8]. Risk decision-making of AVs faces complex scenes and changeable factors, which vary in time and space, thereby efficient machine learning pipelines must incorporate auto-tuning and self-learning procedures [13]. Besides, considering the time and computation constraints, it is more practical to have a lean and fast-response machine learning system that can achieve high performance using small sample data [14]. Certainly, such domain-specific automated machine learning (AutoML) for driving risk prediction enables a lot of benefits but relevant research is still much lacking.

The focus of this study lies on the design of a domain-specific AutoML for risk prediction and driving data mining, with the aim of application in behavioural decision-making and motion trajectory planning of AVs. The methodology is introduced in Section 2. Section 3 elaborates the main steps and performance of AutoML, using vehicle trajectory data as a test scenario. The final two sections cover the discussion and conclusions.

II. METHODOLOGY

A. Framework of AutoML for Risk Decision-Making

An AutoML framework is designed to achieve self-optimised modelling of driving risk prediction and behaviour assessment. The AutoML assembles necessary modelling steps as an end-to-end machine learning pipeline and automates the pipeline to get the features, algorithms, and hyperparameters that return the best performance as measured on validation datasets, which is a process of learning to learn.

The AutoML framework is depicted in Fig. 1. Main modules of the AutoML pipeline include massive feature extraction

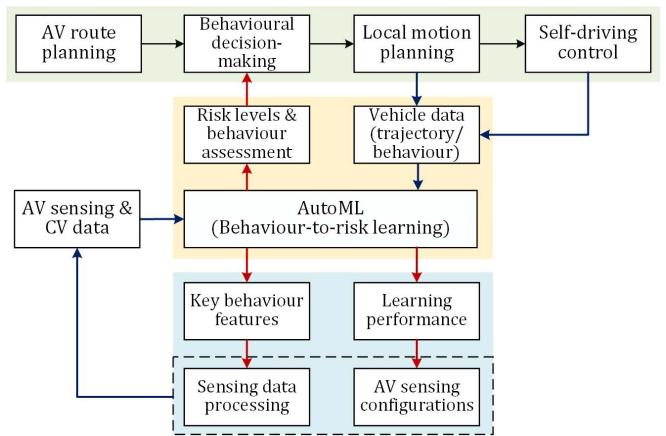


Fig. 2. AutoML for AV decision-making.

and learning-based selection, unsupervised risk identification, risk prediction based on key behaviour features, and model auto-tuning by Bayesian optimisation. XGBoost is used as the key classifier throughout the pipeline, and the clustering algorithm is Fuzzy C-means (FCM) [15]. Bayesian optimisation guides the self-learning process to find the best-suited algorithms and hyperparameter values within specific computation capacity and time constraints. Besides, the pipeline also incorporates pre-processing steps such as imbalanced data resampling, noise filtering, among others. The framework enables an end-to-end AutoML that maps vehicle driving data to risk levels.

Safety is one of the core objectives of AV decision-making, and risk prediction helps to determine an optimal path or trajectory to be executed in local motion planning [16]. Fig. 2 demonstrates the roles of AutoML in the AV decision-making hierarchy. Given a sequence of route segments planned, the behavioural decision-making layer is responsible for generating dynamically feasible sets of movement based on perceived driving scenes (e.g., relative position, assessment of nearby vehicles). Due to uncertainty, making an optimal choice of next-step movements is not easy, which needs to satisfy all the constraints and criteria, such as being collision-free, efficient and comfortable. Herein, the domain-specific AutoML can offer behaviour assessment and predict the risk levels of instantaneous next-step movements, which helps to decide a better local motion to be executed.

B. Risk Identification by Clustering and Surrogate Indicators

The risk identification and data labelling method in the AutoML is the unsupervised risk grading by big data clustering and surrogate risk indicators. This method estimates the comparative risk levels of vehicles in driving from a large scope of vehicle stream. Surrogate risk indicators are expected to punctually distinguish between risk and safety, as well as estimate risk levels [17]. The clustering groups vehicles with similar risk patterns (measured by surrogate indicators) into the same class, and captures the position (i.e., the risk level) of each instance (i.e., each vehicle) in the entire dataset (i.e., vehicle stream). Then a label about the belonging position can be decoded and assigned to the instance [15].

Algorithm 1 Risk Clustering

1. Extraction of risk indicator features:
 - a. Data clean and pre-processing;
 - b. Surrogate measures of vehicle conflicts;
 - c. Risk indicator features of instance i , $x_i^{(r)}$.
2. Cluster $X^r = \{x_i^{(r)}\}$ into k groups ($k \in [3, N]$):
 - a. Select models (FCM), configure hyperparameters;
 - b. Produce partition matrix, $i \in G_p$, $p \in [1, k]$;
 - c. Decode risk pattern, assign risk levels $Y_k = \{y_i\}$.
3. Evaluation by classifiers:
 - a. Model training with (X^r, Y_k) by XGBoost;
 - b. Classification evaluation $E(Y_k)$;
 - c. Select $Y^* = \operatorname{argmax}_k E(Y_k)$.

The methods of risk identification and data labelling module are laid out in Algorithm 1, including the steps of: extraction of risk indicator features, clustering model selection and hyper-parameter configurations, clustering results evaluation and risk pattern decoding. For the clustering results evaluation, given that no ground-truth labels about risk levels are predefined, the method of label identification by classifiers is used to evaluate the clustering performance [15].

C. Risk Prediction Based on Selected Behaviour Features

The AutoML relies on two key aspects to improve the risk prediction and behaviour assessment from individual vehicle viewpoint, namely, in-depth behaviour feature extraction, and reliable feature selection. Massive behaviour features are extracted to provide the fundamental of predictive ability, and a learning-based feature selection method is developed to identify the most important features used for prediction.

The feature selection procedure firstly ranks and filters a set of relatively important features based on XGBoost, and then permutes to find an optimal feature combination using recursive feature elimination (RFE) [18], as laid out in Algorithm 2.

The feature relative importance is measured by the split weight and average gain of each feature, which are generated in the process of XGBoost fitting. The feature filtering by importance ranking efficiently reduces the feature space, and RFE further considers the interactions among features, and identifies the best combination. Based on the selected key features, the risk levels are predicted using XGBoost.

D. Hyperparameter Auto-Tuning

Machine learning is optimised by finding the best algorithm settings and hyperparameter values that maximise the model performance on validation sets. This process usually requires trying out all promising settings and values, which entails huge combinations. Besides, the mapping from the hyperparameters to the performance is known as a black box function, which is tedious and expensive to optimise.

