Accident Analysis and Prevention 158 (2021) 106199

Contents lists available at ScienceDirect

Accident Analysis and Prevention

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

Cost-sensitive learning for semi-supervised hit-and-run analysis

Siying Zhu, Jianwu Wan *

School of Civil and Environmental Engineering, Nanyang Technological University, Singapore

ARTICLE INFO

Keywords:
Hit-and-run
Cost-sensitive
Semi-supervised learning
Imbalanced dataset
Unlabelled data

ABSTRACT

Hit-and-run crashes not only degrade the morality, but also result in delays of medical services provided to victims. However, class imbalance problem exists as the number of hit-and-run crashes is much smaller than that of non-hit-and-run crashes. The missing label problem also exists in the crash analysis due to reasons like data barrier such that the information hidden in the unlabelled samples has not been effectively utilised. In this paper, a cost-sensitive semi-supervised logistic regression (CS^{3}LR) model is proposed for hit-and-run analysis, in order to tackle class-imbalanced data distribution and missing label problem, based on the crash dataset of Victorian, Australia (2013–2019). By performing label estimation with logistic regression jointly utilising both labelled and unlabelled data with pseudo labels in a well-designed cost-sensitive semi-supervised maximum likelihood framework, the proposed model can obtain an unbiased likelihood parameter for hit-and-run prediction and analysis. Comparing the experimental results of CS^{3}LR model with two logistic regression models and seven machine learning methods, better performance of CS^{3}LR model is demonstrated. The most significant contributing factors to hit-and-run crashes extracted by CS^{3}LR with only 10% labelled data show a high degree of consistency with the true contributing factors obtained by the supervised cost-sensitive logistic regression with complete hit-and-run labels. The effects of class-weighted ratio and hyper-parameter λ on the performance of hit-and-run crash prediction model have also been analysed. The results can further provide recommendations and implications on the policies and counter-measures for preventing hit-and-run collisions and crimes. The methodology proposed in this paper can also be employed to analyse crash data with other types of missing labels, such as crash severity.

1. Introduction

In many countries, road traffic collisions are the leading cause of casualties and bring heavy costs to society, and the hit-and-run behaviour of drivers is a significant determinant of morality (Kim et al., 2008; Tay et al., 2008; Jiang et al., 2016). Hit-and-run crashes are defined as the collisions where at least one driver of the colliding vehicles fled the scene before providing information to report the crash or helping the victims (Roshandeh et al., 2016). Delays in emergency medical services can be caused due to the hit-and-run behaviour of drivers such that the victims are subject to increasing risk (Tay et al., 2008). Hit-and-run analysis has been conducted since the last decade to investigate the contributing factors to hit-and-run decision of drivers.

In the domain of hit-and-run analysis, the imbalance issue of the hit-and-run crash dataset has attracted limited attention, as the hit-and-run sample size is much smaller than the non-hit-and-run sample size. If the data imbalance problem is not properly treated, the model will tend to classify the majority non-hit-and-run samples more accurately than the minority hit-and-run samples. The minority class barely contribute to accuracy compared to the majority class due to class imbalance, which is uninformative (Zhou et al., 2019; Mohammadi et al., 2019).

Cost-sensitive learning algorithm is a solution to the imbalanced classification problem in this paper at the algorithmic level, which assigns different cost for misclassifying the hit-and-run and non-hit-and-run crashes. In the past decades, many cost-sensitive machine learning algorithms, e.g., cost-sensitive support vector machine (Masnadi-Shirazi et al., 2019), k-nearest neighbour(Zhang and Zhou, 2010), decision tree (Lomax and Vadera, 2013), etc., have been proposed for the applications such as face recognition (Wan and Wang, 2019) and software defect prediction (Li et al., 2018). However, as far as we know, only a few of them can be applied for traffic factor analysis.

Another critical issue in crash analysis is the missing label problem. To the authors' best knowledge, the existing methods and algorithms were applied under the assumption that all the crash data are supervised

* Corresponding author at: 50 Nanyang Avenue, N1-B1a-2a, Singapore 639798, Singapore.
E-mail addresses: siying001@e.ntu.edu.sg (S. Zhu), jianwuwan@gmail.com (J. Wan).

https://doi.org/10.1016/j.aap.2021.106199
Received 17 October 2020; Received in revised form 23 February 2021; Accepted 15 May 2021
Available online 25 May 2021

with the labels. However, due to the ever-changing research requirement from researchers, the historical data which was usually independently recorded by the traffic manager may not contain the required label information. Sometimes, even if the label information has been collected, the data barriers may prevent researchers from exploiting the label information due to reasons like privacy concerns (Alharthi et al., 2017). In such a scenario that the crash dataset does not have any label information historically while the analysis is still needed in those regions, the possible solution for existing methods is to re-collect the new crash data along with the labels. However, the re-collection process will probably lead to long data collection time for collecting enough amount of data and inaccuracies due to human error.

Fortunately, in hit-and-run analysis, we have the fact that no matter the hit-and-run labels are previously recorded or not, there has to exist a label of hit-and-run decision of driver corresponding to each historical traffic collision. Thus, we propose to relabel only a few historical crashes when considering the expensive costs of relabelling spent on questionnaire or return visit. By doing so, the missing label problem in hit-and-run analysis is reduced to a semi-supervised learning probelm where the crash dataset now contains a few manually labelled historical data and large amount of unlabelled data. As the labelled and unlabelled historical crash data are usually independent identically distributed (Jebara et al., 2009), the semi-supervised learning method in machine learning which constructs models by utilising both labelled data and unlabelled data can improve the accuracy of analysis (Tanha et al., 2017; Zhu, 2005; Tan et al., 2011).

This paper aims to develop a cost-sensitive semi-supervised learning method to tackle the class imbalance and missing label issues in hit-and-run analysis. The logistic regression model (Hastie et al., 2009) most commonly applied in the literature for hit-and-run analysis is applied as the base learner. For the purposes of experiment and performance evaluation, we adopt the Victorian, Australia (2013–2019) (VicRoads, 2019) as the historical crash data and assume only a few crashes have been manually relabelled with the true hit-and-run label information. The remaining large amount of data are considered to be unlabelled, i.e., with unknown hit-and-run labels. It is worth noting that the base learner can be extended to other types of statistical model. Moreover, the cost-sensitive semi-supervised learning method proposed in this paper can also be applied to crash data analysis with other types of missing labels, for example, crash severity, crash involved users, etc. The main contributions of this paper are summarised as below:

- We formulate the logistic regression model in a form of minimum classification error function. By embedding the misclassification costs to minimise the overall misclassification loss, the cost-sensitive logistic regression algorithm is developed to deal with the class imbalance issue in hit-and-run analysis;
- We adopt the classification maximum likelihood criterion to infer the label information of unlabelled historical crash data. A cost-sensitive semi-supervised maximum likelihood framework is developed in an iterative manner for performing cost-sensitive likelihood estimation of logistic regression model and deriving label information jointly with labelled and unlabelled data with pseudo labels;
- Comparing the prediction results of the proposed model with two logistic regression models and seven machine learning methods, the effectiveness of proposed method for hit-and-run analysis with imbalance data distribution and missing label problem is demonstrated;
- The most significant contributing factors to hit-and-run crashes are determined and explained, which demonstrate a high degree of consistency with the true contributing factors obtained by supervised cost-sensitive logistic regression model with complete hit-and-run labels;
- The effects of proportion of labelled historical data, class-weighted ratio and hyper-parameter λ on the performance of hit-and-run crash prediction model are investigated;
- The result can provide some implications for policies and countermeasures for hit-and-run crashes.

2. Related work

2.1. Hit-and-run

Since the last decade, the hit-and-run behaviour of drivers in crashes has been investigated by examining the circumstances and situations where the crash occurred. Kim et al. (2008), Tay et al. (2010), Zhang et al. (2014), Tay et al. (2008) applied logistic regression model to identify the factors that contribute to the hit-and-run crashes or hit-and-run fatal crashes. Benson et al. (2018) analysed the hit-and-run crashes with descriptive statistics. Some other methods have also been adopted in the literature for analysing hit-and-run crashes, for example, geographically regression model (Liu et al., 2018), association rule analysis (Sivasankaran and Balasubramanian, 2018a), classification and regression tree model (Sivasankaran and Balasubramanian, 2018b).

Recently, researchers also focused on specific aspects of hit-and-run behaviour with logistic regression model or hierarchical bayesian binary logit model with random effects. Zhou et al. (2016) considered six different improper driving behaviours; Roshandeh et al. (2016) principally investigated the difference between distracted and non-distracted drivers; Jiang et al. (2016) focused on crashes in urban river-crossing road tunnels; Xie et al. (2017, 2018) was dedicated to an urban freeway with the relatively congested upstream traffic conditions. There were also studies investigated into vehicle-pedestrian hit-and-run crashes (Solnick and Hemenway, 1995; MacLeod et al., 2012; Aidoo et al., 2013; Fujita et al., 2014) and vehicle-bicycle hit-and-run crashes (Bahrololoom et al., 2017; Lopez et al., 2018; Das et al., 2018; Zhou et al., 2019; Zhu, 2020).

2.2. Missing labels & semi-supervised learning

The missing label problem is a critical issue in crash analysis as the crash dataset for analysis does not always have label information historically collected, for example, due to data barriers (Alharthi et al., 2017; Conradie and Choenni, 2014; Janssen et al., 2012). Although many different methods have been applied to analyse the crashes, to the authors' best knowledge, most of them were supervised methods with the complete label assumption. To address this issue, the possible solution that aims to manually relabel all of the unlabelled historical data is infeasible in reality as the human annotation process is boring, time-consuming, expensive and sometimes may require expert or special devices, in which case semi-supervised learning is a solution.

The semi-supervised learning saves the expensive costs of data relabelling, which requires the relabelling process for only a few historical crashes to conduct hit-and-run analysis. The method performs well in both theory and practice as it requires less human effort and gives better accuracy (Zhu, 2005). Typically, the semi-supervised learning algorithm aims to utilise large amount of unlabelled data, together with labeled data to achieve better performance (Tanha et al., 2017). In the field of transportation, the semi-supervised learning method has been applied for pedestrian counting (Tan et al., 2011), real-time driver distraction detection (Liu et al., 2015), driving safety monitoring (Wang et al., 2010) and travel mode classification (Zhu et al., 2015).

2.3. Class imbalance problem

The analysis and classification on imbalanced dataset is a common problem for hit-and-run crash modelling as the proportion of hit-and-run crashes among all crashes is relatively small. According to the literature review, in the case of binary classification, class proportion 1:1 is defined as class balance. On the other hand, the class proportion 1:4/4:1 can lead to a mild imbalance scenario, and 1:15/15:1 refer to a moderate imbalance scenario, such that the class imbalance problem should be addressed for these scenarios (Brzezinski et al., 2019; Weiss, 2013). However, for hit-and-run analysis, to the authors' best knowledge, only Zhou et al. (2019) resampled the hit-and-run crash dataset to tackle the class imbalance problem. In the general field of transportation, the imbalanced classification problem has attracted much attention of researchers. In recent studies, basically, there are two methods to solve the imbalanced classification issue (Ke et al., 2019): (1) cost-sensitive learning method, which paid more attention to the minority class by altering the weights assigned to various classes (Kuang et al., 2019; Dabiri et al., 2020), (2) resampling of the original dataset by means of discarding majority non-hit-and-run samples or generating new minority hit-and-run samples, i.e., under-sampling and over-sampling, respectively. However, the data resampling method is likely to distort the distribution of training data (Seiffert et al., 2008; Weiss et al., 2007) and overfit the model (Shi et al., 2019; Parsa et al., 2019; Chen et al., 2020), especially when only limited labelled crashes are available in the crash dataset. In this paper, we concern the cost-sensitive learning method for class imbalance issue in hit-and-run crash dataset.

On the whole, there were mainly two limitations of previous research on hit-and-run analysis. Firstly, the issue of missing labels in the hit-and-run data has not been addressed in the literature, and the information in the unlabelled data was not utilised. Secondly, the class imbalance issue of hit-and-run data has rarely been addressed in the literature. This paper aims to propose a cost-sensitive learning method for semi-supervised hit-and-run analysis, such that the data imbalance issue can be handled and the performance of model can be satisfiable even with only a small proportion of labelled historical data to save the data collection efforts.

