Accident Analysis and Prevention 129 (2019) 170–179

Contents lists available at ScienceDirect

Accident Analysis and Prevention

journal homepage: www.elsevier.com/locate/aap

A feature learning approach based on XGBoost for driving assessment and risk prediction

Xiupeng Shi[a,*], Yiik Diew Wong[a], Michael Zhi-Feng Li[b], Chandrasekar Palanisamy[c], Chen Chai[d]

[a] School of Civil & Environmental Engineering, Nanyang Technological University, 639798, Singapore
[b] Nanyang Business School, Nanyang Technological University, 639798, Singapore
[c] Land Transport Authority (LTA), 219428, Singapore
[d] College of Transportation Engineering, Tongji University, 201804, China

ARTICLE INFO

Keywords:
Driving behaviour
Feature learning
XGBoost
Risk prediction

ABSTRACT

This study designs a framework of feature extraction and selection, to assess vehicle driving and predict risk levels. The framework integrates learning-based feature selection, unsupervised risk rating, and imbalanced data resampling. For each vehicle, about 1300 driving behaviour features are extracted from trajectory data, which produce in-depth and multi-view measures on behaviours. To estimate the risk potentials of vehicles in driving, unsupervised data labelling is proposed. Based on extracted risk indicator features, vehicles are clustered into various groups labelled with graded risk levels. Data under-sampling of the safe group is performed to reduce the risk-safe class imbalance. Afterwards, the linkages between behaviour features and corresponding risk levels are built using XGBoost, and key features are identified according to feature importance ranking and recursive elimination. The risk levels of vehicles in driving are predicted based on key features selected. As a case study, NGSIM trajectory data are used in which four risk levels are clustered by Fuzzy C-means, 64 key behaviour features are identified, and an overall accuracy of 89% is achieved for behaviour-based risk prediction. Findings show that this approach is effective and reliable to identify important features for driving assessment, and achieve an accurate prediction of risk levels.

1. Introduction

Reliable accident prediction and proactive prevention are undoubtedly of great benefit and necessity. Accident occurrence is a complex mechanism, with many contributing factors (Mannering et al., 2016). Generally, driver-centric factors could be found in most crash accidents, and driving behaviour assessment is an important aspect to enhance safety and reduce crashes. There is a perennial quest about assessing driving behaviour and predicting crash risk potentials in driving.

Many studies have been conducted to evaluate driving behaviour. Typical approaches include self-reported questionnaires, simulator-based experiments, and naturalistic driving studies (NDS) (Hong et al., 2014). In NDS, various characteristics and variables about both driving and drivers' behaviours are investigated, based on detailed information recorded using sensors, in-vehicle devices, and even smartphones (Eftekhari and Ghatee, 2018). A range of features has been developed to describe unsafe behaviours, such as speeding, abrupt braking or jerk, tailgating, frequent and intense lane change, yaw, among others (Bagdadi, 2013; Wahlström et al., 2017). Generally, statistical profiles of movement-related variables are used as behaviour features. For reliable driving assessment and risk prediction, in-depth and multi-view mining of features are necessary, especially features with predictability.

Data labelling of risk levels is another challenging but valuable work. As a complement to accident data, surrogate measures of vehicle conflicts are well accepted for safety evaluation (Zheng et al., 2014; Chai and Wong, 2015a). Many indicators have been developed to re-

* Corresponding author.
E-mail addresses: XSHI004@e.ntu.edu.sg (X. Shi), CYDWONG@ntu.edu.sg (Y.D. Wong), ZFLI@ntu.edu.sg (M.Z.-F. Li), Chandrasekar@lta.gov.sg (C. Palanisamy), chaichen@tongji.edu.cn (C. Chai).

https://doi.org/10.1016/j.aap.2019.05.005
Received 30 November 2018; Received in revised form 16 March 2019; Accepted 5 May 2019
Available online 30 May 2019

flect certain views of risk potentials, such as Time to Collision (TTC). Shi et al. (2018) retrieved two real-world accident cases, and developed hybrid indicators to identify pre-accident risk signals. Besides, kinematic characteristics are also commonly used for risk evaluation (Wu and Jovanis, 2013). For example, a rapid evasive manoeuvre is used to flag a near-crash event (Perez et al., 2017). However, detailed risk levels are inherently problematic to determine, since accurate classification and well-fitting thresholds are difficult to establish. A clear and synthetical assessment of risk levels is therefore of great interest, but is still lacking.

Furthermore, risk assessment is a distinctly imbalanced problem. The class imbalance issue has been discussed in various concepts, such as the safety pyramid model suggested by Hydén (1987). Algorithms often display bias in favour of the numerical majority class (*i.e.*, safe and low-risk cases), and ignore or wrongly discard the minority instances (*i.e.*, events of higher risk levels), which might be treated as noise or outliers (Díez-Pastor et al., 2015). Besides, there are five main observed challenges in class imbalance, related to intrinsic data characteristics (López et al., 2013; Beyan and Fisher, 2015). Class imbalance problems make the behaviour assessment and risk prediction much more complicated.

The focus of this study is to extract and select behaviour features for driving assessment and risk prediction using machine learning. The methodology is introduced in Section 2. Section 3 elaborates on extracting driving behaviour features and risk indicator features. The learning-based feature selection and risk prediction are described in Section 4. The final two sections cover the discussion and conclusions.

2. Methodology

2.1. Machine learning framework

A machine learning framework is designed to select key behaviour features and predict risk levels, which integrates learning-based feature selection, unsupervised risk rating, and imbalanced data resampling.