Bayesian optimisation is an efficient solution in such context, which constructs a probabilistic (surrogate) model of the objective function that maps input values to a probability of a

Algorithm 2 Risk-Behaviour Feature Learning

1. Driving behaviour features, $x_i^{(b)} = \{f_n\}$;
2. Imbalanced data resampling (safe class under-sampling);
3. Feature filtering by importance ranking:
 - a. Train/tune model (XGBoost) using all features $\{f_n\}$;
 - b. Calculate relative importance scores, $\{r_n\}$;
 - c. Filtering by thresholds:
 1. Define thresholds τ ;
 2. For each τ , do:
 - a. Remove $f_n \forall r_n < \tau$;
 - b. Obtain subset S with remaining features;
 - c. Re-training with S ;
 - d. Performance evaluation $E(S)$;
 - c. Select $S^* = \operatorname{argmax}_S E(S)$.
4. RFE:
 - a. N is the feature size of S^* ;
 - b. For $n = N, \dots, 2$, do:
 1. Permute n time, for $k = 1, \dots, n$:
 - a. Remove $f_k^{(n)}$, obtain S' ;
 - b. Re-training with S' ;
 - c. Obtain $E(S')$;
 2. Eliminate f_k^* , for max $E(S')$;
 3. Obtain S'' with $n - 1$ important features, and $E(S'')$.
 - c. Get $\operatorname{argmax} E(S'')$ and key features.
5. Behaviour-based risk prediction.

loss, making it easier to optimise than the actual objective function [13]. Besides, by reasoning from the past search results, the next trials can concentrate on more promising values, which reduces the number of trials while finding a good optimum. The Bayesian hyperparameter optimisation is represented as:

$$x^* = \operatorname{argmin}_{x \in \chi} f(x) \quad (1)$$

$$EI_{y^*}(x) = \int_{-\infty}^{y^*} (y^* - y) p(y | x) dy \quad (2)$$

where $f(x)$ represents the loss of the objective function to minimise, and the cross-entropy loss (log loss) is used for a more nuanced evaluation. y is the actual value of the objective function using hyperparameter x , and χ is the domain space of hyperparameter values to search over. A probability surrogate model $p(y | x)$ is formed to produce a predictive posterior distribution over the performance of the past evaluations. According to demonstrated efficiency and pre-tested performance, this study adopts the Tree-structured Parzen Estimator (TPE) to build the surrogate model (i.e., response surface), and guide the exploration of the domain space [19], [20]. The values with the highest expected improvement (EI) are selected for the next trials, which are expected to potentially maximise the increase in the performance.

The performance is estimated by a set of metrics, scored on testing datasets via stratified cross-validation. For imbalanced classification, recall, precision and AUPRC (area under the precision-recall curve) are recommended [15], [21]. Additional metrics include accuracy and AUC (area under the receiver

TABLE I
AV CONFIGURATIONS AND FEATURE SETS

Configuration	Sensors	Variables for feature extraction	Feature set
Ego vehicle standalone (movement behaviours only)	OBD (on-board diagnostics); position system + IMU (inertial measurement unit), cameras, etc.	Velocity, acceleration, lateral position, lane tracking, steering angular speed, vibration, etc.	I
Basic object detection and distance measurement	Basic sensor fusion of: vision odometry; LiDAR point cloud; Radar; semantic map, etc.	Front gap, relative distance, etc.	II
Advanced surrounding sensing and high-quality measurement	Advanced sensor fusion of: vision odometry; LiDAR point cloud; Radar; semantic map, etc.	Derived variables related to vehicle pairs, interactions and driving scenes, etc.	III
Add-on connected vehicle (CV) communication	V2V; V2I; roadside sensing, etc.	Information related to vehicle platoon and traffic conditions (e.g., average velocity/ or gap of vehicle stream)	IV

operating characteristic curve). The metrics are calculated for each class, and macro-averaged mean (over all classes) and weighted mean (based on the number of true instances for each class) values are calculated, respectively [21].

III. ANALYSIS

A. AV Configurations From Risk Assessment Viewpoints

The proposed AutoML aims to predict risk levels of vehicles in driving and route planning based on behaviour features. However, due to different data availability under different sensing quality, the extracted feature sets are different. Herein, four AV configurations are considered to represent various sensing information acquisition conditions, and the model performances under each condition are benchmarked. Meanwhile, data-driven insights about AV configurations from risk prediction viewpoints is another purpose of this study.

The data availability about feature extraction involving four AV sensing configurations are listed in Table I. First, the AutoML using data involving individual standalone vehicle is tested as a basis for comparison, which is to predict risk levels from behaviour characteristics of ego vehicle only, hence the added values of AVs in safety enhancement can be quantified from a risk decision-making perspective. Second, a basic AV sensing configuration can detect and measure the position of preceding vehicles, in which extra features are extracted from distance-based variables (e.g., relative position, or front gap). Third, with high-quality sensor perception and advanced fusion (e.g., AV Level 4 and above), more variables and finer features can be derived for risk assessment and driving scene understanding, such as using the relative relationships with the ego vehicle to measure the behaviours of surrounding vehicles. Furthermore, if the AVs also configure CV (connected vehicles) functions, additional information is available from

V2I (vehicle-to-infrastructure) and/or V2V (vehicle-to-vehicle) communication. The AutoML performance and scalability in these four risk prediction scenarios are demonstrated.

B. Data

As a test scenario, the vehicle trajectory data provided in FHWA Next Generation Simulation (NGSIM) programme is used as the surrogate data, to emulate the safe and risk conditions that AVs could face in naturalistic flow (e.g., presuppose some safe behaviour vehicles as AVs). The NGSIM program collected detailed real vehicle trajectory data, using several synchronised video cameras, mounted on top of high buildings adjacent to the roadway [22], [23]. NGSIM data recorded the main movement variables of all vehicles in a 640-metre road segment for about 45 minutes, including vehicle trajectory, length, instantaneous velocity and acceleration, etc. After data cleaning, a total of 5,084 instances (vehicles) from NGSIM are used for feature extraction. The data acquisition resolution is 0.1 seconds in relative units.

Herein, surrogate sensing data (e.g., front gap) is calculated using the trajectory data of vehicle pairs, and surrogate connected vehicles data is simulated using the data of all vehicles in the traffic flow. For instance, the front gap data is calculated based on the trajectory and length of the preceding and following vehicles. Data preprocessing uses the Savitzky-Golay filter to smooth out potential noise in data acquisition without much information loss due to cutting off peaks.

