Accident Analysis and Prevention 120 (2018) 250–261

Contents lists available at ScienceDirect

Accident Analysis and Prevention

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

Classification of motor vehicle crash injury severity: A hybrid approach for imbalanced data

Heejin Jeong[a,b,*], Youngchan Jang[b], Patrick J. Bowman[a], Neda Masoud[c]

[a] University of Michigan Transportation Research Institute, 2901 Baxter Road, Ann Arbor, MI, 48109, USA
[b] Department of Industrial and Operations Engineering, University of Michigan, 1205 Beal Avenue, Ann Arbor, MI 48109, USA
[c] Department of Civil and Environmental Engineering, University of Michigan, 2350 Hayward Street, Ann Arbor, MI 48109, USA

ARTICLE INFO

Keywords:
Vehicle crashes
Injury severity classification
Imbalanced data
Machine learning
Data analytics
Automated vehicle safety

ABSTRACT

This study aims to classify the injury severity in motor-vehicle crashes with both high accuracy and sensitivity rates. The dataset used in this study contains 297,113 vehicle crashes, obtained from the Michigan Traffic Crash Facts (MTCF) dataset, from 2016–2017. Similar to any other crash dataset, different accident severity classes are not equally represented in MTCF. To account for the imbalanced classes, several techniques have been used, including under-sampling and over-sampling. Using five classification learning models (i.e., Logistic regression, Decision tree, Neural network, Gradient boosting model, and Naïve Bayes classifier), we classify the levels of injury severity and attempt to improve the classification performance by two training-testing methods including Bootstrap aggregation (or bagging) and majority voting. Furthermore, due to the imbalance present in the dataset, we use the geometric mean (G-mean) to evaluate the classification performance. We show that the classification performance is the highest when bagging is used with decision trees, with over-sampling treatment for imbalanced data. The effect of treatments for the imbalanced data is maximized when under-sampling is combined with bagging. In addition to the original five classes of injury severity in the MTCF dataset, we consider two additional classification problems, one with two classes and the other with three classes, to (1) investigate the impact of the number of classes on the performance of classification models, and (2) enable comparing our results with the literature.

1. Introduction

According to the National Highway Traffic Safety Administration Research Note, motor-vehicle crashes in 2015 led to 35,092 fatalities in the United States, an increase of 7.2% from 2014 (National Center for Statistics and Analysis, 2015). A crash analysis report published in the U.S. Department of Transportation indicates that the total economic cost of motor vehicle crashes occurred in 2010 was more than $200 billion, including the costs due to approximately 33,000 fatalities, 3.9 million non-fatal injuries, and 24 million damaged vehicles (Blincoe et al., 2015). About 31 percent of the total economic costs were in form of property damage costs, while 10 percent of the total costs were in medical costs. Based on these statistics, a better understanding of the relationship between crash risk factors and the injury severity can help enhance driving safety, curb the economic impact of crashes, and reduce the number of fatal crashes.

In classifying crash injury severity, the classification power of a model cannot be simply captured by its correct classification rate. Accident severity datasets are typically imbalanced, with the non-fatal class containing disproportionally more data points compared to the fatal class. If untreated, such a structure could lead to training models that look promising on the outside with high accuracy rates (the accuracy is defined as the ability of the model to correctly predict accident severity classes on a test set; see Eq. (1)), but fail to be informative in reality. An extreme example of a weak model is a trivial model that predicts all accidents to be non-fatal, in a 2-class problem with fatal and non-fatal classes. Such a model would have a very high accuracy rate, while the value of a crash classification model lies mostly on correct classifications of higher severity classes (e.g., fatal crashes), typically referred to as “sensitivity” (i.e., the ability of the model to correctly classify the severity level as ‘fatal’ (Farchi et al., 2007; Parikh et al., 2008; see Eq. (2)). On the other hand, a model that classifies all accidents as fatal would produce a high sensitivity, but a low accuracy score. Hence, there is a clear trade-off between accuracy and sensitivity scores of crash severity models that can only be resolved through appropriate handling of imbalanced data. This imbalanced data structure,

* Corresponding author.
E-mail address: heejinj@umich.edu (H. Jeong).

https://doi.org/10.1016/j.aap.2018.08.025
Received 14 December 2017; Received in revised form 16 August 2018; Accepted 26 August 2018
Available online 30 August 2018
0001-4575/

Table 1
Confusion Matrix and the Four Measurements for 2-Class Classification.