3. Data

The hit-and-run crash dataset for model formulation is from the road traffic collision database of Victoria State of Australia between 2013 and 2019, containing 74,534 crashes (VicRoads, 2019). The hit-and-run decision of drivers are reflected by a dichotomous variable. Based on the dataset selected and literature review, 33 predictors, both categorical and numeric, have been considered for analysis. The detail information of the dataset is summarised in Table 1. In the right-hand-side column of the table, the features 'No. of females, males, pedestrians, etc.' refer to the number of females, the number of males, the number of pedestrians, etc., involved in each traffic collision. The feature 'road condition classification by the State-wide Route Numbering Scheme' is defined as follows: 'M' level is for road with a consistent high standard of driving conditions, which has divided carriageways, four traffic lanes, sealed shoulders and line marking easily visible in all weather conditions; 'A' level is for road with similar high standard of driving conditions on a single carriageway; 'B' level is for sealed road that is wide enough for two traffic lines, providing good centre line and edge line marking, shoulders and a high standard of guidepost delineation; 'C' level is for road with generally two lane sealed with shoulders.

As we have introduced in Section 1, there are two main issues in hit-and-run crash dataset:

- Imbalanced data distribution: The hit-and-run label in the crash dataset is highly imbalanced, where the hit-and-run crashes are the minority cases. For example, in crash dataset of Victorian, Australia (2013–2019), the number of crashes with non-hit-and-run label or hit-and-run label is 70,710 and 3824, respectively. The imbalanced ratio is up to 18:1.
- Missing label problem: Note that the current dataset presented here is with complete hit-and-run labels. We do not choose the crash dataset that really without the hit-and-run label (such as Toronto Police Service (2019), Seattle Department of Transportation (2018)) is because of the feasibility of experiment and performance evaluation. Specifically, in order to formulate the scenario of semi-supervised hit-and-run analysis, i.e., missing label problem as

Table 1
Descriptive statistics of Victorian hit-and-run crash dataset.

Variable	Description	Count
Hit-and-run	No	70,710
	Yes	3824
Month	Jan	5952
	Feb	6252
	Mar	6513
	Apr	5649
	May	5895
	Jun	5234
	Jul	6431
	Aug	6402
	Sep	5905
	Oct	6912
	Nov	6743
	Dec	6646
Year	2013	6832
	2014	14,160
	2015	14,369
	2016	14,282
	2017	12,217
	2018	11,118
	2019	1556
Day of week	Monday	10,312
	Tuesday	10,766
	Wednesday	11,067
	Thursday	11,301
	Friday	11,320
	Saturday	8772
	Sunday	9525
Collision type	Rear end	13,155
	Right through	6545
	Cross traffic	5198
	Left off carriageway into object/parker vehicle	4259
	Right near	3416
	out of control on carriageway	3197
Light condition	Dark no street lights	4038
	Dark street lights off	163
	Dark street lights on	11,261
	Dark street lights unknown	742
	Day	48,780
	Dusk/Dawn	7724
Road Geometry	Cross intersection	16,683
	Multiple intersection	1626
	Not at intersection	38,380
	T intersection	17,507
	Y intersection	132
Severity	Fatal crash	1304
	Serious injury crash	21,433
	Other injury crash	51,796
	Non injury crash	1
Speed zone	40 km/h	4385
	50 km/h	12,321
	60 km/h	24,870
	70 km/h	4962
	80 km/h	10,642
Run off-road	No	60,626
	Yes	13,908
Alcohol related	No	72,067
	Yes	2467
Local government area	Melbourne	4192
	Casey	3041
	Geelong	2805
	Dandenong	2544
	Hume	2372
	Darebin	251
	Brimbank	2336
Type of urbanised area	Large Provincial Cities	4201
	Metropolitan urban area	46,811
	Metropolitan CBD	1023
	Rural Victoria	15,780
	Small provincial cities	3774
	Small towns	653
	Other Towns	2292
No. of males	0–46	74,534

(continued on next page)

Table 1 (*continued*)

Variable	Description	Count
No. of females	0–51	74,534
No. of bicyclists	0–8	74,534
No. of vehicle passengers	0–95	74,534
No. of drivers	0–12	74,534
No. of pedestrians	0–9	74,534
No. of pillions	0–2	74,534
No. of motorcyclists	0–5	74,534
No. of 5–12 year old pedestrians/ cyclists	0–8	74,534
No. of 13–18 year old pedestrians/ cyclists	0–3	74,534
No. of 65 years and older pedestrians	0–3	74,534
No. of 65 years and older drivers	0–3	74,534
No. of 18–25 year old young drivers	0–5	74,534
No. of unlicensed drivers	0–2	74,534
No. of heavy vehicles	0–4	74,534
No. of passenger vehicle	0–13	74,534
No. of motorcycles	0–5	74,534
No. of public vehicles	0–2	74,534
Road classification	Arterial highway	13,863
	Arterial other	27,106
	Freeway	5464
	Local road	26,578
	Non-arterial	16
Road condition by the Statewide Route Numbering Scheme	A	2706
	B	3045
	C	9768
	M	6058
Crash occur on a divided portion of road	Divided	25,437
	Undivided	47,590

introduced in Section 1, we assume that no hit-and-run information has been collected historically and various proportions of hit-and-run labels can be manually relabelled. Then, both the relabelled data and the remaining unlabelled historical data (with unknown hit-and-run labels) are utilised to test the performance of cost-sensitive semi--supervised learning algorithm proposed in this paper.

4. Methodology

The cost-sensitive semi-supervised learning technique is applied such that both labelled data and unlabelled data can be utilised to construct the classifier for hit-and-run crashes. Suppose the hit-and-run crash dataset $\mathrm{X} = [\mathrm{x}_1, \ldots, \mathrm{x}_N] \in \mathbb{R}^{D\times N}$ contains N number of crash records each from the D-dimensional explanatory feature space. Let $y_i = +1$ or -1 represents the possible behaviour of "run" or "not-run" happened after the ith crash for $i = 1, \ldots, N$, respectively. Considering the missing label problem and the expensive cost of relabelling the hit-and-run crashes, only N_l number of crashes in **X**, i.e., $\mathrm{X}_l = [\mathrm{x}_1, \ldots, \mathrm{x}_{N_l}]$, are assumed to be supervised with the manually relabelled hit-and-run label information $\mathrm{Y}_l = [y_1, \ldots, y_{N_l}]^T$. The remaining N_u number of crashes in **X**, i.e., $\mathrm{X}_u = [\mathrm{x}_{N_l+1}, \ldots, \mathrm{x}_N]$, are unsupervised with unknown hit-and-run labels. In such a semi-supervised scenario, we usually have $N_l \ll N_u$ and $N = N_l + N_u$.

4.1. Motivation

4.1.1. Semi-supervised learning problem

Recall that most of the statistical methods such as logit and probit models (Borooah, 2002) usually aim to find the relation **w** between the input explanatory variable **X** and the output behaviour vector **Y** by maximising the likelihood in the form of

$$\max_{\mathrm{w}} : L(\mathrm{X}, \mathrm{Y}, \mathrm{w}), \tag{1}$$

where $L(\cdot)$ is the likelihood function defined according to different data distribution assumptions.

Observing Eq. (1), we find that the likelihood function is defined in a supervised manner. The robustness of learned optimisation variable **w** not only depends on the likelihood function $L(\cdot)$ but also the supervised label information **Y**. In the past decades, researchers usually assumed the supervised crash dataset and focused on designing a more robust likelihood function to obtain an unbiased estimation of **w**. However, due to the missing label problem and the difficulties in relabelling data, only limited supervised crash data $\mathbf{X}_l$ are available to perform parameter estimation. Therefore, the **w** learned by Eq. (1) is probably biased even if the most robust likelihood function is adopted.

4.1.2. Class-imbalanced data distribution

For different statistical models, Eq. (1) usually can be further expressed as

$$\max_{\mathrm{w}} : \sum_{i=1, y_i=+1}^{N} f(\mathrm{x}_i, y_i, \mathrm{w}) + \sum_{i=1, y_i=-1}^{N} f(\mathrm{x}_i, y_i, \mathrm{w}), \tag{2}$$

where $f(\cdot)$ is the logistic or normal distribution function of logistic or probit model, respectively. $N = N_{+1} + N_{-1}$ where $N_{+1} = \sum_{i=1, y_i=+1}^{N} \mathrm{sgn}(y_i)$ and $N_{-1} = \sum_{i=1, y_i=-1}^{N} \mathrm{sgn}(-y_i)$ denote the number of behaviours of hit-and-run and not-run happened in crash dataset **X**, respectively. The step function $\mathrm{sgn}(+1) = 1$ and $\mathrm{sgn}(-1) = 0$.

Obviously, if the data distribution is balanced, i.e., $N_{+1} = N_{-1}$, then the estimated **w** will be unbiased with respect to any classes. However, as we have introduced before, in crash dataset, the proportion of hit-and-run cashes among all cashes is relatively small, i.e., $N_{+1} \ll N_{-1}$. In such a class-imbalanced scenario, the learned **w** will probably be overwhelmed by the majority non-hit-and-run class and thus ignore the minority hit-and-run class.

4.2. The proposed cost-sensitive semi-supervised learning framework

To deal with the imbalanced classification problem, cost-sensitive learning is the most widely used method on the algorithmic level, which considers the various costs for misclassifying various classes (Ke et al., 2019). Obviously, a larger cost value should be assigned to the minority class, such that misclassifying the sample from minority class (hit-and-run crash) leads to higher penalty. Furthermore, motivated by the classification maximum likelihood approach (Symons, 1981; Amini and Gallinari, 2002), we aim to take the label information of unlabelled data as an additional parameter and propose to maximise the cost-sensitive likelihood of supervised and unsupervised data simultaneously. In this way, both the semi-supervised and class-imbalanced problems can be effectively solved in a cost-sensitive semi-supervised maximum likelihood framework defined as follows

$$\max_{\mathrm{w}, \widetilde{\mathrm{Y}}_u} : L(\mathrm{X}_l, \mathrm{Y}_l, \mathrm{C}, \mathrm{w}) + \lambda L(\mathrm{X}_u, \widetilde{\mathrm{Y}}_u, \mathrm{C}, \mathrm{w}), \tag{3}$$

where $\widetilde{\mathrm{Y}}_u = [\widetilde{y}_{N_l+1}, \ldots, \widetilde{y}_N]^T$ denotes the estimated label vector of unlabelled crash data $\mathbf{X}_u$. Symbol $\lambda \in [0, 1]$ is a scale-balance hyper-parameter pre-learned by cross-validation. Cost vector $\mathbf{C} = [c_+, c_-]$ where c_+ or c_- denotes the cost of misclassifying a positive or negative data, respectively. Usually, we set $c_+ > c_-$ to control the balance between the minority and majority class for parameter estimation of **w** and $\widetilde{\mathrm{Y}}_u$.

Observing Eq. (3), it is the canonical form of maximum likelihood approach with the additional optimisation variable $\widetilde{\mathrm{Y}}_u$ and user-defined cost vector **C**. Given an initial input of unsupervised label vector $\widetilde{\mathrm{Y}}_u$ and the cost vector **C**, the parameter **w** will be obtained by maximising the complete and cost-sensitive data likelihood. Then, the **w** can be used in turn to evaluate the $\widetilde{\mathrm{Y}}_u$ with the cost vector **C**. This process is iterated until the maximum cost-sensitive likelihood is achieved. In this way, both the likelihood parameter **w** and label vector $\widetilde{\mathrm{Y}}_u$ are jointly optimised in a unified cost-sensitive maximum likelihood framework. Therefore, due to the supervised label information in $\widetilde{\mathrm{Y}}_u$ and cost information in **C**, an unbiased likelihood parameter **w** can be obtained. Fig. 1 reports the flowchart of proposed cost-sensitive semi-supervised learning framework for hit-and-run analysis.