C. AutoML Pipeline Architecting

This section elaborates the main steps of the AutoML pipeline, in particular, data labelling and feature learning adapted for various AV configurations. AutoML auto-tuning and risk prediction are described in Sections III.D and III.E, respectively.

Step 1: Unsupervised risk identification and labelling of risk exposure levels

The AutoML pipeline uses the FCM clustering algorithm to assess risk levels based on surrogate indicators. The main surrogate risk indicators are TIT (time integrated time-to-collision) and CPI (crash potential index), and several threshold settings are used to cover various sensitivity on risk severity. TIT and CPI are found to be more helpful in pre-accident risk identification, based on a previous evaluation using real-world accident cases [17]. For TIT, the time-to-collision threshold values are 3s, 4s, 5s, respectively, and CPI is measured by two types of MADR (maximum available deceleration rate) calculation.

FCM clustering divides a large scope of vehicles within a road segment into different groups, based on the pattern similarity measured by risk indicators. Given the lack of ground truth labels, several numbers of clusters (i.e., hyper-parameter k) are considered. For each k value, clustering partition with k risk levels are obtained, and the clustering results are evaluated by testing the fitting performance (e.g., cross-validation accuracy) of the clustering partitions and indicator values using independent classifiers. Hence, a clustering result with the best fitting performance can be selected.

TABLE II
DRIVING BEHAVIOUR FEATURE EXTRACTION

	Coded features* (with counts)	Counts
Total		1,357
Ego vehicle related (Feature set I)	{vel; acc; jerk}.{max; mean; min; p01; p05; p95; p99; q1; q2; q3; std}(33); {vel; acc}.{kurt; mad; skew} (6); {vel; acc; y; y'}.{pet}.{ max; mean; min; p01; p05; p95; p99; q1; q2; q3; std; absm}(48); {y'; vel}.{logr}.{ max; mean; min; p01; p05; p95; p99; q1; q2; q3; std; absm}(24); {vel; acc}.{wl; w2; w3}.{rng; crng; sma; msd; rsd; emar}.{mean; std; max; min}(144); y.{std}(1); lane.{mean; std; rng}(3).	259
Gap related features	gap.{kurt; mad; max; mean; min; p01; p05; p95; p99; q1; q2; q3; skew; std}(14); gap.{pct; logr}.{max; mean; min; p01; p05; p95; p99; q1; q2; q3; std; absm}(24); gap.{wl; w2; w3}.{rng; crng; sma; msd; rsd; emar}.{mean; std; max; min}(72).	110
Vehicle pair related	pv.[feature set I](259); {vel; acc; x}.{dtw}(3); {vel; acc; x}.{w1; w2; w3}.{dtw}.{mean; std; p05; q1; q3; p95; max; min}(72); dif.{acc; vel}.{max; mean; min; p01; p05; p95; p99; q1; q2; q3; std}(22); dif.{vel; acc}.pct.{absm; max; mean; min; p01; p05; p95; p99; q1; q2; q3; std}(24); dif.{vel; acc}.{w1; w2; w3}.{rng; crng; sma; msd; rsd; emar}.{mean; std; max; min}(144); {vel; acc; tv; tpv}.{pcor; scor}(8); {vel; acc; tv; tpv}.{w1; w2; w3}.{pcor; scor}.{mean; std; p05; q1; q2; p95; max; min}(192).	724
Vehicle stream related	{acc; gap; vel; jerk}.vfr.{w0; w1; w2; w3}.{max; mean; min; p01; p05; p95; p99; q1; q2; q3; std}(176); dif.{acc; vel}.vfr.{w0; w1; w2; w3}.{max; mean; min; p01; p05; p95; p99; q1; q2; q3; std}(88).	264

* Abbreviations and explanations about the coded features are listed in the Appendix.

According to a previous study [15], four risk levels are defined based on the clustering of 5 groups. Based on pattern decoding from the risk indicator values, the annotation of the risk labelling can be assigned. One annotation of the risk labelling is the safe level (with 3,653 instances, about 71.85% of total data), low-risk level (LR; with 900 instances), moderate-risk level (MR; with 425 instances), and high-risk level (HR; with 106 instances, combined based on two clustered groups with 98 and 8 higher-risk instances, respectively). The safe level indicates the lowest likelihood to be involved in traffic conflicts, and vice versa. Hence, each vehicle can be labelled using the risk level of the respective cluster.

Considering that the clustered safe-risk grouping is highly imbalanced, the dataset is further processed by safe-class data undersampling using the Repeated Edited Nearest Neighbours (RENN) algorithm, which generates datasets with a more balanced class distribution [15]. Undersampling using RENN removes similar instances in the safe level class, in which 1,211 instances are selected from initial 3,653 safe-class instances. Thus, the total data size drops from 5,084 to 2,642.

Step 2: Driving behaviour feature extraction

Massive driving behaviour features are extracted from vehicle trajectory and surrogate sensing data, as listed in Table II.

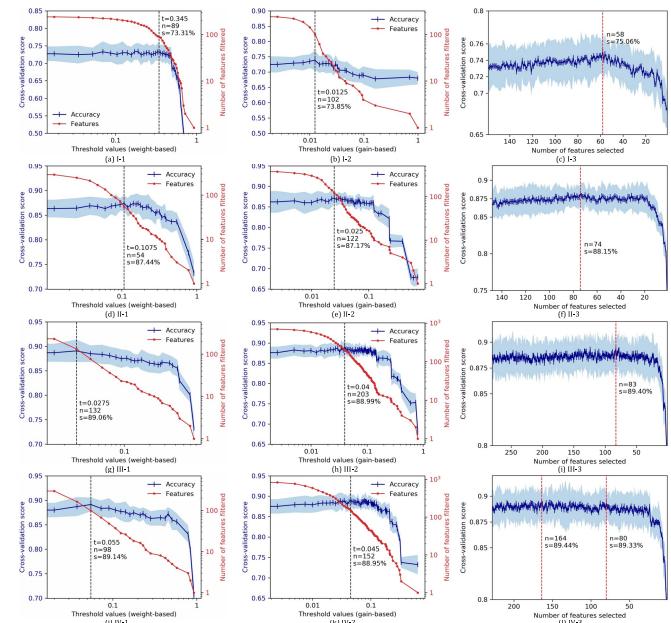


Fig. 3. Feature self-selection and learning performance.

The detailed processing of feature extraction can be found in Shi *et al.* [15]. The interpretations of feature extraction and related terminologies are presented in the Appendix.