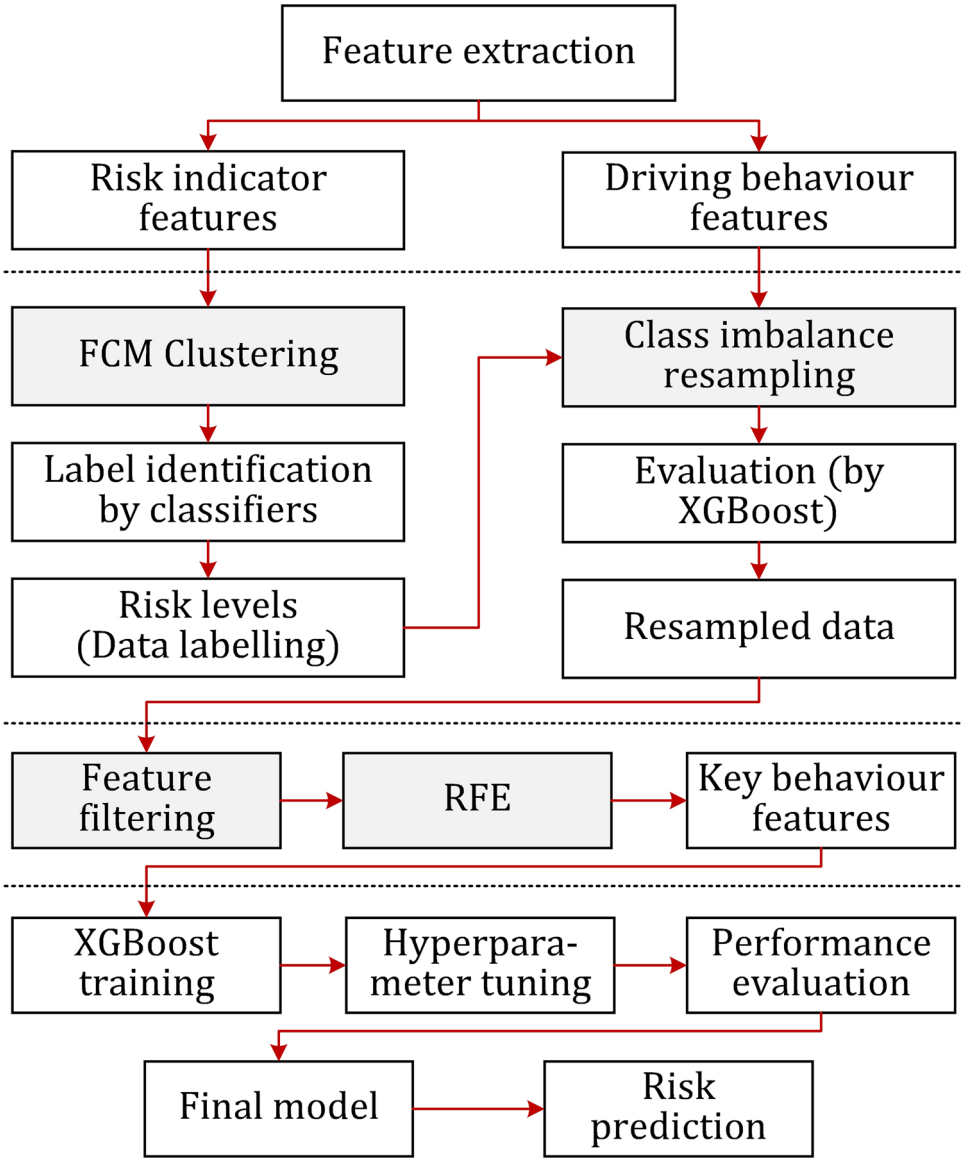

Fig. 1. Machine learning framework.

The framework is illustrated in Fig. 1 and Algorithm 1.

Algorithm 1.

1. Feature extraction:
   a. Driving behaviour features of instance i, $x_i^{(b)} = \{f_n(i)\}$;
   b. Risk indicator features, $x_i^{(r)}$, for risk labelling.
2. Risk labelling by clustering:
   a. Select clustering algorithms (e.g., FCM);
   b. Define potential numbers of clusters, K;
   c. Run clustering on $X^r = \{x_i^{(r)}\}$, with various K independently:
      1. Split X^r into K groups;
      2. Produce partition matrix, $i \in C(k), k \in [1, K]$;
      3. Assign the label, $y_i \leftarrow$ label $C(k)$;
   d. Find the best-suited K based on evaluation by a classifier (i.e., XGBoost);
   e. Risk levels, $Y = \{y_i\}$.
3. Imbalanced data resampling:
   a. Shortlist under-sampling and/or over-sampling strategies;
   b. Apply each strategy on (X^b, Y), where $X^b = \{x_i^{(b)}\}$:
      1. Obtain resampled data $(X^b, Y)'$;
      2. Model (XGBoost) training on $(X^b, Y)'$;
      3. Evaluate model performance;
   c. Select a best-suited resampling strategy.
4. Learning-based feature selection:
   a. Configure hyper-parameters;
   b. Train model (XGBoost) based on $(X^b, Y)'$;
   c. Rank feature relative importance, and remove less important ones;
   d. Recursive feature elimination, and find the best feature subset;
   e. Data set with key features, $(X^b, Y)''$.
5. Risk prediction:
   a. Model training and hyper-parameter tuning based on $(X^b, Y)''$;
   b. Prediction performance evaluation.

The framework uses XGBoost as the key algorithm in the processes of clustering evaluation, resampling evaluation, feature selection, and prediction. XGBoost is short for eXtreme Gradient Boosting, proposed by Chen and Guestrin (2016). XGBoost is an optimised gradient tree boosting system, with some algorithmic innovations (*e.g.*, approximate greedy search, parallel learning) and hyper-parameters (see Section 5.2) to improve learning and control over-fitting. A detailed introduction is described in Chen and Guestrin (2016).

Model performance is evaluated by various metrics via stratified cross-validation. For imbalanced classification, using the recall and precision are recommended to evaluate the minority and majority classes respectively (Díez-Pastor et al., 2015), and the area under the precision-recall curve (AUPRC) is computed as an integrated metric. Other metrics considered include accuracy, F1 score, and AUC (area under the receiver operating characteristic curve), *etc*. The metrics are calculated for each class, and two types of mean values are produced. One is macro-averaged over all classes, which is unweighted mean. Another is the average weighted by the number of true instances for each class (Lever et al., 2016).

2.2. Learning-based feature selection

A two-step hybrid method is developed to rank and select key features by machine learning. This procedure firstly filters a set of relative important features based on XGBoost, and then permutes to find an optimal subset from the filtered features using Recursive Feature Elimination (RFE), as illustrated in Algorithm 2.

This method combines the advantages of both feature ranking procedures. The permutation importance (mean decrease in accuracy, MDA) is used in RFE, which can find the optimal feature combinations, but the search procedure is computationally intensive (Guyon et al., 2002). Tree-based ensemble learning algorithms (*e.g.*, random forest, XGBoost) generate rankings of individual features based on Gini importance (mean decrease in impurity, MDI), which can be integrated to reduce the RFE search space quickly.

In XGBoost, the feature relative importance can be measured by several metrics, such as split weight, average gain, *etc*. Weight is the number of times that a feature is used to split the data across all boosted trees. More important features are used more frequently in building the boosted trees, and the rests are used to improve on the residuals. Instead of counting splits, gain measures the actual decrease in node impurity, which is the average gain across all splits in which the feature is used. The feature rankings of weight-based and gain-based importance can be obtained after XGBoost fitting.