4.3. *Cost-sensitive semi-supervised logistic regression*

In this section, we elaborate our proposed cost-sensitive semi-supervised learning framework. Without loss of generality, we use the log-odd as the likelihood function and thus obtain the cost-sensitive likelihood of supervised and unsupervised data defined as follows

$$\begin{cases} L(\mathrm{X}_l,\mathrm{Y}_l,\mathrm{C},\mathrm{w}) = c_+ \sum_{i=1,y_i=+1}^{N_l} f(\mathrm{x}_i,y_i,\mathrm{w}) + c_- \sum_{i=1,y_i=-1}^{N_l} f(\mathrm{x}_i,y_i,\mathrm{w}) \\ L(\mathrm{X}_u,\widetilde{\mathrm{Y}}_u,\mathrm{C},\mathrm{w}) = c_+ \sum_{i=N_l+1,\widetilde{y}_i=+1}^{N} f(\mathrm{x}_i,\widetilde{y}_i,\mathrm{w}) + c_- \sum_{i=N_l+1,\widetilde{y}_i=-1}^{N} f(\mathrm{x}_i,\widetilde{y}_i,\mathrm{w}) \\ f(\mathrm{x}_i,y_i,\mathrm{w}) = log(\Pr(y_i|\mathrm{x}_i,\mathrm{w})) = -log(1+exp(-y_i\mathrm{w}^T\mathrm{x}_i)) \\ f(\mathrm{x}_i,\widetilde{y}_i,\mathrm{w}) = log(\Pr(\widetilde{y}_i|\mathrm{x}_i,\mathrm{w})) = -log(1+exp(-\widetilde{y}_i\mathrm{w}^T\mathrm{x}_i)) \end{cases} \tag{4}$$

where $\Pr(y_i|\mathbf{x}_i,\ \mathbf{w})$ denotes the posterior probability that data $\mathbf{x}_i$ belongs to class y_i. Suppose the matrix $\widetilde{\mathrm{X}}$ and vector $\widetilde{\mathrm{w}}$ are all augmented, i.e., $\widetilde{\mathrm{X}} = [\mathbf{1};\mathrm{X}] \in \mathbb{R}^{(D+1)\times N}$ and $\widetilde{\mathrm{w}} = [w_0;\mathrm{w}]$. For sake of simplicity, we let the $\mathbf{X} = [\mathbf{X}_l,\ \mathbf{X}_U]$ and **w** in Eq. (4) denote the augmented $\widetilde{\mathrm{X}}$ and $\widetilde{\mathrm{w}}$, respectively.

According to Liu et al. (2009), Fan et al. (2008), the term $\log(1+\exp(-y_i\mathbf{w}^T\mathbf{x}_i))$ in Eq. (4) in fact measures the classification error of $\mathbf{x}_i$ in a logistic manner if the **w** is viewed as a classifier for binary classification not only the parameter of likelihood. This conclusion is reasonable because $\Pr(y_i|\mathbf{x}_i,\ \mathbf{w}) \propto -\log(1+\exp(-y_i\mathbf{w}^T\mathbf{x}_i))$ due to the monotone increasing function $\log(\cdot)$. Obviously, with the increasing of $\Pr(y_i|\mathbf{x}_i,\mathbf{w})$, the corresponding classification error function $\log(1+\exp(-y_i\mathbf{w}^T\mathbf{x}_i))$ will decrease proportionally. By multiplying the classification error with the corresponding cost, i.e., c_{+1} for $y_i=+1$ and c_{-1} for $y_i=-1$, then the cost-sensitive semi-supervised maximum likelihood framework in Eq. (3) can be formulated in a form of logistic loss function that minimises the overall misclassification loss of crash data **X**. Thus, the proposed *cost-sensitive semi-supervised logistic regression* (CS^{3}LR) can be expressed as

$$\begin{aligned} &\underset{\mathrm{w},\widetilde{\mathrm{Y}}_u}{min}:\mathrm{loss}(\mathrm{w},\widetilde{\mathrm{Y}}_u)=c_+\sum_{i=1,y_i=+1}^{N_l} log(1+exp(-\mathrm{w}^T\mathrm{x}_i)) \\ &+c_-\sum_{i=1,y_i=-1}^{N_l} log(1+exp(\mathrm{w}^T\mathrm{x}_i))+\lambda c_+\sum_{i=N_l+1,\widetilde{y}_i=+1}^{N} log(1+exp(-\mathrm{w}^T\mathrm{x}_i)) \\ &+\lambda c_-\sum_{i=N_l+1,\widetilde{y}_i=-1}^{N} log(1+exp(\mathrm{w}^T\mathrm{x}_i))s.t.,\frac{1}{N_u}\sum_{i=N_l+1}^{N}\mathrm{sgn}(\widetilde{y}_i)=r,\ \widetilde{y}_i\in\{+1,-1\} \end{aligned} \tag{5}$$

where the additional balance constraint is proposed to avoid the trivial solution that assigns all the unlabelled instances to the same class. The user-defined parameter r is determined according to the class distribution of supervised data, i.e., $r = \frac{1}{N_l}\sum_{i=1}^{N_l}\mathrm{sgn}(\mathrm{y}_i)$.

5. Method analysis

5.1. *Convergence analysis*

For the proposed objective function in Eq. (5), we use an iterative algorithm which is similar to the famous expectation maximisation (EM) approach (Dempster et al., 1977) to estimate the parameters of **w** and $\widetilde{\mathrm{Y}}_u$. Specifically, as shown in Algorithm 1, the iterative algorithm updates one variable at a time by fixing all the other variables in every step of an iteration. The updating steps of the algorithm in the tth iteration, $t=0,\ 1,\ \ldots,$ are described as follows:

Algorithm 1. The optimisation process of proposed CS^{3}LR

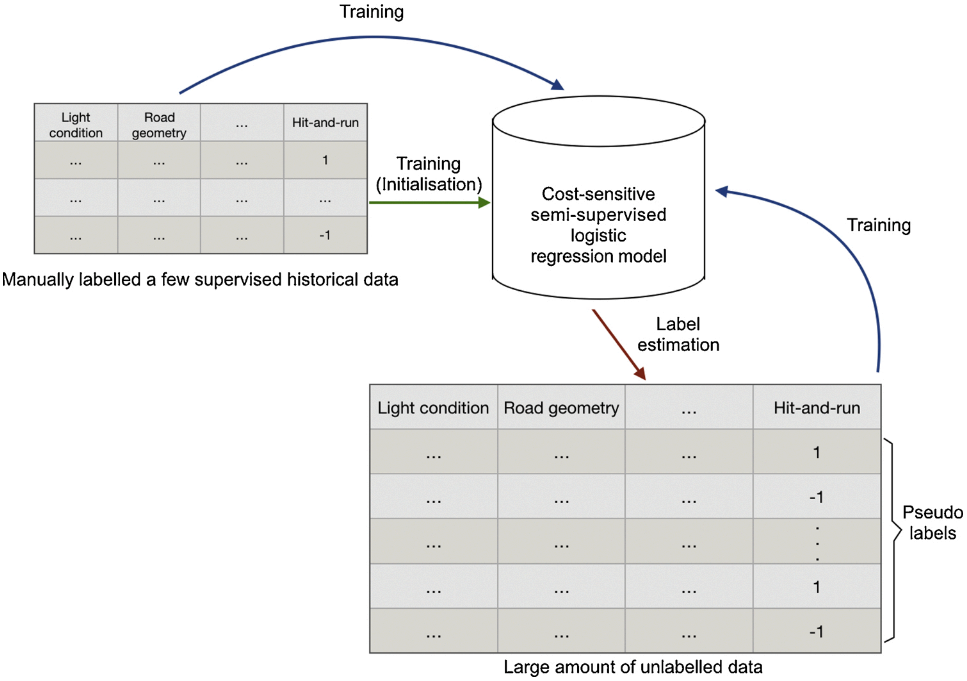
```
Input: Hit-and-run crash dataset (set of labelled data $\mathbf{X}_l$, set of unlabelled data $\mathbf{X}_u$);
       hyper-parameter $\lambda$ and maximum number of iteration $I$.
Output: The likelihood parameter $\mathbf{w}$
Let $t = 0$ and initialise $\mathbf{w}^{[t]}$ by performing logistic regression with labelled data $\mathbf{X}_l$.
while $t < I$ do
    E-Step: Calculate the $\mathrm{Diff}(i)$ for $i = N_l + 1, \cdots, N$ by using Eq. (7) with $\mathbf{w}^{[t]}$.
    C-Step: Update the $\widetilde{\mathbf{Y}}_u^{[t+1]}$ by ranking the $N_u$ number of $\mathrm{Diff}(i)$ in an ascending order and
      choosing the first $r$ percentage data as the positive class "+1". The remaining data are
      considered as the negative class "−1".
    M-Step: Estimate the likelihood parameter $\mathbf{w}^{[t+1]}$ by using the LIBLINEAR package with
      $\widetilde{\mathbf{Y}}_u^{[t+1]}$.
    $t = t + 1$.
end
```

(6) Update the $\widetilde{\mathrm{Y}}_u^{[t+1]}$: By fixing the **w**, maximising Eq. (5) with respect to $\widetilde{\mathrm{Y}}_u$ is equivalent to the cost-sensitive logistic loss function in the form of

$$\underset{\widetilde{\mathrm{Y}}_u}{min}:c_+\sum_{i=N_l+1,\widetilde{y}_i=+1}^{N} log(1+exp(-\mathrm{w}^T\mathrm{x}_i))+c_-\sum_{i=N_l+1,\widetilde{y}_i=-1}^{N} log(1+exp(\mathrm{w}^T\mathrm{x}_i))$$

$$s.t.,\frac{1}{N_u}\sum_{i=N_l+1}^{N}\mathrm{sgn}(\widetilde{y}_i)=r,\ \widetilde{y}_i\in\{+1,-1\} \tag{6}$$

Recall by definition that $\widetilde{y}_i$ is a one-hot vector. Thus, for $i=N_l+1,\ \ldots,\ N$, we enumerate all the 2 possible solutions, i.e., $\widetilde{y}_i=+1$ or -1, and find the one that minimises Eq. (6) with the $\mathbf{w}^{[t]}$ updated in tth


Training
Training (Initialisation)
Cost-sensitive semi-supervised logistic regression model
Training
Label estimation
Pseudo labels

Light condition	Road geometry	...	Hit-and-run
...	...	...	1
...	...	...	...
...	...	...	-1

Manually labelled a few supervised historical data

Light condition	Road geometry	...	Hit-and-run
...	...	...	1
...	...	...	-1
...	...	...	⋮
...	...	...	1
...	...	...	-1

Large amount of unlabelled data

Fig. 1. The flowchart of proposed learning framework.

iteration. However, considering the balance constraint, we propose to calculate the difference of logistic loss resulted by 2 possible choices of $\widetilde{y}_i$, i.e., Diff(i), for $i = N_l + 1, \ldots, N$.

$$\text{Diff}(i) = c_+ log(1 + exp(-\mathrm{w}^{[t]T}\mathrm{x}_i)) - c_- log(1 + exp(\mathrm{w}^{[t]T}\mathrm{x}_i)). \tag{7}$$

Then we rank the Diff(i) in an ascending order and choose the first r percentage data as the positive class "+1" and the remaining data are considered as the negative class "−1". By doing so, we can minimise the logistic loss with the balance constraint.

(7) Update the parameter $\mathbf{w}^{[t+1]}$: By fixing the $\widetilde{\mathrm{Y}}_u$, maximising Eq. (5) with respect to $\mathbf{w}$ is equivalent to

$$\min_{\mathrm{w}} : c_+ \sum_{i=1, y_i=+1}^{N_l} log(1+exp(-\mathrm{w}^T\mathrm{x}_i)) + c_- \sum_{i=1, y_i=-1}^{N_l} log(1+exp(\mathrm{w}^T\mathrm{x}_i))$$
$$+\lambda c_+ \sum_{i=N_l+1, \widetilde{y_i}=+1}^{N} log(1+exp(-\mathrm{w}^T\mathrm{x}_i)) + \lambda c_- \sum_{i=N_l+1, \widetilde{y_i}=-1}^{N} log(1+exp(\mathrm{w}^T\mathrm{x}_i)) \tag{8}$$

which is the canonical form of weighted logistic regression (Fan et al., 2008) with the weights of c_+ and c_- for minority and majority class, respectively. We can thus update the $\mathbf{w}^{[t+1]}$ with $\widetilde{\mathrm{Y}}_u^{[t+1]}$ by using the famous LIBLINEAR package.[1]

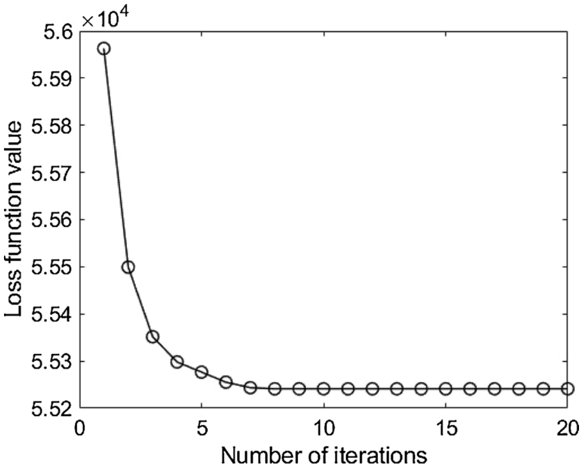


Fig. 2. Loss function value versus number of iterations. The road traffic collision database of Victoria State of Australia between 2013 and 2019 is adopted here where only 10% historical crash data are manually relabelled with the hit-and-run label information.