Step 3: Learning-based feature selection

Auto-selection of the best feature combinations is an advantage of the proposed AutoML, which can improve modelling performance and reduce computation cost. The feature self-selection is based on the fitting of behaviour features and corresponding risk levels by XGBoost, and the importance of each feature and optimal feature combinations can be identified. For the purpose of feature ranking, an appropriate XGBoost model is configured with basic hyperparameter settings, and feature subsampling ratio is set to 1.0. In the process of XGBoost training, the split weight and average gain for each feature in tree building are generated, which are normalised to calculate the weight-based and gain-based relative importance scores, respectively. For efficient feature filtering, a range of threshold values is defined, which forms a series of feature combinations with the importance scores greater than the threshold value. The feature self-selection process and corresponding learning performance of four AV configurations/ feature sets are estimated, as shown in Fig. 3. The number of selected features and accuracy are listed in Table III, while noting that some of the features are duplicated in the two filtered feature subsets. Weight-based selection generally favours features with more classes, and gain-based selection is biased towards the ones with stronger signals (e.g., Gini impurity).

The RFE step further considers the interactions among the filtered features, then an optimal feature combination is identified by model re-training and recursively pruning the features with the least permutation importance from the current set. The learning performance of each iteration is shown in Fig. 3. The selection is based on a trade-off of complexity and performance, for example, one with less complexity,

TABLE III

NUMBER OF FEATURES SELECTED BY TREE-BASED FILTERING AND RFE

Feature set	Initial size	Weight-based	Gain-based	Total filtered	RFE suggested
I	259	90 (73.31%*)	102 (73.85%)	160	58 (75.06%)
II	369 (I+110)	54 (87.44%)	122 (87.17%)	149	74 (88.15%)
III	1,093 (II+724)	132 (89.06%)	203 (88.99%)	281	83 (89.40%)
IV	1,357 (III+264)	98 (89.14%)	152 (88.95%)	229	80 (89.33%)

* Corresponding learning accuracy.

a modest decrease (<0.1% of the best) in estimated accuracy, and lower variance.

D. Auto-Tuning by Bayesian Optimisation

The risk levels of vehicles in driving are predicted based on the selected key behaviour features. Herein, the modelling of behaviour-based risk prediction is improved in two aspects, namely, model training using key features, and hyperparameter auto-tuning by Bayesian optimisation. XGBoost is a type of ensemble learning configured with boosted decision trees. Tree hyperparameters can directly control model complexity, e.g., maximum tree depth, splitting weight, etc. The learning rate tunes the boosting process by weighting, which makes fitting more conservative. Bagging hyperparameters reduce the variance by decorrelation, which improve the model robustness against noise, e.g., random subsampling of instances and features. The hyperparameters and domain distributions are listed in Table IV.

The domain space of hyperparameters over which to search is created as centred around the pre-tested values and is then refined in subsequent searches [15]. The logarithmic uniform distribution is used for the learning rate because it varies across several orders of magnitude. The sampling of values in the domain is equally likely (uniform). The number of estimators is set to 300, but this number will not always be reached because early stopping is used to stop the training when validation scores have not improved for 30 iterations (i.e., 10% of total estimators).

Herein, the objective function is to minimise the log loss of the XGBoost model using specific hyperparameters, via 10-fold stratified cross-validation. The surrogate model of the objective function and the selection function for evaluating which hyperparameter values to choose next are based on TPE, and the setting of 500 iteration trials is used.

The optimised hyperparameter values and corresponding performance scores are listed in Table V. The performance and hyperparameters versus the iterations are plotted to inspect the auto-tuning process, as shown in Fig. 4. The dark triangles indicate top-five optimal values. The average validation scores increase over time (conversely the loss decreases) as expected, indicating the method is trying better hyperparameter values. As the search progresses, the auto-tuning switches from exploration (e.g., trying new values) to exploitation (e.g., selecting values with better past results), which is more efficient

TABLE IV
KEY HYPER-PARAMETERS AND DOMAIN SPACE

Hyperparameter	Description	Distribution	Domain
1. ENSEMBLE HYPERPARAMETERS			
Learning rate	Shrink the feature weights of each boosting step	Continuous log uniform	(0.05, 0.3)
Number of estimators	Number of boosted trees added in model	-	300
2. BOOSTED TREE HYPERPARAMETERS			
Tree depth	Maximum depth of a tree	Discrete uniform (integers spaced evenly)	[3, 4, 5, 6]
Splitting weight	Further partitioning of a leaf node in tree building process	Continuous uniform	(0.6, 1.0)
3. SUBSAMPLING HYPERPARAMETERS			
Instance subsample ratio	Random sample of the training data prior to growing trees in every boosting iteration	Continuous uniform	(0.4, 1.0)
Feature subsample ratio	Random sample of features for each split in tree level	Continuous uniform	(0.4, 1.0)
4. REGULARISATION HYPERPARAMETERS			
Gamma	Minimum split loss reduction required to make a further partition on a leaf node	Continuous uniform	(0, 1.0)
Alpha	L1 regularisation term on weights	Continuous uniform	(0, 1.0)
Lambda	L2 regularisation term on weights	Continuous uniform	(0, 1.0)

TABLE V
OPTIMISED HYPER-PARAMETERS AND PERFORMANCE SCORES

	Feature set			
	I	II	III	IV
1. HYPERPARAMETERS				
Learning rate	0.056	0.086	0.064	0.161
Number of estimators (early stop iteration)	116	153	264	90
Tree depth	6	5	4	5
Splitting weight	0.698	0.628	0.829	0.949
Instance subsample ratio	0.936	0.829	0.677	0.948
Feature subsample ratio (by tree)	0.668	0.833	0.854	0.656
Feature subsample ratio (by level)	0.842	0.645	0.945	0.673
Gamma	0.530	0.517	0.472	0.116
Alpha	0.520	0.934	0.432	0.999
Lambda	0.038	0.999	0.743	0.410
2. PERFORMANCE				
Misclassification	848	387	338	334
Accuracy	0.734	0.870	0.880	0.883
Accuracy (converted)	0.808	0.907	0.915	0.917
AUPRC (macro)	0.646	0.879	0.894	0.901
AUPRC (weighted)	0.777	0.931	0.939	0.943
AUC (macro)	0.870	0.972	0.977	0.978
AUC (weighted)	0.918	0.979	0.983	0.984

compared to uninformed random or grid search methods. The dependence of loss with each hyperparameter is measured by mutual information (MI, also known as information gain), as shown in Fig. 5. Key hyperparameters with more information are identified, especially the learning rate, instance subsample, estimators and tree depth. The loss-hyperparameter distribution relationships are plotted in Fig. 6. There are noticeable trends that the Bayesian optimisation tends to concentrate (i.e., place more probability) the search on evaluating