	Predicted Positive (fatal)	Predicted Negative (non-fatal)
Actual Positive (fatal)	True Positive (TP)	False Negative (FN)
Actual Negative (non-fatal)	False Positive (FP)	True Negative (TN)

therefore, necessitates additional steps in model training and evaluation: (*i*) using appropriate evaluation metrics, and (*ii*) balancing the dataset before training.

Limitations of classification accuracy rate in evaluating model performance could be addressed through using additional statistical measures, namely, true positive, true negative, false positive, and false negative (see Table 1 for a detailed description) to create more informative metrics. Using these measurements, accuracy, sensitivity, and specificity, as defined in Eqs. (1)–(3), respectively, can be easily computed for a 2-class classification problem. These metrics collectively help depict a more comprehensive picture of the overall model performance (Parikh et al., 2008). Ultimately, geometric mean (or G-mean) of sensitivity and specificity can be used as a compact evaluation metric to compare the general performance of different models. The G-mean is calculated as the square root of the product of sensitivity and specificity and will have high values when both sensitivity and specificity are high and the difference between the two metrics is small (Kubat et al., 1997). Finally, while reporting a variety of metrics that can provide a comprehensive picture of model performance is necessary, measures need to be taken to produce high-performance models in the first place. This can be obtained by generating a balanced dataset based on the original imbalanced dataset on which models can be trained.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FN + FP} \tag{1}$$

$$\text{Sensitivity} = \frac{TP}{TP + FN} \tag{2}$$

$$\text{Specificity} = \frac{TN}{FP + TN} \tag{3}$$

Using a different terminology, the G-mean in binary classification can be the square root of the product of class 1’s accuracy (former sensitivity measure) and class 2’s accuracy (former specificity measure). As such, in a *C*-class classification problem the definition of G-mean can be expanded to *C* classes as the following:

$$\text{G-mean} = (Class\ 1\ accuracy \times Class\ 2\ accuracy \times \cdots \times Class\ C\ accuracy)^{1/C}$$

Traditionally, motor vehicle crash injury severity has been modeled using statistical methods, with the goal of identifying the significance of each potential factor in the severity of the outcome of a crash. This type of analysis is valuable in informing future safety-focused planning efforts. Although the proposed methodology here can be used for the same purposes through conducting sensitivity analysis, our main goal is to learn an ensemble of models that can predict the severity of crashes in a fraction of a second and with a high degree of accuracy. Such a model is needed by autonomous vehicles, where fast evaluation of the circumstances and decision making is critical. An autonomous vehicle, having access to such prediction, can take the necessary cautionary measures to avoid (or mitigate the impact of) any potential accidents.

Although autonomous vehicles are equipped with a variety of sensors to help them browse through the surrounding environment, they are highly susceptible to low-quality sensor readings as well as anomalous sensor readings, either due to faulty sensors or malicious cyber-attacks. As such, redundancy in information plays a great role in increasing the reliability of autonomous vehicles under various scenarios, and in the absence of (reliable) sensor readings. With focus on factors that are easy to measure (e.g., whether the driver is under the influence of alcohol using in-vehicle cameras), and can be reliably obtained from other sources (e.g., the roadway alignment using high definition maps, the weather conditions), we provide a prior model on the severity of potential accidents, should they occur. This information can help the autonomous system make better choices, e.g., take over the control of the motor vehicle from the human driver in case the driver is perceived to be under the influence (in semi-autonomous vehicles), or drive at lower speeds in the presence of adverse weather conditions, such as fog, which may reduce the precision of sensors.