The optimal feature subset can be selected based on the trade-off between learning performance and model simplicity (*i.e.*, fewer features). For MDA, a considerable decrease in accuracy indicates that the feature is highly relevant and useful, contributing to learning improvement, and vice-versa. Whereas an irrelevant feature only carries minimal impact, and a redundant feature also has limited contribution due to the high correlation with other more important ones. Therefore, redundant and irrelevant features could be removed, without loss of accuracy. Benefits of feature selection include better interpretability, simplified modelling, shorter learning time, and enhanced generalisation, among others (Guyon and Elisseeff, 2003; García et al., 2016).

Algorithm 2.

1. Train/tune model (XGBoost) using all features $\{f_n\}$;
2. Feature selection based on importance ranking:
   a. Calculate relative importance scores, $\{r_n\}$;
   b. Feature filtering by thresholds:
      1. Define thresholds $\{\tau_i\}$;
      2. For each τ_i, do:
         a. Remove features f_n for all $r_n < \tau_i$;
         b. Obtain subset S_i with remaining features;
         c. Re-training with S_i;
         d. Obtain performance A_i;
   c. Select the subset S' with the max $\{A_i\}$.
3. RFE:
   a. N is the feature size of S';
   b. For $n = N, \ldots, 2$, do:
      1. Permute n time, for $k = 1, \ldots, n$:
         a. Remove feature $f_k^{(n)}$, obtain subset $S'^{(n-1)}_k$;
         b. Re-training with $S'^{(n-1)}_k$;
         c. Obtain performance A_k;
      2. For the max $\{A_k\}$, eliminate the feature $f_k^{(n)}$;
      3. Keep $n-1$ important features, obtain subset $S'^{(n-1)}$ and performance $A^{(n)}$.
   c. Select the feature subset S'' with the max $\{A^{(n)}\}$.

2.3. Unsupervised risk rating

Clustering-based risk grading is proposed to estimate risk levels of vehicles in driving. This method entails using algorithms to cluster vehicles with similar risk patterns (measured by risk indicator features) into the same groups, and then decodes the risk level of each group. Given that there are no ground-truth labels about risk levels, this method can discover data-driven insights about risk exposures, and act as an unsupervised process of data labelling. Various clustering algorithms will be tested, such as K-means + +, Fuzzy C-means (FCM), and Self-Organising Map neural network (SOM) (*e.g.*, Kohonen, 2013; Qin et al., 2017), and a best-suited algorithm or hybrid will be selected based on clustering performance.

For reliable clustering performance, this study proposed an evaluation method, label identification by classifier. Better clustering can produce more reasonable labels, which could be more correctly identified by independent classifiers (*e.g.*, XGBoost). This method also helps to determine the levels of risk rating. Generally, a smaller number of clusters (K) would have a lesser opportunity to make misclassification (fewer borderline and overlapping areas), but the resolution (the ability to figure out more details and the highest risk instances) is also lower. Therefore, the risk rating and levels are determined based on the performance in conjunction with resolution.

2.4. Imbalanced data resampling

After data labelling, adaptive data resampling is integrated to generate datasets with more balanced class distribution. The techniques adopted for resampling include Edited Nearest Neighbours (ENN) undersampling and the SMOTE (synthetic minority oversampling technique). Since useful data might be eliminated, the undersampling is only performed on the safe class. Minority oversampling creates artificial instances by interpolation, which needs careful performance evaluation. Herein, the effects on learning performance of various resampling strategies are compared in order to find the best solution.

3. Feature extraction

3.1. Feature extraction from trajectory

Features are derived values from raw data, and used as input to a machine learning algorithm. High-quality features (*e.g.*, being informative, relevant, interpretable, non-redundant) are the basis for modelling and problem-solving, as well as generating reliable and convincing results.

Two kinds of features are developed, namely, driving behaviour features and risk indicator features. Driving behaviour features are extracted to produce in-depth and multi-view measures on behaviours (*e.g.*, movement characteristics). Risk indicator features are used in labelling risk levels, which are expected to distinguish between risk and safety. In a supervised learning framework, driving behaviour features act as the input, and risk indicator features serve for the target.

Vehicle movement trajectory data are used for feature extraction. The trajectory is an overall reflection of the driver behaviour, including risk perception and response, driving performance and style (Chai and Wong, 2015b). Plenty of spatial-temporal information can be mined from vehicle trajectory.

The procedure of trajectory-based feature extraction is elaborated as follows.

Step 1. A series of variables involving driving behaviour are derived from raw trajectory data, including velocity, acceleration, lateral position, the preceding and following vehicles, *etc*. Variables related to vehicle pairs, vehicle platoon and traffic conditions are also computed, such as, front gap, average velocity of vehicle stream within a lane segment for a given time window.

Step 2. Some functions are designed to mine in-depth information and build multi-variable relationships. Surrogate measures of traffic conflicts are used to build risk indicator features, such as TTC. Driving behaviours are explored from various measures and scales, such as, relative change, performance comparison, similarity match, correlation coefficient, small time windows.

Step 3. Some operations are defined to summarise the key information and profile the data series, including statistical descriptions, threshold-based filtering, aggregated or accumulated values, *etc*.

Finally, some informative and interpretable features are preliminarily shortlisted, and learning-based feature selection is developed to select the most important ones.

3.2. Driving behaviour features

Based on vehicle movement trajectory, a total of 1328 behaviour features are extracted to describe various characteristics about driving. The variables, functions, operations developed for feature extraction are shown in Table 1. The extracted driving behaviour features are listed in Table 2.

Feature fusion and in-depth mining can enhance reliability and predictability. To capture safety-critical conditions, various behaviours are assessed, such as braking response, speed matching, gap maintenance. For specific behaviours, multiple measures and functions are designed. For example, the instantaneous changes of movement are measured by several ratio-based functions, including percentage ratio, log ratio and bias ratio, since different forms of ratio have different sensitivities to changes. The intensity of abrupt braking can be reflected from high magnitude jerk, the high percentage change of velocity, when decelerating, *etc*. The lane keeping features (*e.g.*, y.std) are measured based on vehicle lateral trajectory data that do not involve lane-changing. From predictable perspectives, compared with extreme values, percentile values provide a kind of early signals, which are more helpful in early diagnostics of risk conditions. Besides, microscale behaviours can be measured based on features defined by small moving time windows. The proposed feature extraction procedure is scalable for such kind of trajectory time series data.

3.3. Risk indicator features

To conduct unsupervised data labelling, risk indicator features are also extracted, which are used as the input of the clustering-based risk rating. Previous study has evaluated the feasibility of using risk indicators to assess pre-accident risk conditions, and the usefulness of the hybrid indicators of Time Integrated TTC (TIT) and Crash Potential Index (CPI) is demonstrated (Shi et al., 2018).