In the following, we discuss the convergence of proposed CS^{3}LR algorithm. Firstly, the minimisation problem in Eq. (6) for label estimation of unlabelled data, i.e., $\widetilde{\mathrm{Y}}_u$, though not necessarily convex, ensures that the overall loss function is at least non-increasing. For the weighted logistic regression in Eq. (8), the overall loss function is strictly convex with one global minimum value (Fan et al., 2008). Thus, the loss function is strictly decreasing when updating $\mathbf{w}$ with quasi Newton method implemented in LIBLINEAR package. Accordingly, for all $t = 0, 1, \ldots$, we have

$$\text{loss}(\mathrm{w}^{[t]}, \widetilde{\mathrm{Y}}_u^{[t]}) \geq \text{loss}(\mathrm{w}^{[t]}, \widetilde{\mathrm{Y}}_u^{[t+1]}) > \text{loss}(\mathrm{w}^{[t+1]}, \widetilde{\mathrm{Y}}_u^{[t+1]}). \tag{9}$$

For illustration purpose, Fig. 2 plots the overall loss function value of CS^{3}LR in Eq. (5) achieved in each round of iteration, where we can observe that the proposed CS^{3}LR can converge after 10 times of iteration.

5.2. Behaviour analysis

After the unbiased estimation of $\mathbf{w}$ has been obtained by minimising

[1] https://www.csie.ntu.edu.tw/~cjlin/liblinear.

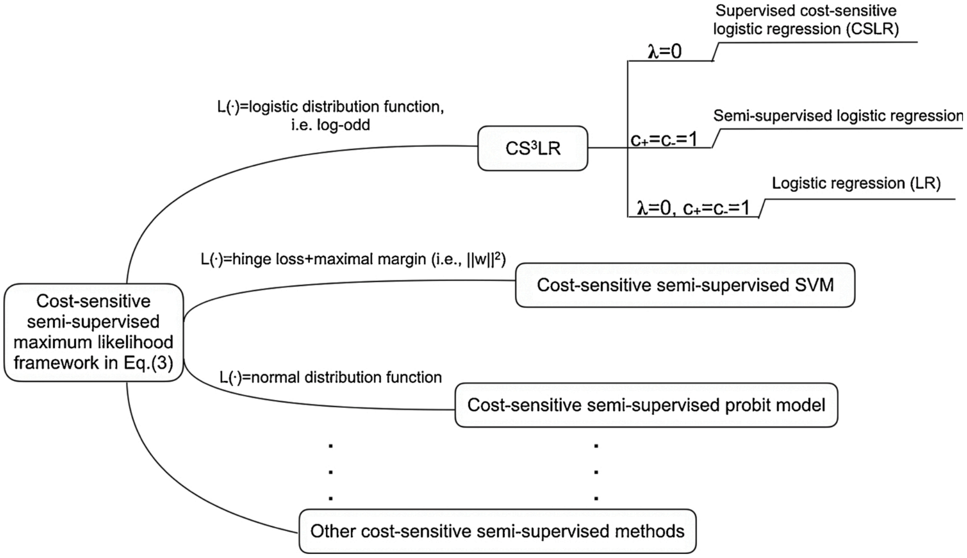

Supervised cost-sensitive logistic regression (CSLR)
λ=0
L(·)=logistic distribution function, i.e. log-odd
CS³LR
Semi-supervised logistic regression
c₊=c₋=1
Logistic regression (LR)
λ=0, c₊=c₋=1
L(·)=hinge loss+maximal margin (i.e., ||w||²)
Cost-sensitive semi-supervised SVM
Cost-sensitive semi-supervised maximum likelihood framework in Eq.(3)
L(·)=normal distribution function
Cost-sensitive semi-supervised probit model
Other cost-sensitive semi-supervised methods

Fig. 3. Extension to other methods.

Eq. (5), we perform behaviour analysis to find out the positive and negative factors of hit-and-run problem. Specifically, according to (Borooah, 2002), we take the derivative of $\Pr(y_i|\mathrm{x}_i, \mathrm{w}) = \frac{1}{1+exp(-y_i\mathrm{w}^T\mathrm{x}_i)}$ with respect to each explanatory variable x_{ki} for $k = 1, \ldots, D$ in crash $\mathbf{x}_i$, and then have

$$\frac{\partial \Pr(y_i|\mathrm{x}_i, \mathrm{w})}{\partial x_{\mathrm{ki}}} = y_i w_k (1 + exp(-y_i\mathrm{w}^T\mathrm{x}_i))^{-2} exp(-y_i\mathrm{w}^T\mathrm{x}_i). \tag{10}$$

Obviously, for the purpose of hit-and-run analysis, i.e., let $y_i = +1$ and $\Pr(y_i|\mathbf{x}_i, \mathbf{w}) = \Pr(+1|\mathbf{x}_i, \mathbf{w})$, we have

$$\begin{cases} \dfrac{\partial \Pr(+1|\mathrm{x}_i, \mathrm{w})}{\partial x_{\mathrm{ki}}} > 0, & \text{if} \quad w_k > 0 \\ \dfrac{\partial \Pr(+1|\mathrm{x}_i, \mathrm{w})}{\partial x_{\mathrm{ki}}} < 0, & \text{if} \quad w_k < 0 \end{cases} \tag{11}$$

From Eq. (11), we can find that the positive or negative factors of hit-and-run problem can be determined according to their corresponding likelihood parameter, i.e., if $w_k > 0$ for $k = 1, \ldots, D$, then the derivative in Eq. (10) becomes a positive value and thus the explanatory variable x_{ki} has a positive effects on the posterior probability $\Pr(+1|\mathbf{x}_i, \mathbf{w})$. Similarly, if $w_k < 0$, then the explanatory variable x_{ki} has a negative effects on the $\Pr(+1|\mathbf{x}_i, \mathbf{w})$.

5.3. Extension to other methods

The proposed CS^{3}LR model in Eq. (5) can be easily extended to many other methods for hit-and-run analysis. Specifically, if we let hyper-parameter $\lambda = 0$, then the CS^{3}LR model is degenerated into the supervised cost-sensitive logistic regression (CSLR) with only a few relabelled historical crashes; if the cost values are set to $c_+ = c_- = 1$, then the CS^{3}LR becomes the cost insensitive logistic regression model with both labelled and unlabelled crash data; if the parameter $\lambda = 0$ and $c_+ = c_- = 1$, then the likelihood function of CS^{3}LR and logistic regression (LR) (Hastie et al., 2009) are the same.

		Actual label	
		Positive	Negative
Predicted label	Positive	True Positive	False Positive
	Negative	False Negative	True Negative

Fig. 4. Confusion matrix.

It is worth noting that the base learner of logistic regression in Eq. (5) can also be extended to other types of learning model, as illustrated in Fig. 3. For example, we can use the probit model (Borooah, 2002) or machine learning model of support vector machine (SVM) (Hastie et al., 2009) to elaborate the proposed cost-sensitive semi-supervised learning framework in Eq. (3).

6. Experimental results

6.1. Data preparation

To prepare the data for analysis, the categorical features in the hit-and-run crash dataset in Section 3 are sparsified such that 258 dummy variables or predictors are created. Among them, zero variance or near zero variance predictors exist, which take constant or almost constant value across the entire crash dataset (Kuhn and Johnson, 2013). These predictors are identified according to two criteria: (1) the ratio of the number of unique values relative to the number of samples is low (less than 10%); (2) the ratio of the most common value's frequency to the second most common value's frequency is high (greater than 95). Zero variance or near zero variance predictors are further eliminated in this stage, as they uninformative and compromise the accuracy of model, such that 67 predictors are retained. Correlation test is also applied afterwards and 63 variables are finally retained for the experiment. In our experiments, the total 74,534 crash records are divided into two sets, where 80% of the data are selected as the training data and the remaining 20% of the data are in the test set.

6.2. Measure of effectiveness

Since the hit-and-run crash dataset is highly imbalanced, different evaluation metrics other than the overall accuracy need to be considered (Li et al., 2014). Confusion matrix is applied to describe the classification model, where samples are categorised into True Positive (TP), False Positive (FP), True Negative (TN), False Negative (FP) (Fig. 4).

Based on the confusion matrix, the metrics of sensitivity, specificity, total cost, G-mean and F-measure values are calculated by Eqs. (12)–

(16) respectively:

$$\text{sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{12}$$

$$\text{specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} \tag{13}$$

$$\text{total} \quad \text{cost} = c_{+} * \text{FN} + c_{-} * \text{FP} \tag{14}$$

$$G - \text{mean} = \sqrt{\text{sensitivity} \times \text{specificity}} \tag{15}$$

$$F - \text{measure} = \frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}} \tag{16}$$

Receiver operating characteristic (ROC) corresponds to various specific values of true positive rate (sensitivity) and false positive rate (1-specificity). The area under the ROC curve (AUC) which also summarises how well the classifier differentiate the two classes is included for model performance evaluation as well.

6.3. *Comparison results of CS^{3}LR & LR/CSLR*

In this section, we vary the proportion of manually labelled historical hit-and-run crash data in training set from 1% to 100% to compare the performance of LR model commonly applied in hit-and-run analysis literature, CSLR model (supervised cost-sensitive logistic regression model) and CS^{3}LR model (i.e., cost-sensitive semi-supervised learning model) proposed in this paper. In order to follow the independent identically distributed assumption introduced in Section 1 for semi-supervised learning, in our experimental part, we assume that the ground-truth ratio of positive/negative class (hit-and-run/non-hit-and-run crashes) is known and then generate the labelled and unlabelled data in training set, as well as the test set with the known class imbalance ratio accordingly. In other words, a constant class-weighted ratio is applied to test the performance of examined models under various proportions of manually labelled historical hit-and-run crash data. For the three types of models considered in this section, LR model does not tackle the class imbalance issue and the unlabelled training data is not utilised; CSLR model further address the class imbalance issue, but the unlabelled training data is not utilised either; lastly, the CS^{3}LR model solves both the class imbalance issue and missing label issue. Please note that refer to Fig. 3 mentioned in previous section, both LR model and CSLR model are special cases of the proposed CS^{3}LR model. Based on the parameter sensitivity analysis in Section 6.6.1, we set class-weighted ratio $\gamma = \frac{c_{+}}{c_{-}} = 20:1$ for proposed CS^{3}LR. Since the random generation of missing labels can give various results for various trials, the experiment for each missing ratio is repeated ten times and the mean and standard deviation of total cost, AUC, G-mean, F-measure, sensitivity, and specificity values are computed, as shown in Table 2. For a more rigorous study of the experimental results, we also conduct a paired *t*-test at significance level $\alpha = 0.05$ (Rice, 2006) in Table 2 to show the statistical difference between the proposed CS^{3}LR model and the LR/CSLR models. Some results and findings are discussed as follows:

Comparing the results of LR model and CSLR model in Table 2, it is observed that the CSLR model always wins on the metrics of total cost, G-mean and F-measure, no matter the proportion of manually labelled data is high or low. The results can be explained by the effective consideration of class imbalance issue in the CSLR model. The values of AUC generated from the two models are relatively comparable. Similar to Ke et al. (2019), this can be explained by the inherit property of AUC, as AUC value is not decided by certain specificity and sensitivity value, but a set of thresholds, such that AUC is not sensitive to the imbalanced data. The better specificity value in LR model is probably due to the trade-off between the value of specificity and sensitivity where LR model in Table 2 can always achieve a low value of sensitivity on identifying a real hit-and-run crash case.

Furthermore, comparing the results of CSLR model and the proposed

Table 2
Comparison of CS^{3}LR and LR/CSLR with different proportion of manually labelled data.