The contributions of this paper to the literature are three-fold. First, we use a total of 5 machine learning techniques to model crash injury severity levels. These models are trained in isolation, and as ensemble models, providing insights on the degree to which various machine learning techniques are appropriate for crash injury severity classification. Second, we use under-sampling and over-sampling to treat the inherent imbalance of the crash dataset before learning and discuss the effects of these treatments. Finally, we provide various performance statistics and show that several of our models out-perform models in the literature by achieving both high accuracy and sensitivity rates at the same time.

2. Literature review

To date, many previous studies have modeled the traffic crash injury severity with potential risk factors, using statistical and machine learning methods (e.g., Abdelwahab and Abdel-Aty, 2001; Chang and Wang, 2006; Eluru et al., 2008; Zhu and Srinivasan, 2011; Castro et al., 2013; Xu et al., 2013; Yu and Abdel-Aty, 2013; Lee and Li, 2015; Chen et al., 2015, 2016). For example, Abdelwahab and Abdel-Aty (2001) used two neural network models (namely, multilayer perceptron and fuzzy adaptive resonance theory) to classify driver injury severity with driver, vehicle, roadway, and environmental factors. Chang and Wang (2006) estimated the effect of several risk factors (e.g., driver/vehicle, highway/environmental variables) on injury severity (i.e., fatal, injury, and no-injury) using classification and regression tree (CART). They analyzed crash data from police records collected in Taiwan and found that vehicle type is the most important factor associated with injury severity. Chen et al. (2015) used a hybrid method that combines multinomial logit and Bayesian network to classify the driver injury severity, with crash data from New Mexico. They identified several risk factors for motor vehicle crash fatalities, including environmental factors such as windy weather and inferior lighting conditions.

Among the studies on classifying motor vehicle crash injury severity in the literature (e.g., Abdelwahab and Abdel-Aty, 2001; Chang and Wang, 2006; Eluru et al., 2008; Castro et al., 2013; Chang and Chien, 2013; Xu et al., 2013; Yu and Abdel-Aty, 2013; Lee and Li, 2015; Chen et al., 2016), some report only the classification accuracy (e.g., Abdelwahab and Abdel-Aty, 2001). Other studies report both the classification accuracy and the classification accuracy of the class of interest by reporting statistics such as sensitivity or specificity, but only the classification accuracy has been the focus of discussion (e.g., Chang and Chien, 2013; Chang and Wang, 2006). Table 2 summarizes studies in the literature that report more statistics than just the general classification accuracy. Moreover, none of these studies have taken measures to address data imbalance, although two studies (i.e., Chang and Wang, 2006; Chen et al., 2015) have pointed out this issue as a limitation of their work.

3. Data description

Crash data used for fatality analysis were obtained from the Michigan Traffic Crash Facts (MTCF) database that contains official Michigan year-end crash data (Office of Highway Safety Planning, 2017). In this study, two years of crash data (restricted to vehicle crashes only; neither pedestrian nor bicycle) were collected (from 2016 to

Table 2
Summary of Classification Performance in Previous Studies.