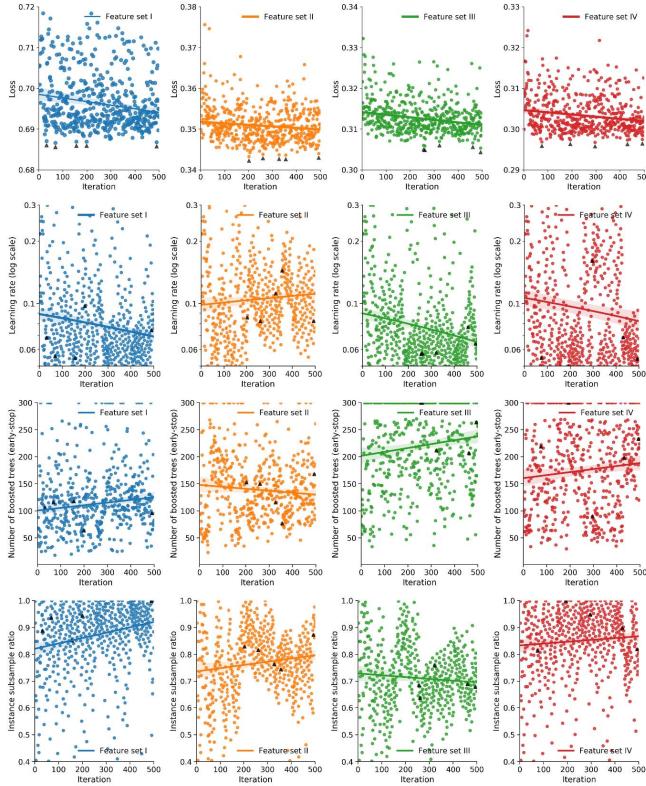


Fig. 4. Hyperparameters auto-tuning and corresponding performance.

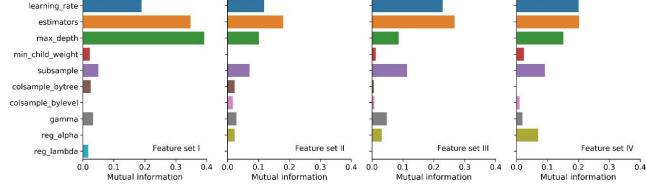


Fig. 5. Hyperparameter comparison by mutual information of loss.

more promising values. The identification of key hyperparameters and value distribution contributes to the better design of domain-specific AutoML.

After the Bayesian optimisation phase, an ensemble of several versatile models is constructed, by iteratively adding the model that maximises ensemble validation performance. The automatic ensemble is more robust and less prone to be over-fitting, compared in favour of selecting one best model. With the optimised features and hyperparameters, the final model ensemble is generated by re-training using the whole data set.

E. Prediction Performance and Key Features

The potential capacity of risk prediction under the four AV configurations (corresponding to four feature sets) is compared. AutoML provides the optimised models for each AV configuration. From Table V, AutoML achieves satisfactory results of behaviour-based risk prediction. The converted accuracy is about 91.7%, which is converted based on the raw data distribution before safe-class undersampling. The detailed prediction performances for each risk level and AV configurations

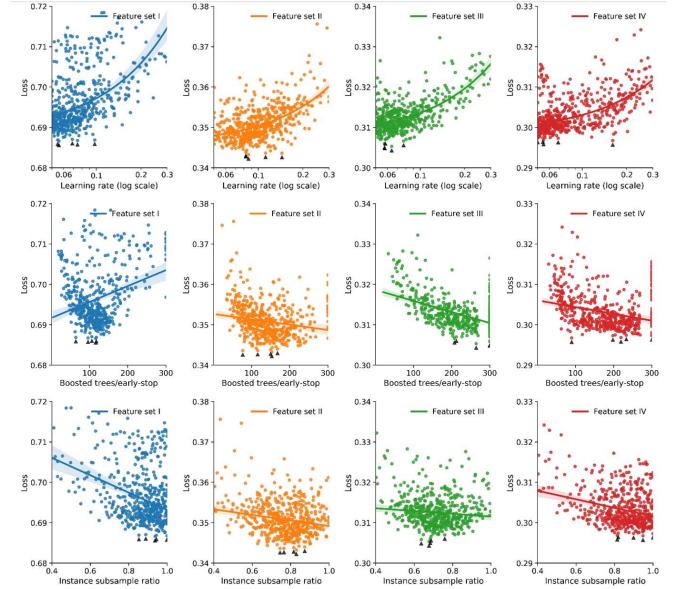


Fig. 6. Loss versus key hyperparameters.

TABLE VI
PREDICTION EVALUATION FOR DETAILED RISK LEVELS

Feature set	Risk levels	Precision	Recall	AUC	AUPRC
I	Safe	0.839	0.888	0.939	0.927
	LR	0.639	0.797	0.857	0.733
	MR	0.618	0.296	0.830	0.541
	HR	0.647	0.208	0.854	0.385
II	Safe	0.926	0.947	0.987	0.985
	LR	0.844	0.852	0.958	0.926
	MR	0.787	0.739	0.960	0.824
	HR	0.729	0.660	0.983	0.779
III	Safe	0.938	0.952	0.990	0.989
	LR	0.856	0.866	0.966	0.938
	MR	0.774	0.767	0.967	0.831
	HR	0.817	0.632	0.986	0.817
IV	Safe	0.940	0.955	0.990	0.990
	LR	0.856	0.867	0.967	0.938
	MR	0.788	0.760	0.967	0.846
	HR	0.813	0.698	0.987	0.831

(i.e., various feature sets) are estimated in Table VI. The detailed partition of risk levels helps to uncover potential risk conditions from the data without real accident cases, and the predictive power of greater than 95% accuracy is expected for safe-risk distinction.