Herein, five risk indicator features are constructed based on TIT and CPI. TIT is calculated based on the threshold of TTC, which takes into account the accumulated impact of risk exposures in both severity and duration (Minderhoud and Bovy, 2001). Typical values of TTC thresholds range from 1.5 s to 4.0 s (Shi et al., 2018). Thus, three TIT-based features are built, namely, TIT.t1 (TTC threshold = 2 s), TIT.t2 (TTC threshold = 3 s) and TIT.t3 (TTC threshold = 4 s). CPI measures the probability that a vehicle's Deceleration Rate to Avoid Crash (DRAC) exceeds its Maximum Available Deceleration Rate (MADR) or braking capacity during a given period (Cunto and Saccomanno, 2008). Two CPI-based features are developed, namely, CPI.m1 and CPI.m2, based on two MADR measures adopted in CPI.

Detailed information about risk surrogate indicators is described in Shi et al. (2018). The formula of each indicator was illustrated in the respective references (*e.g.*, Minderhoud and Bovy, 2001; Cunto and Saccomanno, 2008).

4. Analysis and results

4.1. Data description and preprocessing

This study utilises the vehicle trajectory data provided in FHWA Next Generation Simulation (NGSIM) program. The vehicle trajectory data were collected from a 640-metre road segment of US Route 101 (Hollywood Freeway), for about 45 min, on June 15th, 2005. Main variables include vehicle trajectory, length, instantaneous velocity and acceleration, *etc*. The data acquisition resolution is 0.1 s in relative units. The features are calculated using the full data set available. After data cleaning, a total of 5084 instances (vehicles) is used for feature modelling, involving 3,203,867 records.

Savitzky-Golay filter is used in data preprocessing to smooth out potential noise (*e.g.*, unphysical fluctuation) in data acquisition. Savitzky-Golay filter approximates a given signal using a sliding

Table 1
Variables, functions and operations for feature extraction.

	Code	Description	No.
VARIABLES			**1–14**
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	1; 2
Trajectory of the preceding vehicle	pv.x; pv.y	$x_{i-1}(t)$; $y_{i-1}(t)$, for preceding vehicles (pv) $i-1$	3; 4
Lane	lane	Lane number of a vehicle travelling on	5
Vehicle type	class	0-motocycle; 1-car; 2-truck	6
Vehicle length	L	L_{i-1}, vehicle length of preceding vehicle	7
Velocity	vel	$v_i(t) = d[x_i(t)]/dt$	8
Acceleration	acc	$a_i(t) = d[v_i(t)]/dt$, negative value indicates deceleration	9
Jerk	jerk	$j_i(t) = d[a_i(t)]/dt$	10
Front gap	gap	Calculated by $x_{i-1}(t) - x_i(t) - L_{i-1}$	11
Variables of preceding vehicles	pv.vel; pv.acc; pv.jerk	$v_{i-1}(t)$; $a_{i-1}(t)$; $j_{i-1}(t)$	12–14
FUNCTIONS			**15–30**
Moving time windows	w1; w2; w3	Moving windows defined by time intervals of 1.0 s (w1), 5.0 s (w2), and 10 s (w3), return time series data	15–17
Difference	dif	$s_{i-1}(t) - s_i(t)$, measure difference of variable s between subject vehicle and preceding vehicles	18
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)	19
Log ratio	logr	Measure relative change on a logarithmic scale, per 1.0 second, calculated by $log\frac{s_i(t)}{s_i(t-1)}$	20
Vehicle to flow ratio	vfr	Comparison between a vehicle and the average performance of vehicle platoon in the same lane segment	21
Range	rng	Calculated by $max(s_i(t)) - min(s_i(t))$	22
Coefficient of range	crng	Calculated by $\frac{max(s_i(t)) - min(s_i(t))}{max(s_i(t)) + min(s_i(t))}$	23
Simple moving average	sma	Mean value of the time series data within a moving window	24
Moving standard deviation	msd	Standard deviation of the time series data within a moving window	25
Relative standard deviation	rsd	Relative variability and unitised measure, defined as the ratio of std to mean	26
Bias 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 $(t-w,\ t)$	27
Dynamic time warping	dtw	Using DTW algorithm to measure similarity between two temporal sequences	28
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	29;30
OPERATIONS			**31–45**
Basic centre and dispersion	mean; std	Values of mean and standard deviation	31;32
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	33–36
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	37-41
Mean absolute deviation	mad	Measure variability or dispersion	42
Profile shape	krt; skw	Unbiased kurtosis (krt) over data using Fisher's definition; unbiased skew (skw), normalised by n-1	43;44
Absolute mean	absm	Mean of absolute value	45

Table 2
Driving behaviour features.

Features	Code and counts	Total
Basic characteristics of subject vehicle	{vel; acc; gap}.{kurt; mad; max; mean; min; p01; p05; p95; p99; q1; q2; q3; skew; std}(42); {jerk}.{max; mean; min; p01; p05; p95; p99; q1; q2; q3; std}(11); {clas}(1)	54
Lane keeping and changing	{y}.{std}(1); {lane}.{mean; std; rng}(3)	4
Relative comparison with respect to proceeding vehicles or vehicle platoon	{acc; vel}.{dif}.{max; mean; min; p01; p05; p95; p99; q1; q2; q3; std}(22); {acc; gap; vel; jerk}.{vfr}.{max; mean; min; p01; p05; p95; p99; q1; q2; q3; std}(44); {acc; vel}.{dif}.{vfr}.{max; mean; min; p01; p05; p95; p99; q1; q2; q3; std}(22)	88
Basic characteristics of preceding vehicles	pv.{vel; acc}.{kurt; mad; max; mean; min; p01; p05; p95; p99; q1; q2; q3; skew; std}(28); pv.jerk.{kurt; mad; max; mean; min; p01; p05; p95; p99; q1; q2; q3; skew; std}(11); pv.{clas}.{mean; truck; motorcycle}(3)	42
Relative change measured by percentage ratio and log ratio	{vel; acc; gap; vel.dif; acc.dif; pv.vel; pv.acc; y; pv.y}.{pct}.{absm; max; mean; min; p01; p05; p95; p99; q1; q2; q3; std}(108); {y; vel; gap; pv.y; pv.vel}.{logr}.{max; mean; min; p01; p05; p95; p99; q1; q2; q3; std}(55)	163
Microscale behaviour defined by moving windows	{vel; pv.vel; acc; pv.acc; gap; vel.dif; acc.dif}.{w1; w2; w3}.{rng; crng; sma; msd; rsd; emar}.{mean; std; max; min}(504); {acc; vel}.{dif}.{vfr}.{w1; w2; w3}.{max; mean; min; p01; p05; p95; p99; q1; q2; q3; std}(66); {acc; gap; vel; jerk}.{vfr}.{w1; w2; w3}.{max; mean; min; p01; p05; p95; p99; q1; q2; q3; std}(132)	702
Behaviour similarity match by dynamic time warping (DTW)	{vel; acc; x}.{dtw}(3); {vel; acc; x}.{w1; w2; w3}.{dtw}.{mean; std; p05; q1; q3; p95; max; min}(72)	75
Behaviour correlations	{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)	200
Σ		1328