Model	Total cost (↓)	AUC (↑)	G-Mean (↑)	F-measure (↑)	Sensitivity (↑)	Specificity (↑)
Manually labelling 1% historical hit-and-run crash data						
LR	13,946 ± 474	0.6097 ± 0.032	0.3358 ± 0.056	0.1318 ± 0.028	0.1207 ± 0.041	0.9639 ± 0.012
CSLR	12,767 ± 570	0.6024 ± 0.032	0.5129 ± 0.029	0.1460 ± 0.016	0.3153 ± 0.035	0.8370 ± 0.017
CS^{3}LR	**11,096** ± 652	**0.6687** ± 0.025	**0.6226** ± 0.024	**0.1514** ± 0.012	0.5768 ± 0.036	0.6726 ± 0.019
Paired *t*-test with $\alpha = 0.05$ (w/t/l)	2/0/0	2/0/0	2/0/0	1/1/0	/	/
Manually labelling 2% historical hit-and-run crash data						
LR	14,475 ± 496	0.6754 ± 0.022	0.2367 ± 0.065	0.0950 ± 0.044	0.0610 ± 0.037	0.9910 ± 0.047
CSLR	11,193 ± 625	0.6640 ± 0.025	0.6109 ± 0.024	0.1630 ± 0.013	0.5012 ± 0.049	0.7475 ± 0.031
CS^{3}LR	**10,144** ± 342	**0.7063** ± 0.014	**0.6553** ± 0.012	**0.1711** ± 0.007	0.6153 ± 0.022	0.6984 ± 0.016
Paired *t*-test with $\alpha = 0.05$ (w/t/l)	2/0/0	2/0/0	2/0/0	2/0/0	/	/
Manually labelling 5% historical hit-and-run crash data						
LR	14,674 ± 280	0.7387 ± 0.012	0.2022 ± 0.044	0.0763 ± 0.031	0.0431 ± 0.020	0.9963 ± 0.015
CSLR	9673 ± 430	0.7317 ± 0.014	0.6710 ± 0.015	0.1850 ± 0.010	0.6232 ± 0.030	0.7232 ± 0.018
CS^{3}LR	**9315** ± 378	**0.7435** ± 0.012	**0.6823** ± 0.014	**0.2012** ± 0.009	0.6160 ± 0.026	0.7562 ± 0.012
Paired *t*-test with $\alpha = 0.05$ (w/t/l)	2/0/0	1/1/0	2/0/0	2/0/0	/	/
Manually labelling 10% historical hit-and-run crash data						
LR	14,790 ± 0.269	0.7547 ± 0.009	0.1785 ± 0.050	0.0628 ± 0.032	0.0344 ± 0.002	0.9974 ± 0.001
CSLR	9065 ± 265	0.7557 ± 0.009	0.6921 ± 0.009	0.1930 ± 0.007	0.6715 ± 0.023	0.7138 ± 0.020
CS^{3}LR	**8815** ± 279	**0.7601** ± 0.008	**0.6999** ± 0.010	**0.2148** ± 0.009	0.6390 ± 0.018	0.7667 ± 0.012
Paired *t*-test with $\alpha = 0.05$ (w/t/l)	2/0/0	2/0/0	2/0/0	2/0/0	/	/
Manually labelling 100% historical hit-and-run crash data						
LR	15,123 ± 184	0.7721 ± 0.009	0.1055 ± 0.016	0.0220 ± 0.006	0.0114 ± 0.003	0.9988 ± 0.001
CSLR	**8404** ± 199	**0.7807** ± 0.009	**0.7136** ± 0.007	**0.1995** ± 0.004	0.7288 ± 0.012	0.6987 ± 0.004
CS^{3}LR	**8404** ± 199	**0.7807** ± 0.009	**0.7136** ± 0.007	**0.1995** ± 0.004	0.7288 ± 0.012	0.6987 ± 0.004
Paired *t*-test with $\alpha = 0.05$ (w/t/l)	1/1/0	1/1/0	1/1/0	1/1/0	/	/

For the evaluation metrics, smaller (↓) total cost and larger (↑) AUC, G-mean, F-measure, sensitivity, specificity indicate better model performance. Under each setting of the proportion of labelled data, the model with best performance on each evaluation metric is highlighted as bold and the last line shows the win/tie/loss counts of CS^{3}LR versus LR and CSLR models (refer to Fig. 3). Note that, we do not present the *t*-test results on sensitivity and specificity as they are a compromise.

Table 3
Comparing label estimation methods based on CSLR with 1% manually labelled data.

	Total cost (↓)	AUC (↑)	G-Mean (↑)	F-measure (↑)	Sensitivity	Specificity
KNN+CSLR	14,417 ± 224	**0.6748** ± 0.019	0.2616 ± 0.036	0.1029 ± 0.021	0.0709 ± 0.019	0.9844 ± 0.006
LR+CSLR	13,647 ± 500	0.6090 ± 0.032	0.3780 ± 0.050	0.1451 ± 0.026	0.1534 ± 0.041	0.9497 ± 0.011
SVM (linear)+CSLR	13,647 ± 855	0.6467 ± 0.041	0.3496 ± 0.013	0.1379 ± 0.055	0.1486 ± 0.075	0.9549 ± 0.023
SVM (gaussian)+CSLR	12,871 ± 555	0.6423 ± 0.034	0.4908 ± 0.036	0.1506 ± 0.002	0.2801 ± 0.047	0.8677 ± 0.025
SVM (polynomial)+CSLR	15,269 ± 15.2	0.5704 ± 0.030	0.0328 ± 0.022	0.0031 ± 0.018	0.0016 ± 0.001	0.9990 ± 0.001
CS-SVM (linear)+CSLR	11,558 ± 662	0.6479 ± 0.030	0.6034 ± 0.023	0.1483 ± 0.014	0.5212 ± 0.035	0.7000 ± 0.033
CS-SVM (gaussian)+CSLR	11,471 ± 678	0.6576 ± 0.028	0.6072 ± 0.024	0.1486 ± 0.014	0.5345 ± 0.041	0.6918 ± 0.037
CS-SVM (polynomial)+CSLR	11,772 ± 713	0.6432 ± 0.030	0.5934 ± 0.025	0.1469 ± 0.015	0.4927 ± 0.034	0.7158 ± 0.030
CSLR+CSLR	12,721 ± 580	0.5986 ± 0.031	0.5189 ± 0.029	0.1454 ± 0.016	0.3260 ± 0.036	0.8286 ± 0.018
CS^{3}LR	**11,096** ± 652	0.6687 ± 0.025	**0.6226** ± 0.024	**0.1514** ± 0.012	0.5768 ± 0.036	0.6726 ± 0.019
Paired t-test with $\alpha = 0.05$ (w/t/l)	9/0/0	8/1/0	9/0/0	4/5/0	/	/

The parameters used in the models are listed as follows: KNN (K = 3), SVM with linear kernel (c = 100), SVM with gaussian kernel (c = 100), SVM with polynomial kernel (d = 10), CS-SVM with linear kernel (c = 0.01), CS-SVM with gaussian kernel (c = 0.1), CS-SVM with polynomial kernel (d = 1), class-weighted ratio for cost-sensitive models (γ is [20:1]). For the evaluation metrics, smaller (↓) total cost and larger (↑) AUC, G-mean, F-measure indicate better model performance. On each evaluation metric, the model with best performance is highlighted as bold and the last line shows the win/tie/loss counts of CS^{3}LR versus other label estimation methods. Note that, we do not present the t-test results on sensitivity and specificity as they are a compromise.

Table 4
Comparing label estimation methods based on CSLR with 2% manually labelled data.

	Total cost (↓)	AUC (↑)	G-Mean (↑)	F-measure (↑)	Sensitivity	Specificity
KNN+CSLR	14,064 ± 366	0.7054 ± 0.014	0.2948 ± 0.046	0.1367 ± 0.033	0.0902 ± 0.027	0.9886 ± 0.005
LR+CSLR	13,699 ± 558	0.6754 ± 0.022	0.3435 ± 0.055	0.1578 ± 0.034	0.1241 ± 0.043	0.9777 ± 0.008
SVM (linear)+CSLR	15,251 ± 462	0.6658 ± 0.021	0.0312 ± 0.035	0.0041 ± 0.006	0.0022 ± 0.003	0.9996 ± 0.001
SVM (gaussian)+CSLR	11,383 ± 567	0.6869 ± 0.027	0.5805 ± 0.025	**0.1867** ± 0.020	0.4007 ± 0.034	0.8427 ± 0.017
SVM (polynomial)+CSLR	15,251 ± 46.2	0.6658 ± 0.021	0.0312 ± 0.035	0.0044 ± 0.006	0.0022 ± 0.003	0.9996 ± 0.001
CS-SVM (linear)+CSLR	10,325 ± 433	0.7048 ± 0.014	0.6418 ± 0.017	0.1512 ± 0.009	0.7116 ± 0.004	0.5815 ± 0.048
CS-SVM (gaussian)+CSLR	10,563 ± 650	0.6933 ± 0.025	0.6316 ± 0.028	0.1494 ± 0.016	0.7005 ± 0.060	0.5766 ± 0.084
CS-SVM (polynomial)+CSLR	10,732 ± 583	0.6869 ± 0.023	0.6346 ± 0.019	0.1577 ± 0.012	0.6006 ± 0.043	0.6726 ± 0.039
CSLR+CSLR	11,149 ± 615	0.6630 ± 0.025	0.6161 ± 0.022	0.1587 ± 0.013	0.5283 ± 0.048	0.7213 ± 0.034
CS^{3}LR	**10,144** ± 342	**0.7063** ± 0.014	**0.6553** ± 0.012	0.1711 ± 0.007	0.6153 ± 0.022	0.6984 ± 0.016
Paired t-test with $\alpha = 0.05$ (w/t/l)	9/0/0	7/2/0	9/0/0	7/1/1	/	/

The parameters used in the models are listed as follows: KNN (K = 3), SVM with linear kernel (c = 1), SVM with gaussian kernel (c = 100), SVM with polynomial kernel (d = 1), CS-SVM with linear kernel (c = 0.001), CS-SVM with gaussian kernel (c = 0.1), CS-SVM with polynomial kernel (d = 1), class-weighted ratio for cost-sensitive models (γ is [20:1]). For the evaluation metrics, smaller (↓) total cost and larger (↑) AUC, G-mean, F-measure indicate better model performance. On each evaluation metric, the model with best performance is highlighted as bold and the last line shows the win/tie/loss counts of CS^{3}LR versus other label estimation methods. Note that, we do not present the t-test results on sensitivity and specificity as they are a compromise.

Table 5
Comparing label estimation methods based on CSLR with 5% manually labelled data.

	Total cost (↓)	AUC (↑)	G-Mean (↑)	F-measure (↑)	Sensitivity	Specificity
KNN+CSLR	13,293 ± 302	0.7382 ± 0.010	0.3751 ± 0.027	0.2022 ± 0.021	0.1436 ± 0.022	0.9854 ± 0.003
LR+CSLR	13,443 ± 539	0.7399 ± 0.010	0.3589 ± 0.053	0.1871 ± 0.038	0.1338 ± 0.040	0.9854 ± 0.005
SVM (linear)+CSLR	15,124 ± 100	0.7333 ± 0.011	0.1021 ± 0.031	0.0219 ± 0.012	0.0114 ± 0.006	0.9987 ± 0.001
SVM (gaussian)+CSLR	9842 ± 404	0.7329 ± 0.016	0.6592 ± 0.015	**0.2025** ± 0.014	0.5518 ± 0.027	0.7883 ± 0.022
SVM (polynomial)+CSLR	15,121 ± 90.0	0.7308 ± 0.011	0.0104 ± 0.029	0.0223 ± 0.012	0.0116 ± 0.006	0.9986 ± 0.001
CS-SVM (linear)+CSLR	9428 ± 328	0.7424 ± 0.012	0.6754 ± 0.011	0.1695 ± 0.007	0.7254 ± 0.032	0.6300± 0.029
CS-SVM (gaussian)+CSLR	9647 ± 390	0.7342 ± 0.014	0.6614 ± 0.017	0.1604 ± 0.011	0.7548 ± 0.044	0.5827 ± 0.055
CS-SVM (polynomial)+CSLR	9507 ± 346	0.7378 ± 0.011	0.6753 ± 0.011	0.1730 ± 0.010	0.6957 ± 0.033	0.6565 ± 0.025
CSLR+CSLR	9666 ± 392	0.7309 ± 0.014	0.6710 ± 0.013	0.1730 ± 0.008	0.6734 ± 0.028	0.6693 ± 0.020
CS^{3}LR	**9315** ± 378	**0.7435** ± 0.012	**0.6823** ± 0.014	0.2012 ± 0.009	0.6160 ± 0.026	0.7562 ± 0.012
Paired t-test with $\alpha = 0.05$ (w/t/l)	9/0/0	6/3/0	9/0/0	6/3/0	/	/

The parameters used in the models are listed as follows: KNN (K = 3), SVM with linear kernel (c = 1), SVM with gaussian kernel (c = 100), SVM with polynomial kernel (d = 3), CS-SVM with linear kernel (c = 0.001), CS-SVM with gaussian kernel (c = 0.1), CS-SVM with polynomial kernel (d = 1), class-weighted ratio for cost-sensitive models (γ is [20:1]). For the evaluation metrics, smaller (↓) total cost and larger (↑) AUC, G-mean, F-measure indicate better model performance. On each evaluation metric, the model with best performance is highlighted as bold and the last line shows the win/tie/loss counts of CS^{3}LR versus other label estimation methods. Note that, we do not present the t-test results on sensitivity and specificity as they are a compromise.