The prediction performances of various feature sets provide data-driven insights about AV sensing configurations from the perspective of the information needed for risk decision-making. Compared with basic features (feature set I), the gap-related features (feature set II) can vastly improve the performance of the higher risk levels, especially the recall of MR and HR. Hence, accurate front gap measurement is critical for risk decision-making. For high-quality measurement about vehicle pair relationships (feature set III), there is about 12% improvement on the precision of the HR level, and about 2% improvement on the overall performance. Additionally, the information about CV (feature set IV) helps to improve about 10% on the recall of HR prediction, but for other metrics, the added values are not obvious. Thus, the surrounding

TABLE VII
MAIN VARIABLES AND FUNCTIONS INVOLVED IN
KEY FEATURES SELECTED

	Feature set			
	I	II	III	IV
Number of selected key features	58	74	83	80
Variables in key features selected (with counts)	vel (30); acc (19); y (6); jerk (3)	gap (34); vel (27); acc (8); y (4), etc.	vel (39); gap (24); pv (19); acc (13); y (6), etc.	gap (32); vel (31); pv (13); acc (11); y (2), etc.
Functions in key features selected (with counts)	rng (12); w3 (12); w1 (11); pct (10); crng (8); logr (7); sma (5); w2 (5); rsd (4); emar (4); msd (3)	w1 (21); rng (18); pct (14); w2 (11); crng (10); logr (9); w3 (7); sma (5); msd (5); rsd (3); pcor (3); etc.	w1 (20); dif (15); pct (13); sma (12); logr (12); w2 (8); rng (8); logr (12); pct (7); sma (8); crng (7); pcor (7); emar (5); scor (5); scor (4); rsd (3), etc.	w1 (21); vfr (18); w3 (18); dif (16); w2 (14); logr (12); pct (7); sma (5); crng (5); emar (5); scor (4); rsd (3), etc.

risk evaluation (e.g., behaviour comparison, risk force field, relative driving performance of preceding vehicles) is helpful to assess the behaviour of individual vehicles pertinently, and contributes to risk decision-making, such as, identify a trajectory with fewer likelihoods of conflicts caused by surrounding vehicles, or motion planning with defensive driving, etc.

The results of feature self-selection provide data-driven insights about the optimal processing and information mining of sensing data. The main variables and functions involved in the key features selected are listed in Table VII. For variables acquisition from sensing data, the gap, velocity and acceleration are the most informative variables to assess risk levels. In terms of behaviour data mining, this study designs a series of ratio functions, and most of them have appeared in the selected key features, such as log-ratio, percentage change, and bias ratio. These ratio functions help to measure the abnormal changes of movement, and capture potential risk signals. Moreover, different forms of ratio have different sensitivities to changes, hence multiple functions complement the feature pattern mining. Besides, data processing at shorter time intervals is also important, given that a lot of key features are defined based on small moving windows. In-depth data mining can enhance the reliability and predictability of the AutoML.

The tree-based AutoML with domain-specific features has the advantages of high transparency, being robust and fault-tolerant, etc. The AutoML can be integrated into the self-driving system to predict the risk levels of instantaneous motion trajectory, as well as evaluating AV behavioural decision-making based on benchmarks of clustered safe driving.

IV. DISCUSSION

A. Application Potentials

AutoML for predicting detailed risk levels is more challenging but valuable. Accurate assessment of risk levels

can improve the confidence level of AV decision-making, which complements intuition-like decision-making in end-to-end deep learning. Enhanced performance from the isolated validation indicates an accurate and reliable predictive power, whilst allowing the model to work well on data that is not used in the modelling. The proposed AutoML is helpful in the planning of non-collisional and optimal motion trajectory by adding a clear assessment of risk levels (e.g., accurate safe boundary and optimal risk buffer). Moreover, the risk levels provide early signals for crash potentials and likelihood, which can be used to plan trajectory and/or motion based on pre-emptive identification of unsafe conditions [24].

AutoML reveals the most important behaviour features used for risk assessment, which provides useful insights for information processing and sensor configuration, as well as interpretable risk decision-making mechanisms. Feature fusion and hybrid measures show good performances (e.g., low false alarms), and redundancy risk assessment is desirable to promote system reliability. For AVs and ADAS (advanced driver-assistance systems), the AutoML for behaviour-based risk prediction can be deployed as a co-pilot system, which can be used to offer reliable and appropriate risk warning and behaviour assessment, including the risk level estimation of surrounding vehicles. Conversely, for human-AV interaction, if the detected responses of drivers to hazards are inconsistent with the predicted risk levels, the system would take over control of the vehicle to reduce the crash risk. Besides, reliable risk prediction and behaviour assessment are important for improving driver trust in the AV system.

For practical application with computation constraints, the AutoML also supports offline warm start using an ensemble of pre-trained models. The meta-characteristics of new datasets are first captured to identify and match similar datasets in the known data space [25]. Thus, similar pre-trained models can be ensembled and work directly, or the AutoML training can start with similar configurations to seed the Bayesian optimisation. This can boost the overall modelling within a reasonable amount of time or computation limits. Besides, in cognisance of resource limitations, the trade-off between optimisation time versus the settings of pipelines can also be tested.

B. Limitations

Since the data about real-world traffic flow involving AVs is not massively collectable currently, this study used surrogate data to emulate the scenes and data available under AV and CV environments. The NGSIM data is about human driver vehicles, of which the behaviours are used to train AV driving. Moreover, the model performance is highly dependent on data quality and quantity. Since the size of the dataset is relatively very small, the benefits of hyperparameter optimisation are not well presented. Besides, the improvement may have diminishing returns, because of noisy data, small size, and hidden variables that are not measured [26].

Bayesian optimisation is effective, but not guaranteed to find the best hyperparameters, since it has the risk of getting stuck in a local minimum of the objective function. This can

be checked by starting an entirely different search. If the subsequent searches focus on similar values, such values are expected to be optimal. Although the random search does not suffer from this issue, given the high dimensionality and complex interactions between hyperparameters, the random search is much more computation expensive.

For the AutoML pipelines, one challenge is to verify the unsupervised risk assessment, since there is a lack of crash instances in the raw dataset. Further validation is needed, such as by examining the accident records and insurance claims. To further improve modelling, in-depth feature extraction and high-quality data are suggested, which may involve a broader range of risk potentials (e.g., accident cases and near-miss conditions) and driving scenarios (e.g., lane-changing [27]). The feature extraction helps to make risk assessment more interpretable and reliable, as well as providing early signals for risk mitigation.

V. CONCLUSIONS

The end-to-end AutoML is effective and flexible to predict detailed risk levels based on driving behaviour and (surrogate) sensing data, which can benefit the risk decision-making and motion trajectory planning of AVs, and provide data-driven insights about AV system design.

The AutoML integrates the main steps of risk prediction into an auto-optimisable pipeline, including unsupervised risk identification by FCM clustering and risk indicators, risk-behaviour feature learning by the hybrid of XGBoost-based filtering and RFE, imbalanced data resampling, model selection and hyperparameter tuning by Bayesian optimisation, among others. Two main mechanisms are designed to improve the modelling performance, one is the massive feature extraction and learning-based selection, another is the hyperparameter auto-tuning. Massive in-depth features are extracted to capture more useful information on vehicle driving and risk potentials, and the feature subsets with the best modelling performance are self-learned by the AutoML. XGBoost is the key classifier of the AutoML, and interpretable risk decision rules can thus be generated using the tree-based AutoML.