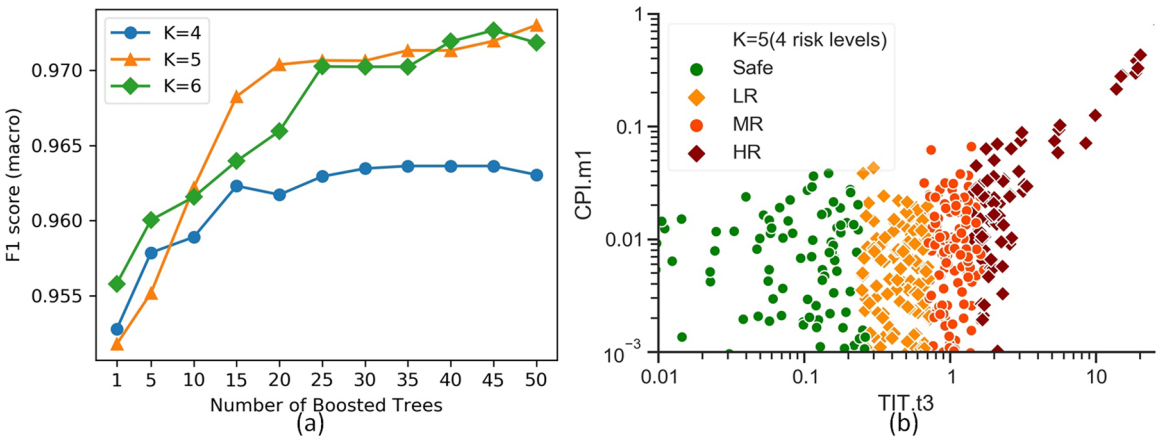

Fig. 2. Clustering result evaluation.

window and a low degree polynomial to model data within that window, and also incorporates the introduced error in the approximation process using linear least squares. This filter reduces the disadvantage of cutting off peaks. The samples of velocity and acceleration are fitted using the 1st and 2nd order polynomials respectively, based on the 1.0 s filter window.

4.2. Labelling of risk levels using FCM

Based on pattern similarity of risk indicator features, vehicles are clustered into different groups with graded risk ratings. The clustering captures the position (*i.e.*, the risk level) of an instance in the entire dataset, and a label about the belonged position is assigned to the instance. The clustering performance of the shortlisted algorithms is tested based on preliminary experiments. Compared with K-means++ and SOM, FCM is best-suited in this problem, which presents a better overall performance, especially providing stable and reasonable clustering results. Besides, Fuzzy theories are interpretable, and widely used in behaviour analysis for traffic safety (Chai and Wong, 2015b).

Given the imbalanced problems and lack of ground truth labels, several values of the number of clusters (K) are considered in the FCM clustering, ranging from 4 to 6. The clustering results and evaluation are presented in Fig. 2 and Table 3. Label identification by XGBoost provides an evaluation of the clustering results, using models built with various numbers of boosted trees to represent both weak and strong classifiers, as shown in Fig. 2(a). The clustering with 5 groups shows better performance. A detailed comparison is presented in Table 3, based on an XGBoost ensembled with 20 boosted trees (a moderate-level classifier). The high identification accuracy can imply that an underlying structure close to ground truth labels is promising to be discovered by both clustering and classifier.

Four risk levels are obtained from the clustering of 5 groups. The reason is that, for the highest risk level, the number of instances is limited (only 8 vehicles), hence they are combined with the sub-highest class. One annotation of the risk labelling is the safe level (group 1; with 3653 instances), low risk level (group 2; LR; with 900 instances), moderate risk level (group 3; MR; with 425 instances), high risk level (combination of groups 4 and 5; HR; with 106 instances). The safe level indicates a near-zero risk, which has the lowest likelihood to involve in traffic conflicts. A scatter plot of the clustered risk levels is illustrated in Fig. 2(b). TIT contributes to distinguishing the range of each risk level, and CPI further figures out the instances with high likelihood to involve an accident.

Table 3
Clustering results and label identification by XGBoost.

K	Clustered groups	Accuracy	F1-score#	AUC#	AUPRC#
4	(3921; 943; 212; 8)*	0.994	0.962	0.993	0.942
5	(3653; 900; 425; 98; 8)	0.993	0.970	0.999	0.974
6	(3445; 812; 525; 224; 70; 8)	0.992	0.966	0.977	0.936

* Number of instances in each clustered group, from safe to higher risk levels. # Macro-averaged over all classes.

4.3. Feature importance ranking and recursive elimination

The linkages between behaviour features and corresponding risk levels are built using XGBoost. To improve model fitting, the imbalanced dataset is processed by safe-class under-sampling using Repeated ENN, in which 1211 instances are selected from initial 3653 safe-class instances. Thus, the total data size drops from 5084 to 2642. The detailed data resampling procedure is discussed in Section 5.1.

Hyper-parameters are configured to build an appropriate XGBoost model for feature ranking. After model training, the split weight and average gain for each feature are generated, which are normalised to calculate the weight-based and gain-based relative importance scores, respectively. The scores measure the usefulness of a feature in building the boosted trees in XGBoost. Higher scores indicate greater relative importance. A range of score thresholds is defined for a quick feature filtering. In each iteration, the features with the importance scores greater than the threshold value are kept, and the learning performance is estimated via 10-fold stratified cross-validation. The feature filtering by weight-based and gain-based importance ranking are demonstrated in Fig. 3(a) and 3(b), respectively.

From Fig. 3, a total of 148 promising features are filtered, including 102 features selected according to weight-based importance ranking, and 67 features with higher Gini importance, while noting that some features are duplicated in the two filtered feature subsets. Weight-based selection generally favours the features with more classes, and gain-based selection is biased towards the ones with stronger signals (*e.g.*, impurity).

In the RFE process, an optimal feature combination is selected from the filtered features, by model re-training and recursively pruning the feature with the least permutation importance from the current set. The learning performance of each iteration is shown in Fig. 4. The combination with the best mean accuracy has 122 features. However, the performance at the subset with 64 features reaches an accuracy of 88.91%, but lower variance. Considering the trade-off of complexity and performance, the combination with 64 features is suggested, which has less complexity and a modest decrease in estimated accuracy from 89.03% down to 88.91%. The selected key features are listed in Table 4.