CS^{3}LR model in Table 2, better performance of CS^{3}LR model can also be demonstrated, which is due to CS^{3}LR models' process of utilising the hidden label information in the unlabelled training data. We find that the performance of CS^{3}LR with even only 10% labelled historical data is satisfiable when compared with the supervised CSLR that using complete hit-and-run labels (i.e., the CS^{3}LR model with 100% labelled historical data) and improves in comparison with CSLR model.

Note that when higher proportion of labelled data is available, it is also observed that the performance of CS^{3}LR model improves in terms of total cost, AUC, G-mean and F-measure while the performance of LR

Table 6
Comparing label estimation methods based on CSLR with 10% manually labelled data.

	Total cost (↓)	AUC (↑)	G-Mean (↑)	F-measure (↑)	Sensitivity	Specificity
KNN+CSLR	12,396 ± 407	0.7520 ± 0.009	0.4494 ± 0.031	**0.2613** ± 0.023	0.2072 ± 0.030	0.9799 ± 0.004
LR+CSLR	12,976 ± 662	0.7574 ± 0.007	0.3989 ± 0.057	0.2233 ± 0.045	0.1652 ± 0.048	0.9844 ± 0.005
SVM (linear)+CSLR	14,550 ± 308	0.7550 ± 0.008	0.2220 ± 0.046	0.0903 ± 0.033	0.0517 ± 0.021	0.9958 ± 0.001
SVM (gaussian)+CSLR	9198 ± 360	0.7492 ± 0.011	0.6860 ± 0.013	0.2055 ± 0.012	0.6194 ± 0.029	0.7609 ± 0.023
SVM (polynomial)+CSLR	14,550 ± 308	0.7550 ± 0.008	0.2220 ± 0.046	0.0903 ± 0.033	0.0517 ± 0.021	0.9958 ± 0.001
CS-SVM (linear)+CSLR	9060 ± 180	0.7573 ± 0.010	0.6873 ± 0.007	0.1757 ± 0.006	0.7442 ± 0.027	0.6357 ± 0.029
CS-SVM (gaussian)+CSLR	9366 ± 338	0.7529 ± 0.011	0.6688 ± 0.019	0.1629 ± 0.010	0.7789 ± 0.032	0.5766 ± 0.050
CS-SVM (polynomial)+CSLR	9053 ± 205	0.7567 ± 0.001	0.6876 ± 0.008	0.1760 ± 0.006	0.7441 ± 0.027	0.6363 ± 0.029
CSLR+CSLR	9185 ± 233	0.7544 ± 0.009	0.6828 ± 0.009	0.1725 ± 0.006	0.7433 ± 0.002	0.6278 ± 0.025
CS^{3}LR	**8815** ± 279	**0.7601** ± 0.008	**0.6999** ± 0.010	0.2148 ± 0.009	0.6390 ± 0.018	0.7667 ± 0.012
Paired *t*-test with $\alpha = 0.05$ (w/t/l)	9/0/0	8/1/0	9/0/0	7/1/1	/	/

The parameters used in the models are listed as follows: KNN (K = 3), SVM with linear kernel (c = 1), SVM with gaussian kernel (c = 100), SVM with polynomial kernel (d = 1), CS-SVM with linear kernel (c = 0.01), CS-SVM with gaussian kernel (c = 0.1), CS-SVM with polynomial kernel (d = 1), class-weighted ratio for cost-sensitive models (γ is [20:1]). For the evaluation metrics, smaller (↓) total cost and larger (↑) AUC, G-mean, F-measure indicate better model performance. On each evaluation metric, the model with best performance is highlighted as bold and the last line shows the win/tie/loss counts of CS^{3}LR versus other label estimation methods. Note that, we do not present the *t*-test results on sensitivity and specificity as they are a compromise.

model does not improve accordingly. The result can also be explained by the fact that LR model which does not cater for the data imbalance issue and wrongly estimate the labels for the unlabelled data in test set, when the proportion of missing label is large. To sum up, the results in Table 2 validates that the CS^{3}LR model which tackles the class imbalance issue and missing label issue in the crash dataset can improve the performance of hit-and-run crash analysis model.

6.4. The performance of label estimation

To demonstrate the effectiveness and validity of label estimation process in CS^{3}LR for semi-supervised hit-and-run analysis, we adopt nine widely used learners for comparison, including LR model, CSLR model, KNN, SVM (linear/gaussian/polynomial kernel) and cost-sensitive SVM (linear/gaussian/polynomial kernel) (Masnadi-Shirazi et al., 2019). Specifically, we firstly use the nine learners mentioned above to infer the hit-and-run labels of unsupervised training data based on the given supervised training data. Then the CSLR model is trained based on the completely labelled hit-and-run crash dataset to predict the testing data. The testing results under various proportions of labelled data are shown in Tables 3–6, respectively.

Table 7
Top predictors.

Importance	Predictor	w value	Effect
(1) CS^{3}LR with 10% manually labelled data			
1	**No. of pedestrians**	1.3684	Positive
2	**No. of females**	−0.9245	Negative
3	**No. of males**	−0.7907	Negative
4	**No. of passengers**	0.7336	Positive
5	**No. of bicyclists**	0.6753	Positive
6	**Type of urbanised area (rural Victoria)**	−0.6347	Negative
7	Light condition (dark street light on)	0.5577	Positive
8	Collision type (rear end vehicles in same lane)	0.5358	Positive
9	**Run off-road (yes)**	−0.4716	Negative
10	**Collision type (right through)**	−0.4208	Negative
(2) CS^{3}LR with 100% manually labelled data			
1	**No. of pedestrians**	1.3851	Positive
2	**No. of females**	−1.3181	Negative
3	**No. of passengers**	1.1038	Positive
4	**Run off-road (yes)**	−1.0861	Negative
5	**No. of males**	−1.0745	Negative
6	**No. of bicyclists**	0.8627	Positive
7	**Type of urbanised area (rural Victoria)**	−0.7571	Negative
8	**Collision type (right through)**	−0.7261	Negative
9	No. of old drivers	−0.3796	Negative
10	Light condition (day)	−0.3759	Negative

Several findings are discussed as follows: CSLR, CS-SVM (linear), CS-SVM (gaussian) and CS-SVM (polynomial) model perform better in comparison with LR, SVM (linear), SVM (gaussian) and SVM (polynomial) model respectively for the semi-supervised hit-and-run label estimation. The results illustrate the necessity and validity of the application of cost-sensitive learning algorithm on handling hit-and-run crash dataset with imbalance issue.

Moreover, across the four tables, the CS^{3}LR model wins or ties in the majority cases when compared with the other nine models at 5% level of significance, in terms of total cost, AUC value, G-mean and F-measure. The only two cases of loss are achieved on the F-measure with 2% or 10% manually labelled data. This can be explained by the fact that even though the learners (KNN, SVM, CS-SVM, CSLR) are strong learners, each of the algorithm have its own preference in the mechanism for label estimation or prediction. The simple combination of CSLR classifier with other strong learners for label estimation may not be valid as the two steps in semi-supervised learning, i.e., label estimation of unlabelled data followed by supervised classification with CSLR model based on the observed and inferred data labels, are independent with each other. The precomputed label information is kept fixed, which may become suboptimal in the subsequent learning process and hence degrade the classification performance (Wan and Wang, 2019). However, for the proposed CS^{3}LR model, both the label propagation and logistic regression are conducted in a unified cost-sensitive framework defined by Eq. (5). By learning iteratively to minimise the value of loss function until convergence with Algorithm 1, both the two steps can give feedback to each other and thus improve the classification performance.

6.5. Important feature extraction

6.5.1. Comparison of extracted important feature

As discussed in Section 5.2, the calculated importance ranking scores **w** demonstrate the association between the crash-related predictors and hit-and-run decision of drivers. The higher the absolute value of **w**, the more significant the predictor. The sign of **w** indicates the positive and negative effect of each predictor on the likelihood of hit-and-run decision of drivers. Table 7 shows the most significant contributing factors to hit-and-run crashes derived from two models, namely, CS^{3}LR with only 10% manually labelled data and supervised CSLR with complete hit-and-run labels (i.e., CS^{3}LR with 100% labelled historical data).

Comparing the results of part (1) and part (2) in Table 7, eight out of the ten top predictors (as highlighted in bold) generated based on supervised CSLR with complete hit-and-run labels in part (2) are also included part (1), which is generated based on only 10% labelled data with the CS^{3}LR algorithm proposed in this paper. The high degree of consistency in part (1) and part (2) of Table 7 illustrates that the cost-sensitive semi-supervised learning algorithm proposed in this study is effective in identifying the most significant features contributing to hit-and-run crashes even when large proportion of unlabelled data is presented.

6.5.2. Discussion

Different predictors of hit-and-run crashes can be explained by a cost-benefit framework, as drivers make hit-and-run decisions after crashes primarily according to their perception of the probability of apprehension, benefits of running away, and costs of being arrested (Tay et al., 2008). In this section, the effect of the eight significant predictors which appear in both part (1) and part (2) of Table 7 are explained for illustration purpose.

Pedestrians and bicyclists are considered as vulnerable non-motorised road users who are subject to higher injury and fatality risk in comparison with automobile occupants (Chong et al., 2010; Lopez et al., 2018; Jung et al., 2016). From Table 7, it is observed that as the number of pedestrians and bicyclists involved in the crash increases, drivers are more likely to run after the crash. The result is consistent with Zhang et al. (2014), which can be explained by drivers' fear of receiving more serious punishment due to the crash if they are arrested.

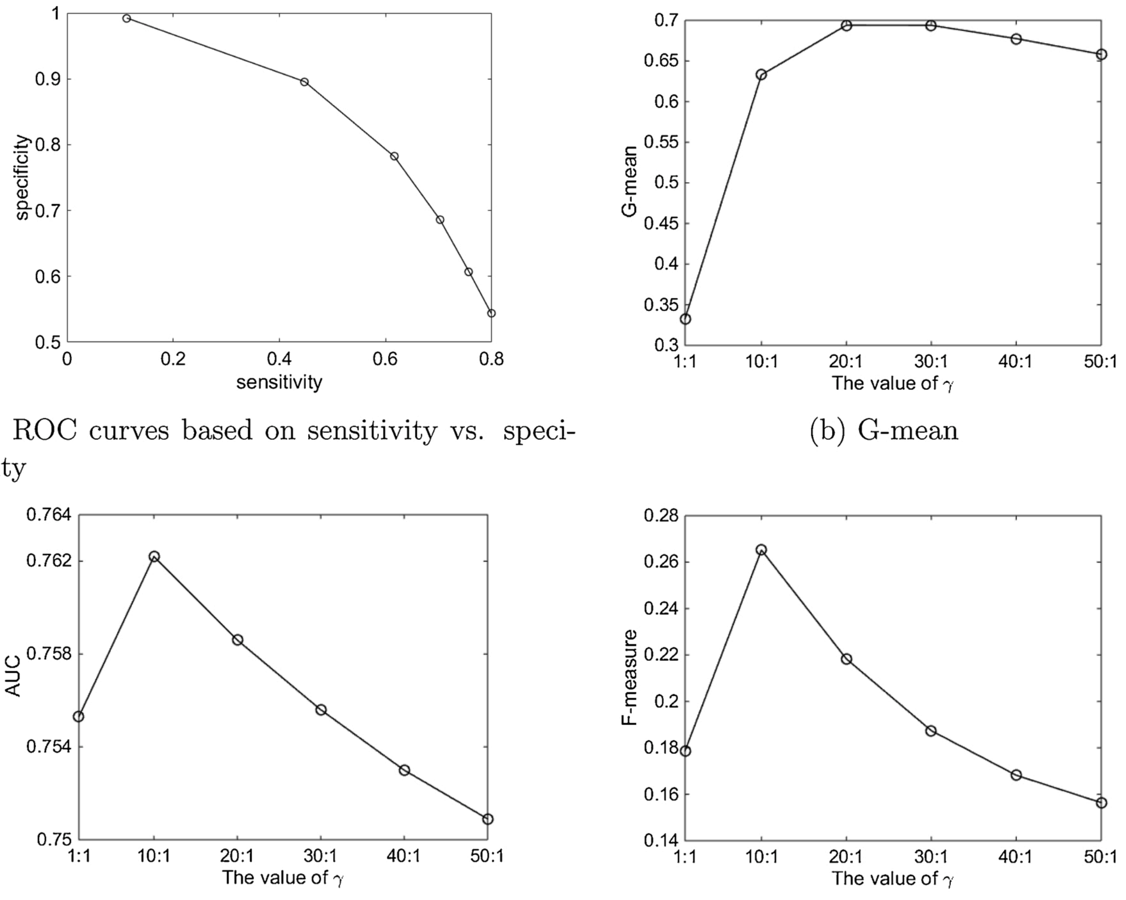
The number of males and females involved in the traffic collision are found to negatively affect the hit-and-run possibility. When the number of males and females involved in the crash increases, drivers are less likely to leave the scene, as the drivers are more likely to infer a higher probability of being witnessed and related to the collision. It is also observed that the increment in the number of females involved in crash have more significant effect on the reduction of hit-and-run likelihood than that of the number of males involved in the crash. The result can be explained by the findings from Solnick and Hemenway (1995), Tay et al. (2008), which suggested that male drivers were 30% more likely to run after the road traffic collision than female drivers. We also find that as the number of vehicle passengers involved in the crash increases, the drivers are more likely to run after crash. The result might be explained by the distraction caused by the passengers (Roshandeh et al., 2016; Zhou et al., 2016), such that drivers' awareness of the collision is reduced.