AutoML achieves satisfactory results of behaviour-based risk prediction, which has a predictive power of 91.7% overall accuracy, and greater than 95% accuracy for safe-risk distinction. Based on the unified AutoML procedure, the prediction performances of various AV sensing configurations have been compared, which provides data-driven insights about AV safety from the perspective of the information needed for risk decision-making. Moreover, the AutoML reveals the most important behaviour features used for risk assessment, which uncovers useful insights about sensor data mining. Finally, additional application potentials and limitations are also discussed, such as risk prediction co-pilot system.

APPENDIX

Additional interpretation of feature extraction and related terminologies are shown in Table VIII.

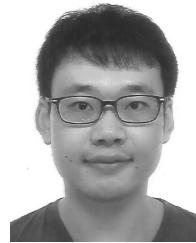
TABLE VIII
VARIABLES AND FUNCTIONS FOR FEATURE EXTRACTION

	Code	Description
Variables (pre-coded)		
Vehicle trajectory	x; y	Time series data, for vehicle i , $x_i(t)$ is longitudinal position defined by vehicle front centre; $y_i(t)$ is the lateral position
Trajectory of the preceding vehicle (pv)	pv.x; pv.y	$x_{i-1}(t)$; $y_{i-1}(t)$, for preceding vehicles (pv) $i - 1$
Lane	lane	Lane number of a vehicle travelling on
Vehicle type	class	0-motorcycle; 1-car; 2-truck
Vehicle length	L	L_{i-1} , vehicle length of preceding vehicle
Variables (derived)		
Velocity	vel	$v_i(t) = d[x_i(t)]/dt$
Acceleration	acc	$a_i(t) = d[v_i(t)]/dt$, negative value indicates deceleration
Jerk	jerk	$j_i(t) = d[a_i(t)]/dt$
Front gap	gap	Calculated by $x_{i-1}(t) - x_i(t) - L_{i-1}$
Variables of preceding vehicles	pv.vel; pv.acc; pv.jerk	$v_{i-1}(t); a_{i-1}(t); j_{i-1}(t)$
Functions (data mining)		
Moving time windows	w1; w2; w3; w0	Moving windows defined by time intervals of 1.0 s (w1), 5.0 s (w2), and 10 s (w3), 'w0' represents the full time period.
Difference	dif	$s_{i-1}(t) - s_i(t)$, measure difference of variable s between subject vehicle and preceding vehicles
Percentage change	pct	$\frac{s_i(t) - s_i(t-1)}{s_i(t-1)} * 100$, percentage change of a variable (per 1.0 second)
Log ratio	logr	Measure relative change on a logarithmic scale, per 1.0 second, calculated by $\log \frac{s_i(t)}{s_i(t-1)}$
Vehicle to flow ratio	vfr	Comparison between a vehicle and the average performance of vehicle platoon in the same lane segment
Range	rng	Calculated by $\max(s_i(t)) - \min(s_i(t))$
Coefficient of range	crg	Calculated by $\frac{\max(s_i(t)) - \min(s_i(t))}{\max(s_i(t)) + \min(s_i(t))}$
Simple moving average	sma	Mean value of the time series data within a moving window
Moving standard deviation	msd	Standard deviation of the time series data within a moving window
Relative standard deviation	rsd	Relative variability and unitised measure, defined as the ratio of std to mean
Bias ratio (exponential moving average ratio)	emar	Calculated by $\frac{s_i(t)}{s_i(t) - EMA_i(t)}$ where $EMA_i(t)$ is the exponential moving average of the data within a moving window defined by $(t - w, t)$
Dynamic time warping	dtw	Using dynamic time warping (DTW) algorithm to measure similarity between two temporal sequences
Correlation coefficient	scor; pcor	Compute pairwise correlation of two variables, by Spearman correlation (scor), and Pearson correlation (pcor). Besides, TTC-vel relationship (tv), and TTC-pv.vel relationship (tpv) are calculated to refer the responsibility in a conflict condition
Functions (descriptive)		
Basic centre and dispersion	mean; std	Values of mean and standard deviation
Extreme values	min; max; p01; p99	Values of minimum, maximum; also consider using 1 th and 99 th percentiles values (p01, p99) to deal with outliers and noise
Percentile values	p05; q1; q2; q3; p95	The 5 th , 25 th , 50 th , 75 th and 95 th percentiles to represent data profile and distribution pattern
Mean absolute deviation	mad	Measure variability or dispersion
Profile shape	krt; skw	Unbiased kurtosis (krt) over data using Fisher's definition; unbiased skew (skw), normalised by n-1
Absolute mean	absm	Mean of absolute value

REFERENCES

- [1] B. Okumura *et al.*, "Challenges in perception and decision making for intelligent automotive vehicles: A case study," *IEEE Trans. Intell. Vehicles*, vol. 1, no. 1, pp. 20–32, Mar. 2016.