From a practical perspective, the identification of key features guides the procedures of data collection, mining and understanding. From Table 4, the gap, velocity and acceleration are the most

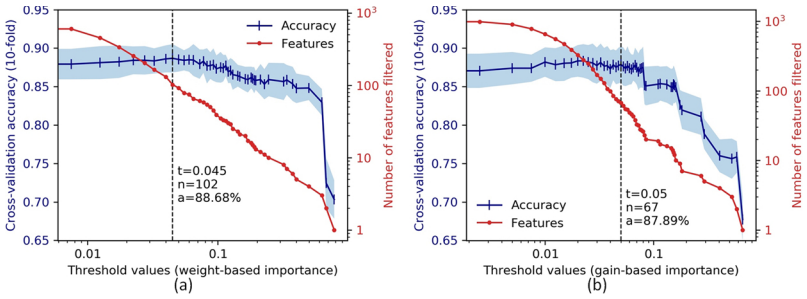

Fig. 3. Feature filtering by relative importance ranking.

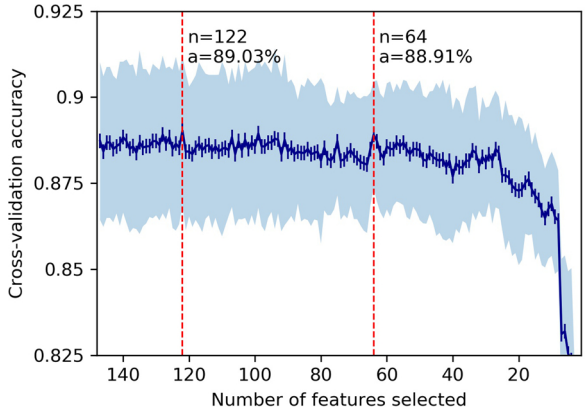

Fig. 4. Feature combination selection by RFE.

informative variables involving risk assessment. Data capture and processing at shorter time intervals are also important, given that 38 key features are defined based on small moving windows. For data mining, 17 ratio-based features are selected (*i.e.*, 7 log ratio, 6 percentage change, and 4 bias ratio), which can measure the abnormal changes of movement, and capture potential risk signals. More than half of the features involve behaviour comparison and relative performance, which indicates that dynamic benchmarking of the overall performance of vehicle stream is helpful to assess the behaviour of individual vehicles pertinently. Moreover, the fusion of multi-view and in-depth features makes the system more robust and fault-tolerant, and also has high transparency.

4.4. Behaviour-based crash risk prediction

The risk levels of vehicles in driving are predicted based on the key behaviour features selected. For model building and unbiased evaluation, nested cross-validation is applied, which uses inner loops for hyper-parameter configuration and model training, and outer loops for out-of-sample validation. A hyper-parameter tuning procedure is demonstrated in Section 5.2.

With the tuned hyper-parameters, the final prediction model is obtained by re-training using the complete data. The predictive power (generalisation ability) of the final model can be estimated using the cross-validation performance. Herein, a satisfactory prediction with an overall accuracy of 89% is achieved based on XGBoost and key behaviour features. Detailed estimation of the prediction performance is shown in Table 5.

Predicting detailed risk levels is more challenging but valuable. Great performance by isolated validation indicates an accurate and reliable predictive power, whilst allowing the model to work well on unseen data. The prediction of risk levels can be used as early signals for crash potentials and likelihood, which allows measures to be taken to reduce crash proneness, as well as make a crash prediction ahead of time.

Table 5
Prediction performance evaluation.

Group	Precision	Recall	AUC	AUPRC
Safe	0.952	0.946	0.990	0.990
LR	0.855	0.890	0.966	0.931
MR	0.798	0.781	0.966	0.835
HR	0.824	0.660	0.989	0.844
Macro	0.857	0.819	0.978	0.900
Weighted	0.889	0.889	0.983	0.939

5. Discussion

5.1. Data resampling procedure

To reduce the impact of class imbalance, six resampling strategies are investigated, as shown in Table 6. Under-samplings by Repeated ENN (RENN) and All K ENN (AKNN) reduce the safe-class instances to an amount close to the LR group. Four strategies of combining both under-sampling and over-sampling (*i.e.*, from R3 to R6) are also examined. Over-samplings create instances in the HR group, reaching an

Table 4
Key features selected.

Variable	Features	Counts
Gap	gap.vfr.w3.min; gap.pct.min; gap.pct.q1; gap.p01; gap.logr.p05; gap.vfr.w3.p05; gap.pct.p01; gap.logr.std; gap.min; gap.w1.msd.max; gap.vfr.w3.p01; gap.pct.p05; gap.w1.crng.std; gap.logr.absm; gap.vfr.min; gap.logr.p01; gap.vfr.w2.p95; gap.w1.emar.min; gap.vfr.w2.min; gap.w1.sma.min; gap.w1.crng.max; gap.w2.emar.min	22
Acceleration	acc.dif.w2.sma.mean; acc.dif.vfr.w3.p01; acc.w3.crng.std; acc.dif.w2.emar.std; acc.dif.vfr.w2.q2; acc.dif.w1.sma.max; acc.dif.vfr.w2.p01; acc.vfr.w3.mean; acc.w1.emar.max; acc.dif.vfr.w1.max; acc.dif.w2.rsd.min; acc.dif.mean; acc.w2.sma.mean; acc.dif.vfr.w2.q3	14
Velocity	vel.dif.p99; vel.dif.w3.sma.min; vel.dif.p95; vel.dif.max; vel.dif.w1.sma.mean; vel.w2.scor.std; vel.logr.p05; vel.w1.msd.std; vel.dif.w1.sma.max; vel.dif.p01; vel.dif.w1.rng.mean; vel.dif.mean; vel.dif.w1.msd.max	13
Preceding vehicles	pv.vel.logr.p05; tpv.pcor; pv.vel.logr.p99; tpv.w1.pcor.q3; pv.acc.p05; tpv.w2.pcor.max; pv.acc.w2.rsd.max; pv.vel.pct.std; tpv.w1.pcor.p95; tpv.w2.scor.p95; tpv.w2.pcor.p95	11
Jerk	jerk.vfr.w3.std; jerk.vfr.min	2
Lateral position	y.std; y.pct.p99	2
Total		64

Table 6
Resampling strategies and resampled datasets.

No.	Under-sampling	Over-sampling	Counts of resampled data
R1	ERNN	–	2642 (1211; 900; 425; 106)
R2	AKNN	–	2587 (1156; 900; 425; 106)
R3	AKNN	SMOTE + ENN	2783 (1033; 900; 425; 425)
R4	AKNN	SVM-SMOTE	2906 (1156; 900; 425; 425)
R5	RENN	SMOTE + ENN	2817 (1067; 900; 425; 425)
R6	RENN	SVM-SMOTE	2961 (1211; 900; 425; 425)

amount close to the MR group. Two SMOTE-based hybrid oversampling techniques are applied, which are SMOTE with ENN cleaning (SMOTE + ENN) and SMOTE with classification by support vector machine (SVM-SMOTE). The effects of resampling strategies on learning performance are compared in Fig. 5. Detailed information about these resampling techniques can be found in Tomek (1976); Batista et al., (2004); Nguyen et al., (2009), among others.