As for the collision type, the probability of hit-and-run decision of drivers is negatively correlated with right through collision. The result is consistent with Tay et al. (2008), which can be explained by the fact that right turns are mostly undertaken under lower speeds in comparison with driving straight. Thereby, the severity of crash can be reduced and drivers are less likely to flee after crash.

It is also found that, when the crash involves a vehicle running off the road, the driver is less likely to flee the crash scene. The run-off-road (ROR) crashes, which are mainly single vehicle crashes, are considered as one of the most urgent traffic safety problem on rural roads. The share of fatal and severe ROR crashes is higher in comparison with other types of crashes (Wegman et al., 2014). In this case, the vehicle is more likely to be seriously damaged, hence being spotted and apprehended after hit-and-run. It is also spotted that crashed occurred in rural Victoria are less likely to be a hit-and-run crash, in comparison with other types of urbanised areas.

6.6. Parameter sensitivity analysis

In this section, parameter sensitivity analysis is designed to analyse the impact of various class-weighted ratios and hyper-parameter λ



(a) ROC curves based on sensitivity vs. specificity

(b) G-mean

(c) AUC

(d) F-measure

Fig. 5. Influence of class-weighted ratio γ.

values on the performance of modelling hit-and-run decision of drivers with class imbalance and semi-supervised issues. Based on the results from Section 6.3, the proportion of manually labelled historical crash here is set to 10% for sensitivity analysis. Similar to Shi et al. (2019), the metrics of sensitivity and specificity values are applied to evaluate the minority and majority classes, while area under the receiver operating characteristic curve (AUC) value, G-mean, F-measure and total cost are applied as integrated performance evaluation metrics for the imbalanced classification problem. For more robust parameter learning, we adopt the five-fold cross validation performed on the training dataset.

6.6.1. *Class-weighted ratio*

Recall that the c_+ and c_- in Eq. (5) represent the cost of positive class and negative class respectively. It is clear in Eq. (5) that normalising the objective function of CS^{3}LR by c_- do not change the optimisation results. By fixing $c_- = 1$, the class-weighted ratio $\gamma = \frac{c_+}{c_-}$ in fact depicts the importance of positive class, i.e., the hit-and-run cashes.

Fig. 5 shows the effect of γ on sensitivity, specificity, G-mean, AUC and F-measure values respectively with the value of γ chosen from the set $\{1:1,\ 10:1,\ 20:1,\ 30:1,\ 40:1,\ 50:1\}$. In hit-and-run crash prediction, a low specificity value indicates more false alarms, which can relax the vigilance of people, while a low sensitivity value indicates the low accuracy on identifying a real hit-and-run crash case.

From Fig. 5(a), we can observe a clear trade-off between specificity and sensitivity. That is, when the class-weighted ratio γ increases, specificity decreases but sensitivity increases. Thus, it is possible for us to adjust the output of predictor for different requirements of manager by varying the class-weighted ratio γ.

From Fig. 5(b)–(d), it can be seen that the values of G-mean, AUC and F-measure increase first and then decrease. This observation indicates that the proper trade-off between specificity and sensitivity by varying class-weighted ratio γ can improve the prediction performance. However, the performance will degenerate when we excessively sacrifice the specificity to improve the sensitivity, i.e., setting a large γ value for the CS^{3}LR model. Observing Fig. 5(b)–(d), the proper γ can be chosen from the range [10:1, 20:1].

It is worth noting that the variation tendency of G-mean, AUC and F-measure in Figs. 5(b)–(d) are different. This is probably due to their definitions introduced in Section 6.2, where a large G-mean value prefers a balanced prediction results of sensitivity and specificity, while F-measure in class-imbalanced scenario is mainly affected by the number of false positive if the true positive and false negative are not significantly varied. Different from G-mean and F-measure that are calculated under the 0.5 decision threshold of logistic regression, the metric of AUC is determined by the order of prediction probabilities of crash data.

Table 8
Comparison of computational cost (in seconds).

Comparing methods	Training time	Test time
LR only using the few labelled data	0.0302 ± 0.0240	0.0008 ± 0.0004
KNN+CSLR	3.3190 ± 0.0692	0.0008 ± 0.0004
LR+CSLR	1.2943 ± 0.0619	0.0008 ± 0.0005
SVM (linear)+CSLR	7.2694 ± 0.2195	0.0008 ± 0.0004
SVM (gaussian)+CSLR	18.4466 ± 1.2776	0.0008 ± 0.0005
SVM (polynomial)+CSLR	5.1013 ± 0.0841	0.0008 ± 0.0005
CS-SVM (linear)+CSLR	23.5724 ± 1.1011	0.0008 ± 0.0004
CS-SVM (gaussian)+CSLR	25.8266 ± 0.9011	0.0008 ± 0.0005
CS-SVM (polynomial)+CSLR	23.7301 ± 1.1317	0.0008 ± 0.0004
CSLR+CSLR	1.5359 ± 0.0432	0.0008 ± 0.0004
CS^{3}LR	4.5293 ± 0.5330	0.0008 ± 0.0005

6.6.2. *The hyper-parameter* λ

The hyper-parameter $\lambda \in [0,\ 1]$ in Eq. (5) sets the importance of unlabelled data for proposed cost-sensitive semi-supervised maximum likelihood framework. In the extreme case where $\lambda = 0$, we do not consider any unlabelled crash data for hit-and-run analysis. On the other hand, when $\lambda = 1$, the unlabelled data and their estimated labels are treated equally as the labelled ones. Fig. 6 shows the effect of λ on total cost with λ varied from the set $\{0,\ 2^{-7},\ 2^{-6},\ 2^{-5},\ 2^{-4},\ 2^{-3},\ 2^{-2},\ 2^{-1},\ 2^{0}\}$. It can be seen that the total cost is increased when $\lambda = 0$ and $\lambda > 2^{-4}$. Thus, the proper value of λ may be chosen from the range $[2^{-7}, 2^{-3}]$.

6.7. *Computational cost*

In this section, we compare the running time of the proposed CS^{3}LR model with the nine semi-supervised methods reported in Section 6.4 and the LR model only using the few labelled training data. The proportion of manually labelled historical data in training set is set to 10%. All the comparing methods are implemented by Matlab R2016b on a machine with 3.6 GHz CPU and 32 GB RAM. Observing Table 8, we see that the LR model trained based on a few labelled data is more training efficient than the ten semi-supervised methods including the proposed model. However, as we discussed above, both the supervised label information hidden in the large amount of unlabelled data as well as the class imbalance issue are ignored by the LR model. Among the ten semi-supervised methods, label estimation by LR, CSLR, KNN and the proposed CS^{3}LR are more efficient than these SVM and CS-SVM based models in the training process.

Please note that training is usually done offline and the test time is more of a concern in practice. Table 8 reports that the test time of all the comparing methods are nearly the same. This is probably due to the using of the same classification protocol, i.e., logistic regression.

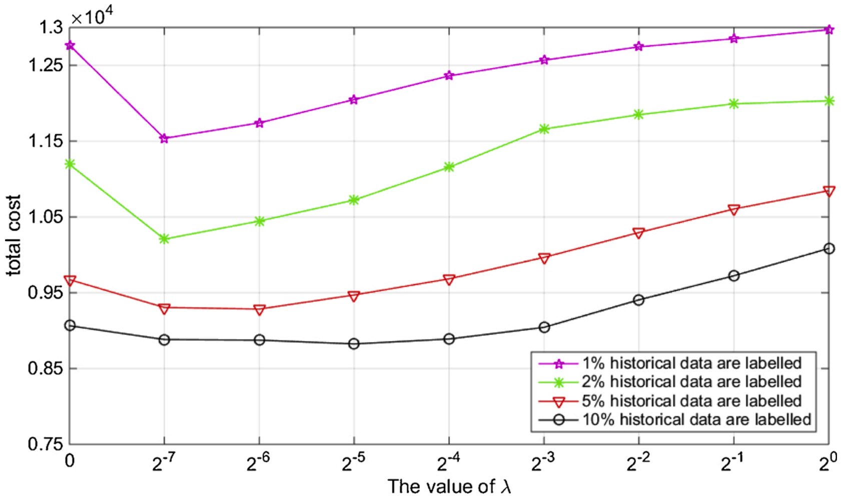


Fig. 6. Influence of hyper-parameter λ on *total cost*.

7. Conclusions

Hit-and-run crashes degrade the morality and result in delays of medical services provided to victims. The imbalance issue of hit-and-run crash dataset exists due to the small proportion of hit-and-run crashes. The crash analysis also suffers from missing label problem due to reasons like data barrier such that the topological information of large amount of unlabelled samples becomes inefficacious. This paper proposed a cost-sensitive learning method for semi-supervised hit-and-run analysis in order to effectively utilise the unlabelled samples and tackle the dataset imbalance problem based on the crash dataset of Victorian, Australia (2013–2019).

To deal with the class imbalance issue, various misclassification costs are adopted for hit-and-run cases and non-hit-and-run cases respectively and the logistic regression model in a form of minimum classification error function is formulated to minimise the overall misclassification loss. Then the label information of unlabelled historical crash data is estimated with cost-sensitive classification maximum likelihood criterion of logistic regression model in an iterative label propagation manner. In each iteration, the label information of both labelled and unlabelled data with pseudo labels is utilised. The experimental results of the proposed model is compared with two logistic regression models and seven machine learning methods, where better performance of the proposed algorithm is demonstrated. Methodologically, the reasons that the performance of the proposed model is satisfiable can be summarised as follows: (1) The cost-sensitive algorithm has been applied to handle the class imbalance issue. (2) The supervised label information hidden in the unsupervised/unlabelled data has been effectively explored and utilised. (3) The label propagation process and cost-sensitive classification algorithm has been unified in a single framework instead of two independent steps.

The result also shows that the most significant features contributing to hit-and-run crashes derived by the cost-sensitive semi-supervised learning algorithm proposed in this paper with large proportion of unlabelled data are highly consistent with the true contributing factors determined by supervised cost-sensitive logistic regression model with complete hit-and-run labels. Sensitivity analysis of class-weighted ratio and hyper-parameter λ on the performance of hit-and-run crash prediction model have also been conducted.

As the cost-sensitive semi-supervised learning framework allows us to determine the most significant contributing factors to hit-and-run crash occurrences, some counter-measures are recommended based on the hit-and-run analysis results to prevent drivers from hit-and-run crashes. For example, CCTV installation is recommended on roads with lower traffic flow. Public awareness and proper education on traffic safety and morality should also be enhanced. Publicity should be targeted especially at hit-and-run crashes involving pedestrians and bicyclists who are vulnerable road users and passengers who may distract the drivers. Proper insurance is also recommended to help driver alleviate the loss from the traffic collisions.

This paper is limited to the information provided by the dataset. Future research may consider more site-specific information, such as the psychological factors of drivers. The cost-sensitive semi-supervised learning framework proposed in this paper can also be applied to analyse crash datasets with other types of missing labels, like crash severity, road users involved in crash, etc. The base learner can also be extended to other types of statistical models according to the needs of individual research.

Author statement

Siying Zhu do the literature review, content planning, result discussion, and manuscript writing.

Jianwu Wan conducted methodology development, result analysis and content discussion.

Conflict of interest

There is no conflict of interest to disclose.