- [2] B. Paden, M. Cap, S. Z. Yong, D. Yershov, and E. Frazzoli, "A survey of motion planning and control techniques for self-driving urban vehicles," *IEEE Trans. Intell. Vehicles*, vol. 1, no. 1, pp. 33–55, Mar. 2016.
- [3] S. Hecker, D. Dai, and L. van Gool, "End-to-end learning of driving models with surround-view cameras and route planners," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 435–453.
- [4] H. Xu, Y. Gao, F. Yu, and T. Darrell, "End-to-end learning of driving models from large-scale video datasets," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2174–2182.
- [5] C. Hubmann, J. Schulz, M. Becker, D. Althoff, and C. Stiller, "Automated driving in uncertain environments: Planning with interaction and uncertain maneuver prediction," *IEEE Trans. Intell. Vehicles*, vol. 3, no. 1, pp. 5–17, Mar. 2018.
- [6] Q. Wang, B. Ayalew, and T. Weiskircher, "Predictive maneuver planning for an autonomous vehicle in public highway traffic," *IEEE Trans. Intell. Transport. Syst.*, vol. 20, no. 4, pp. 1303–1315, Apr. 2019.
- [7] R. McAllister *et al.*, "Concrete problems for autonomous vehicle safety: Advantages of Bayesian deep learning," in *Proc. Int. Joint Conf. Artif. Intell.*, 2017, pp. 4745–4753.
- [8] M. Feurer, A. Klein, K. Eggensperger, J. Springenberg, M. Blum, and F. Hutter, "Efficient and robust automated machine learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 2962–2970.
- [9] K. Eggensperger, M. Lindauer, and F. Hutter, "Pitfalls and best practices in algorithm configuration," *J. Artif. Intell. Res.*, vol. 64, pp. 861–893, Apr. 2019.
- [10] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 785–794.
- [11] Q. Li, L. Chen, M. Li, S.-L. Shaw, and A. Nuchter, "A sensor-fusion drivable-region and lane-detection system for autonomous vehicle navigation in challenging road scenarios," *IEEE Trans. Veh. Technol.*, vol. 63, no. 2, pp. 540–555, Feb. 2014.
- [12] S. García, S. Ramírez-Gallego, J. Luengo, J. M. Benítez, and F. Herrera, "Big data preprocessing: Methods and prospects," *Big Data Analytics*, vol. 1, no. 1, p. 9, Dec. 2016.
- [13] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. de Freitas, "Taking the human out of the loop: A review of Bayesian optimization," *Proc. IEEE*, vol. 104, no. 1, pp. 148–175, Jan. 2016.
- [14] C. Xu, W. Wang, and P. Liu, "A genetic programming model for real-time crash prediction on freeways," *IEEE Trans. Intell. Transport. Syst.*, vol. 14, no. 2, pp. 574–586, Jun. 2013.
- [15] X. Shi, Y. D. Wong, M. Z.-F. Li, C. Palanisamy, and C. Chai, "A feature learning approach based on XGBoost for driving assessment and risk prediction," *Accident Anal. Prevention*, vol. 129, pp. 170–179, Aug. 2019.
- [16] C. Katrakazas, M. Quddus, W.-H. Chen, and L. Deka, "Real-time motion planning methods for autonomous on-road driving: State-of-the-art and future research directions," *Transp. Res. C, Emerg. Technol.*, vol. 60, pp. 416–442, Nov. 2015.
- [17] X. Shi, Y. D. Wong, M. Z. F. Li, and C. Chai, "Key risk indicators for accident assessment conditioned on pre-crash vehicle trajectory," *Accident Anal. Prevention*, vol. 117, pp. 346–356, Aug. 2018.
- [18] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *J. Mach. Learn. Res.*, vol. 3, pp. 1157–1182, Jan. 2003.
- [19] J. Snoek, H. Larochelle, and R. P. Adams, "Practical Bayesian optimization of machine learning algorithms," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 2951–2959.
- [20] J. S. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for hyper-parameter optimization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2011, pp. 2546–2554.
- [21] J. Lever, M. Krzywinski, and N. Altman, "Classification evaluation," *Nature Methods*, vol. 13, no. 8, pp. 541–542, 2016.
- [22] J. Colyar and J. Halkias, "Next Generation SIMulation fact sheet," US Dept. Transp. FHWA, Washington, DC, USA, Tech. Rep. FHWA-HRT-06-135, Dec. 2006.
- [23] V. Punzo, M. T. Borzacchiello, and B. Ciuffo, "On the assessment of vehicle trajectory data accuracy and application to the next generation SIMulation (NGSIM) program data," *Transp. Res. C, Emerg. Technol.*, vol. 19, no. 6, pp. 1243–1262, Dec. 2011.
- [24] J. Wang, J. Wu, X. Zheng, D. Ni, and K. Li, "Driving safety field theory modeling and its application in pre-collision warning system," *Transp. Res. C, Emerg. Technol.*, vol. 72, pp. 306–324, Nov. 2016.
- [25] M. Feurer, J. T. Springenberg, and F. Hutter, "Using meta-learning to initialize Bayesian optimization of hyperparameters," in *Proc. Int. Conf. Meta-Learn. Algorithm Selection*, vol. 1201, 2014, pp. 3–10.
- [26] F. L. Mannerling, V. Shankar, and C. R. Bhat, "Unobserved heterogeneity and the statistical analysis of highway accident data," *Analytic Methods Accident Res.*, vol. 11, pp. 1–16, Sep. 2016.
- [27] T. Chen, X. Shi, Y. D. Wong, and X. Yu, "Predicting lane-changing risk level based on vehicles' space-series features: A pre-emptive learning approach," *Transp. Res. C, Emerg. Technol.*, vol. 116, Jul. 2020, Art. no. 102646.



Xiupeng Shi received the B.Eng. and M.Eng. degrees from Beijing Jiaotong University, China, in 2010 and 2012, respectively, and the Ph.D. degree from Nanyang Technological University (NTU), Singapore, in 2019.

From 2012 to 2015, he worked as a Business General Manager. He is currently a Research Fellow with the Centre for Infrastructure Systems, NTU. His primary research interests include risk prediction, data science, automated machine learning, smart mobility, and supply chain finance.



Yiik Diew Wong received the B.E. degree in civil engineering and the Ph.D. degree in transportation engineering from the University of Canterbury, New Zealand, in 1983 and 1990, respectively.

He is currently an Associate Chair (Academic) and an Associate Professor with the School of Civil and Environmental Engineering, Nanyang Technological University (NTU), Singapore. He is also the Director of the Centre for Infrastructure Systems, NTU. His research interests include sustainable urban mobility, road safety engineering, driver behaviors, and active mobility.



Chen Chai (Member, IEEE) received the B.Eng. degree from Beijing Jiaotong University, China, in 2010, the M.Sc. degree from The Hong Kong University of Science and Technology, Hong Kong, in 2011, and the Ph.D. degree from Nanyang Technological University, Singapore, in 2015.

She is currently an Associate Professor with the College of Transportation Engineering, Tongji University, China. Her research interests include human-vehicle cooperative driving systems and mixed traffic flow of manual vehicles and autonomous vehicles. She is also a member of the Transportation Research Board Standing Committee ACH50: Human Factors and Behavioral Research Methods. She was a recipient of the Shanghai Sailing Talent Award and the Shanghai Chengguang Talent Award in 2018.



Michael Zhi-Feng Li received the B.Sc. degree in mathematics from Beijing Normal University, China, in 1983, the Ph.D. degree in mathematics from the University of Regina, Canada, in 1989, and the Ph.D. degree in business from The University of British Columbia, Canada, in 1994.

He is currently an Associate Professor with the Nanyang Business School, Nanyang Technological University (NTU), Singapore. He is also the Director of the Future China Advanced Leadership Programme and the Nanyang Executive MBA Programme in NTU. His research interests include operations research, transport economics, revenue management, and network congestion pricing.