Under-sampling can improve the learning performance of the LR class, and RENN is slightly better than AKNN in this experiment. There is no evidence to show that oversampling the HR class could improve the learning performance of classes of MR and/or LR. In Fig. 5(a), the initial dataset and four datasets processed by over-sampling have higher accuracy values, the reasons being that more instances are in the safe class and oversampled HR class, which are also more homogeneous and less overlapping, hence easier to classify. Besides, the obvious improvements of the HR class are based on interpolated data, to which careful attention should be paid. Herein, safe-class under-sampling by RENN is selected to reduce the class imbalance, for better modelling.

5.2. Hyper-parameter tuning

A range of hyper-parameters is tuned using Grid Search for model optimisation, as shown in Table 7 and Fig. 6. XGBoost is a kind of ensemble learning configured with boosted trees. The learning rate helps to shrink the boosting process by weighting, which makes fitting more conservative. For individual boosted trees, tree hyper-parameters can directly control model complexity, such as maximum tree depth, splitting weight, *etc*. Besides, random subsampling of instances and features also help to decorrelate and improve the model robustness against noise, hence reducing the variance.

Fig. 6 shows the model performance of each Grid Search, measured by the cross-entropy loss via 10-fold stratified cross-validation, with the standard deviation values shown as error bars. The cross-entropy loss (log loss) is used for a more nuanced evaluation, which is defined by the negative log-likelihood of the true labels given a probabilistic prediction. Generally, hyper-parameters values with the best mean and relatively low standard deviation are selected; if several values produce similar performance, the tuned value is selected when a plateau in performance or a point of diminishing returns is observed. The tuned values and corresponding accuracy are listed in Table 7.

5.3. Potential applications

This study can contribute to a range of applications. Reliable driving assessment can be used for behaviour-based insurance (also called “pay how you drive”, PHYD). Through PHYD, insurance-based incentives can be applied to encourage carefully driving, such as insurance rewards and premium discount (Eftekhari and Ghatee, 2018).

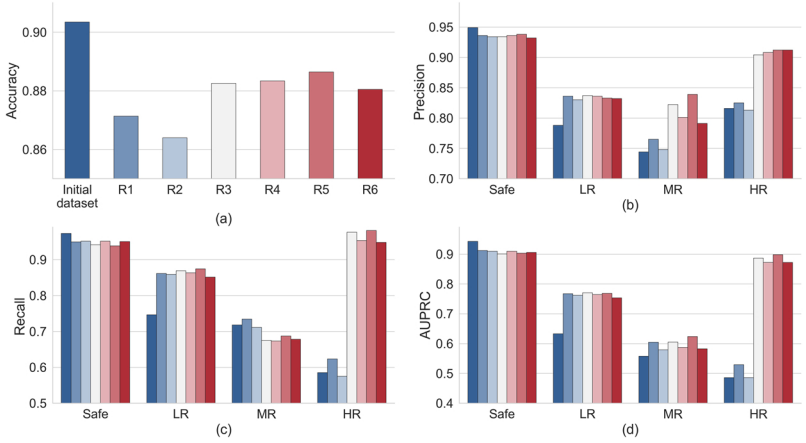

Fig. 5. Performance comparison of resampling strategies.

Table 7
Key hyper-parameters and tuned values.

Hyper-parameters	Description	Tuned value	Accuracy (%)
1. Ensemble hyper-parameters			
Learning rate	Shrink the feature weights of each boosting step	0.1	88.61
Number of estimators	Number of boosted trees added in model	150	88.61
2. Boosted tree hyper-parameters			
Tree depth	Maximum depth of a tree	5	88.72
Splitting weight	Further partitioning of a leaf node in tree building process	1	88.72
3. Subsampling hyper-parameters			
Instance subsample ratio	Random sample of the training data prior to growing trees in every boosting iteration	0.7	88.99
Feature subsample ratio	Random sample of features for each split in tree level	0.8	88.99
4. Regularisation hyper-parameters			
Gamma	Minimum split loss reduction required to make a further partition on a leaf node	0	88.99
Alpha	L1 regularisation term on weights	0	88.99
Lambda	L2 regularisation term on weights	1	88.99
5. Tuning update			89.01

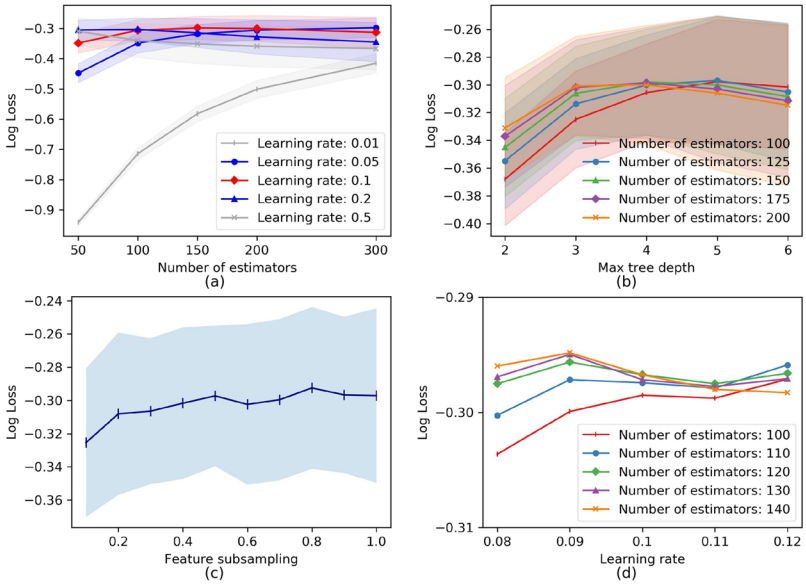

Fig. 6. Hyper-parameters tuning and corresponding performance comparison.

Recognition of unsafe behaviour provides a basis for proactive safety management, such as, the identification and remediation of crash-prone locations and layouts, driving enhancement systems for public transport and commercial vehicles. Combinations of features can be used to assess driving performance (short-term) and driving style (long-term), while detailed risk levels and behaviours are helpful to improve the evaluation and design of targeted countermeasures. Moreover, the trajectory data are also flexible to collect, and interpretable features are promising to be extracted in a non-interference manner.

5.4. Limitations

For clustering-based risk grading, the major challenge is the lack of crash instances, which makes it hard to verify the linkages between the highest risk level and actual crash occurrence. A potential way of validation is to examine the accident records of the drivers/vehicles which are clustered as higher risk levels.