References

Aidoo, E.N., Amoh-Gyimah, R., Ackaah, W., 2013. The effect of road and environmental characteristics on pedestrian hit-and-run accidents in ghana. Accid. Anal. Prev. 53, 23–27.

Alharthi, A., Krotov, V., Bowman, M., 2017. Addressing barriers to big data. Bus. Horiz. 60, 285–292.

Amini, M.R., Gallinari, P., 2002. Semi-supervised logistic regression. In: The 15th European Conference on Artificial Intelligence, 2002. Amsterdam, Netherlands.

Bahrololoom, S., Moridpour, S., Tay, R., Young, W., 2017. Factors affecting hit and run bicycle crashes in Victoria, Australia. In: Australasian Road Safety Conference, 2017. Perth, Western Australia, Australia.

Benson, A., Arnold, L., Tefft, B., Horrey, W.J., 2018. Hit-and-run Crashes: Prevalence, Contributing Factors and Countermeasures.

Borooah, V.K., 2002. Quantitative Applications in the Social Sciences: Logit and Probit. SAGE Publications, Thousand Oaks, CA.

Brzezinski, D., Stefanowski, J., Susmaga, R., Szczech, I., 2019. On the dynamics of classification measures for imbalanced and streaming data. IEEE Trans. Neural Netw. Learn. Syst. 31, 2868–2878.

Chen, T., Shi, X., Wong, Y.D., Yu, X., 2020. Predicting lane-changing risk level based on vehicles' space-series features: a pre-emptive learning approach. Transp. Res. Part C Emerg. Technol. 116, 102646.

Chong, S., Poulos, R., Olivier, J., Watson, W.L., Grzebieta, R., 2010. Relative injury severity among vulnerable non-motorised road users: comparative analysis of injury arising from bicycle-motor vehicle and bicycle-pedestrian collisions. Accid. Anal. Prev. 42, 290–296.

Conradie, P., Choenni, S., 2014. On the barriers for local government releasing open data. Gov. Inf. Q. 31, S10–S17.

Dabiri, S., Marković, N., Heaslip, K., Reddy, C.K., 2020. A deep convolutional neural network based approach for vehicle classification using large-scale gps trajectory data. Transp. Res. Part C Emerg. Technol. 116, 102644.

Das, S., Dutta, A., Kong, X., Sun, X., 2018. Hit and run crashes: knowledge extraction from bicycle involved crashes using first and frugal tree. Int. J. Transp. Sci. Technol.

Dempster, A.P., Laird, N.M., Rubin, D.B., 1977. Maximum likelihood from incomplete data via the em algorithm. J. R. Stat. Soc. Ser. B 39, 1–38.

Fan, R., Chang, K., Hsieh, C., Wang, X., Lin, C., 2008. Liblinear: a library for large linear classification. J. Mach. Learn. Res. 9, 1871–1874.

Fujita, G., Okamura, K., Kihira, M., Kosuge, R., 2014. Factors contributing to driver choice after hitting a pedestrian in Japan. Accid. Anal. Prev. 72, 277–286.

Hastie, T., Tibshirani, R., Friedman, J., 2009. Unsupervised learning. The Elements of Statistical Learning. Springer, pp. 485–585.

Janssen, M., Charalabidis, Y., Zuiderwijk, A., 2012. Benefits, adoption barriers and myths of open data and open government. Inf. Syst. Manag. 29, 258–268.

Jebara, T., Wang, J., Chang, S.F., 2009. Graph construction and b-matching for semi-supervised learning. Proceedings of the 26th Annual International Conference on Machine Learning 441–448.

Jiang, C., Lu, L., Chen, S., Lu, J.J., 2016. Hit-and-run crashes in urban river-crossing road tunnels. Accid. Anal. Prev. 95, 373–380.

Jung, S., Qin, X., Oh, C., 2016. Improving strategic policies for pedestrian safety enhancement using classification tree modeling. Transp. Res. Part A Policy Pract. 85, 53–64.

Ke, J., Zhang, S., Yang, H., Chen, X., 2019. Pca-based missing information imputation for real-time crash likelihood prediction under imbalanced data. Transp. A Transp. Sci. 15, 872–895.

Kim, K., Pant, P., Yamashita, E.Y., 2008. Hit-and-run crashes: use of rough set analysis with logistic regression to capture critical attributes and determinants. Transp. Res. Record 2083, 114–121.

Kuang, L., Yan, H., Zhu, Y., Tu, S., Fan, X., 2019. Predicting duration of traffic accidents based on cost-sensitive bayesian network and weighted k-nearest neighbor. J. Intell. Transp. Syst. 23, 161–174.

Kuhn, M., Johnson, K., 2013. Applied Predictive Modeling, vol. 26. Springer.

Li, H., Parikh, D., He, Q., Qian, B., Li, Z., Fang, D., Hampapur, A., 2014. Improving rail network velocity: a machine learning approach to predictive maintenance. Transp. Res. Part C Emerg. Technol. 45, 17–26.

Li, Z., Jing, X., Wu, F., Zhu, X., Xu, B., Ying, S., 2018. Cost-sensitive transfer kernel canonical correlation analysis for heterogeneous defect prediction. Autom. Softw. Eng. 25, 201–245.

Liu, J., Chen, J., Ye, J., 2009. Large-scale sparse logistic regression. In: ACM SIGKDD International Conference On Knowledge Discovery and Data Mining, 2009. Paris, France.

Liu, J., Khattak, A.J., Chen, C., Wan, D., Ma, J., Hu, J., 2018. Revisiting hit-and-run crashes: a geo-spatial modeling method. Transp. Res. Record 2672, 81–92.

Liu, T., Yang, Y., Huang, G.B., Yeo, Y.K., Lin, Z., 2015. Driver distraction detection using semi-supervised machine learning. IEEE Trans. Intell. Transp. Syst. 17, 1108–1120.

Lomax, S., Vadera, S., 2013. A survey of cost-sensitive decision tree induction algorithms. ACM Comput. Surv. 45, 1–35.

Lopez, D., Glickman, M.E., Soumerai, S.B., Hemenway, D., 2018. Identifying factors related to a hit-and-run after a vehicle-bicycle collision. J. Transp. Health 8, 299–306.

MacLeod, K.E., Griswold, J.B., Arnold, L.S., Ragland, D.R., 2012. Factors associated with hit-and-run pedestrian fatalities and driver identification. Accid. Anal. Prev. 45, 366–372.

Masnadi-Shirazi, H., Vasconcelos, N., Iranmehr, A., 2019. Cost-sensitive support vector machines. Neurocmputing 343, 50–64.

Mohammadi, R., He, Q., Ghofrani, F., Pathak, A., Aref, A., 2019. Exploring the impact of foot-by-foot track geometry on the occurrence of rail defects. Transp. Res. Part C Emerg. Technol. 102, 153–172.

Parsa, A.B., Taghipour, H., Derrible, S., Mohammadian, A.K., 2019. Real-time accident detection: coping with imbalanced data. Accid. Anal. Prev. 129, 202–210.

Rice, J.A., 2006. Mathematical Statistics and Data Analysis. Nelson Education.

Roshandeh, A.M., Zhou, B., Behnood, A., 2016. Comparison of contributing factors in hit-and-run crashes with distracted and non-distracted drivers. Transp. Res. Part F Traffic Psychol. Behav. 38, 22–28.

Seattle Department of Transportation, 2018. Collisions. . [Online]; (accessed 09.05.20). https://www.seattle.gov/Documents/Departments/SDOT/GIS/Collisions_OD.pdf.

Seiffert, C., Khoshgoftaar, T.M., Van Hulse, J., Napolitano, A., 2008. A comparative study of data sampling and cost sensitive learning. In: 2008 IEEE International Conference on Data Mining Workshops. IEEE, pp. 46–52.

Shi, X., Wong, Y.D., Li, M.Z.F., Palanisamy, C., Chai, C., 2019. A feature learning approach based on xgboost for driving assessment and risk prediction. Accid. Anal. Prev. 129, 170–179.

Sivasankaran, S., Balasubramanian, V., 2018a. Investigating factors associated with hit-and-run crashes in Indian metropolitan city using association rules. In: Australasian Road Safety Conference, 2018. Sydney, New South Wales, Australia.

Sivasankaran, S.K., Balasubramanian, V., 2018b. Data mining based analysis of hit-and-run crashes in metropolitan city. Congress of the International Ergonomics Association 113–122.

Solnick, S.J., Hemenway, D., 1995. The hit-and-run in fatal pedestrian accidents: victims, circumstances and drivers. Accid. Anal. Prev. 27, 643–649.

Symons, M.J., 1981. Clustering criteria and multivariate normal mixtures. Biometrics 37, 35–43.

Tan, B., Zhang, J., Wang, L., 2011. Semi-supervised elastic net for pedestrian counting. Pattern Recognit. 44, 2297–2304.

Tanha, J., van Someren, M., Afsarmanesh, H., 2017. Semi-supervised self-training for decision tree classifiers. Int. J. Mach. Learn. Cybern. 8, 355–370.

Tay, R., Kattan, L., Sun, H., 2010. Logistic model of hit and run crashes in calgary. Can. J. Transp. 4.

Tay, R., Rifaat, S.M., Chin, H.C., 2008. A logistic model of the effects of roadway, environmental, vehicle, crash and driver characteristics on hit-and-run crashes. Accid. Anal. Prev. 40, 1330–1336.

Toronto Police Service, 2019. KSI. . [Online]; (accessed 09.05.20). http://data.torontopolice.on.ca/datasets/ksi.

VicRoads, 2019. Crashes Last Five Years. . [Online]; (accessed 22.04.19). https://vicroadsopendata-vicroadsmaps.opendata.arcgis.com/datasets/crashes-last-five-years.

Wan, J., Wang, Y., 2019. Cost-sensitive label propagation for semi-supervised face recognition. IEEE Trans. Inf. Forensics Secur. 14, 1729–1743.

Wang, J., Zhu, S., Gong, Y., 2010. Driving safety monitoring using semisupervised learning on time series data. IEEE Trans. Intell. Transp. Syst. 11, 728–737.

Wegman, F., et al., 2014. Analyzing road design risk factors for run-off-road crashes in the Netherlands with crash prediction models. J. Saf. Res. 49, 121–131.

Weiss, G.M., 2013. Foundations of Imbalanced Learning. Imbalanced Learning: Foundations, Algorithms, and Applications, pp. 13–41.

Weiss, G.M., McCarthy, K., Zabar, B., 2007. Cost-sensitive learning vs. sampling: which is best for handling unbalanced classes with unequal error costs? DMIN 7, 24.

Xie, M., Cheng, W., Gill, G.S., Falahati, R., Jia, X., Choi, S., 2017. Predicting Likelihood of Hit-and-Run Crashes Using Real-Time Loop Detector Data and Hierarchical Bayesian Binary Logit Model With Random Effects. Technical Report.

Xie, M., Cheng, W., Gill, G.S., Zhou, J., Jia, X., Choi, S., 2018. Investigation of hit-and-run crash occurrence and severity using real-time loop detector data and hierarchical Bayesian binary logit model with random effects. Traffic Inj. Prev. 19, 207–213.

Zhang, G., Li, G., Cai, T., Bishai, D.M., Wu, C., Chan, Z., 2014. Factors contributing to hit-and-run crashes in China. Transp. Res. Part F Traffic Psychol. Behav. 23, 113–124.

Zhang, Y., Zhou, Z., 2010. Cost-sensitive face recognition. IEEE Trans. Pattern Anal. Mach. Intell. 32, 1758–1769.

Zhou, B., Li, Z., Zhang, S., Zhang, X., Liu, X., Ma, Q., 2019. Analysis of factors affecting hit-and-run and non-hit-and-run in vehicle-bicycle crashes: a non-parametric approach incorporating data imbalance treatment. Sustainability 11, 1327.

Zhou, B., Roshandeh, A.M., Zhang, S., Ma, Z., 2016. Analysis of factors contributing to hit-and-run crashes involved with improper driving behaviors. Proc. Eng. 137, 554–562.

Zhu, S., 2020. Investigation of vehicle-bicycle hit-and-run crashes. Traffic Inj. Prev. 21, 506–511.

Zhu, W., Ash, J., Li, Z., Wang, Y., Lowry, M., 2015. Applying semi-supervised learning method for cellphone-based travel mode classification. In: 2015 IEEE First International Smart Cities Conference (ISC2). IEEE, pp. 1–6.

Zhu, X.J., 2005. Semi-Supervised Learning Literature Survey. University of Wisconsin-Madison Department of Computer Sciences. Technical Report.