The feature extraction is far from being exhaustive. In-depth feature extraction is recommended to further improve modelling, which should cover a broader range of driving behaviours and risk conditions, such as lane-changing, conflicts between motorcycles and vehicles. The interests of feature extraction are mainly twofold, namely, making risk assessment more reliable, and providing early signals for risk-based crash prediction.

6. Conclusions

This study designs an integrated approach to assess driving behaviours and predict risk levels, which combines supervised feature selection and unsupervised data labelling. This study contributes to the safety domain and associated literature in four areas.

(1) Extraction of massive features from vehicle trajectory. To mine information about driving behaviours, for individual vehicles, more than a thousand features are extracted comprehensively, which produce in-depth and multi-view measures on behaviours. The feature extraction procedure is scalable for trajectory time series data.
(2) Clustering-based risk rating and data labelling. To estimate risk potentials of vehicles in driving, risk indicator features are constructed, which are used to group vehicles by clustering, and achieve unsupervised data labelling. Four risk levels are obtained to assess unsafe driving behaviours.
(3) Key feature selection by importance ranking and recursive elimination. The linkages between behaviour features and corresponding risk levels are built using XGBoost, and 64 key behaviour features are identified. Besides, under-sampling of the safe-class data is conducted to amend the class imbalance.
(4) Satisfactory results of behaviour-based risk prediction by XGBoost. The risk levels of vehicles in driving are predicted based on the key features, and predictive power with an overall accuracy of 89% is achieved. The approach is effective and reliable to identify important features for risk assessment, and contributes to guiding the direction of feature engineering, which is the key to improve modelling.

Acknowledgements

The research reported in this paper is part of the PhD research programme of the first author.

References

Bagdadi, O., 2013. Assessing safety critical braking events in naturalistic driving studies. Transp. Res. Part F: Traff. Psychol. Behav. 16, 117–126.
Batista, G.E., Prati, R.C., Monard, M.C., 2004. A study of the behavior of several methods for balancing machine learning training data. ACM SIGKDD Explorat. Newslett. 6 (1), 20–29.
Beyan, C., Fisher, R., 2015. Classifying imbalanced data sets using similarity based hierarchical decomposition. Pattern Recognit. 48 (5), 1653–1672.
Chai, C., Wong, Y.D., 2015a. Comparison of two simulation approaches to safety assessment: cellular automata and SSAM. J. Transp. Eng. 141 (6), 05015002.
Chai, C., Wong, Y.D., 2015b. Fuzzy cellular automata model for signalized intersections. Comp. Aid. Civ. Infrastr. Eng. 30 (12), 951–964.
Chen, T., Guestrin, C., 2016. XGBoost: a scalable tree boosting system. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 785–794.
Cunto, F., Saccomanno, F.F., 2008. Calibration and validation of simulated vehicle safety performance at signalized intersections. Accid. Anal. Prev. 40 (3), 1171–1179.
Díez-Pastor, J.F., Rodríguez, J.J., García-Osorio, C.I., Kuncheva, L.I., 2015. Diversity techniques improve the performance of the best imbalance learning ensembles. Inform. Sci. 325, 98–117.
Eftekhari, H.R., Ghatee, M., 2018. Hybrid of discrete wavelet transform and adaptive neuro fuzzy inference system for overall driving behavior recognition. Transp. Res. Part F: Traff. Psychol. Behav. 58, 782–796.
García, S., Ramírez-Gallego, S., Luengo, J., Benítez, J.M., Herrera, F., 2016. Big data preprocessing: methods and prospects. Big Data Analytics 1 (1), 9.
Guyon, I., Elisseeff, A., 2003. An introduction to variable and feature selection. J. Mach. Learn. Res. 3, 1157–1182.
Guyon, I., Weston, J., Barnhill, S., Vapnik, V., 2002. Gene selection for cancer

classification using support vector machines. Mach. Learn. 46 (1-3), 389–422.
Hong, J.H., Margines, B., Dey, A.K., 2014. A smartphone-based sensing platform to model aggressive driving behaviours. In Proceedings of the 32nd Annual ACM Conference on Human Factors in Computing Systems. pp. 4047–4056.
Kohonen, T., 2013. Essentials of the self-organizing map. Neural Networks 37, 52–65.
Lever, J., Krzywinski, M., Altman, N., 2016. Classification evaluation. Nat. Methods 13 (8), 541–542.
López, V., Fernández, A., García, S., Palade, V., Herrera, F., 2013. An insight into classification with imbalanced data: empirical results and current trends on using data intrinsic characteristics. Inform. Sci. 250, 113–141.
Mannering, F.L., Shankar, V., Bhat, C.R., 2016. Unobserved heterogeneity and the statistical analysis of highway accident data. Analytic Methods in Accid. Res. 11, 1–16.
Minderhoud, M.M., Bovy, P.H., 2001. Extended time-to-collision measures for road traffic safety assessment. Accid. Anal. Prev. 33 (1), 89–97.
Nguyen, H.M., Cooper, E.W., Kamei, K., 2009. Borderline over-sampling for imbalanced data classification. Proceedings of Fifth International Workshop on Computational Intelligence and Applications 1, 24–29.
Perez, M.A., Sudweeks, J.D., Sears, E., Antin, J., Lee, S., Hankey, J.M., Dingus, T.A., 2017. Performance of basic kinematic thresholds in the identification of crash and near-crash events within naturalistic driving data. Accid. Anal. Prev. 103, 10–19.
Qin, J., Fu, W., Gao, H., Zheng, W.X., 2017. Distributed k-means algorithm and fuzzy c-means algorithm for sensor networks based on multiagent consensus theory. IEEE Trans. Cybern. 47 (3), 772–783.
Shi, X., Wong, Y.D., Li, M.Z.F., Chai, C., 2018. Key risk indicators for accident assessment conditioned on pre-crash vehicle trajectory. Accid. Anal. Prev. 117, 346–356.
Tomek, I., 1976. An experiment with the edited nearest-neighbor rule. IEEE Trans. Syst., Man, and Cybern. 6, 448–452.
Wahlström, J., Skog, I., Händel, P., 2017. Smartphone-based vehicle telematics: a ten-year anniversary. IEEE Trans. Intell. Transp. Syst. 18 (10), 2802–2825.
Wu, K.F., Jovanis, P.P., 2013. Defining and screening crash surrogate events using naturalistic driving data. Accid. Anal. Prev. 61, 10–22.
Zheng, L., Ismail, K., Meng, X., 2014. Traffic conflict techniques for road safety analysis: open questions and some insights. Can. J. Civ. Eng. 41 (7), 633–641.