Studies	● Data (year) ● Number of crash data ● (training: testing data) ● Number of variables	● Number of classes ● Class break-down (%; from Class 1 to 5)	Algorithms to classify crash injury severity	Classification results on the test set (ordered from most to least severe)						
				Overall accuracy (%)	Class 1	Class2	Class 3	Class 4	Class 5	G-mean (%)
[1] Chang and Wang (2006)	● Crash data in Taipei, Taiwan (2001) ● N = 26,831 (16,960:9,871) ● 20	● 3 ● Fatality (0.4) / Injury (59.9) / No-injury (39.7)	Classification and regression tree (CART)	91.7	0	96.4	85.1	N/A	N/A	0
[2] Li et al. (2012)	● Freeway crash data in Florida (2004-2006) ● N = 5,538 (4:1) ● 18	● 5 ● Fatal (0.9) / Incapacitating (5.1) / Non-incapacitating (15.1) / Possible (or invisible, 26.4) / No-injury (52.5)	Support vector machine (SVM)	48.8	1.7	2.3	10.1	25.4	77.0	9.5
		● 2 ● Fatal (or injury, 47.5) / No-injury (52.5)	Support vector machine (SVM)	57.6	53.6	61.3	N/A	N/A	N/A	57.3
		● 5 ● Fatal (0.9) / Incapacitating (5.1) / Non-incapacitating (15.1) / Possible (or invisible, 26.4) / No-injury (52.5)	Ordered probit (OP)	44.0	0	0	0.1	52.6	56.8	0
[3] Xie et al. (2012)	● Florida Traffic Crash Records Database (2005) ● N = 4,285 (3,000:1,285) ● 53	● 5 ● Fatal / Incapacitating / Non-incapacitating / Possible / No-injury (unknown)	Multinomial logit (MNL)	43.7	30.5	20.2	44.8	11.2	74.9	29.7
			Latent class logit (LCL)	63.1	44.5	30.5	64.4	48.4	86.2	51.6
[4] Chang and Chien (2013)	● Truck accidents in Taiwan (2005-2006) ● N = 1,620 (1,134:486) ● 21	● 3 ● Fatal (4.8) / Injury (35.0) / No-injury (60.2)	Non-parametric classification and regression tree (CART)	67.7	47.8	42.6	84.7	N/A	N/A	55.7
[5] Zhao and Khattak (2015)	● Highway–rail grade crossing crash data and the national highway (2009-2013) ● N = 6874 (5,641:1,233) ● 10	● 3 ● Fatal (8.2) / Injury (29.0) / No-injury (62.8)	Ordered probit (OP)	65.0	2.2	42.0	84.3	N/A	N/A	19.7
			Multinomial logit (MNL)	57.4	22.8	36.8	72.0	N/A	N/A	39.3
			Random parameter logit (RPL)	57.5	23.7	36.8	72.1	N/A	N/A	39.8
[6] Chen et al. (2015)	● Motor vehicle crashes in New Mexico (2010-2011) ● N = 23,433 (11,486:11,947) ● 27	● 3 ● Fatal (8.2) / Injury (29.0) / No-injury (62.8)	MNL & Bayesian network	65.8	27.3	33.2	85.2	N/A	N/A	42.6

Note: Studies are ordered by publication date.

Table 3
Variable Summary.

Category	No.	Features	Definitions/Details	Frequency	
				Training	Testing
Driver factor	1	Alcohol	Whether the crash involved alcohol (at least one individual)		
			1 = alcohol	7,867	3,293
			0 = non-alcohol	200,133	85,840
	2	Drug	Whether the crash involved drugs (at least one individual)		
			1 = drug	2,161	914
			0= non-drug	205,819	88,219
	3	Young Age	Whether the crash involved one driver aged 24 or less		
			1 = young driver (< 24 y.o.)	61,436	26,268
			0 = non- young driver	146,544	62,865
	4	Old Age	Whether the crash involved one driver aged 64 or more		
			1 = old driver (≥64 y.o.)	44,903	19,479
			0 = non-old driver	163,077	69,654
	5	Distracted	Whether the driver distraction was involved		
			1 = distracted	9,522	4,031
			0 = non-distracted	198,458	85,102
Environmental factor	6	Curve	Whether the crash involved curved road alignment		
			1 = curved	19,436	6,616
			0 = non-curved	188,544	82,517
	(Weather)		The prevailing atmospheric conditions that existed at the time of the crash		
	7	Cloudy	1 = cloudy	43,749	18,706
			0 = non-cloudy	164,231	70,427
	8	Rain	1 = rain	17,313	7,408
			0 = non-rain	190,667	81,725
	9	Snow	1 = snow	19,436	8,365
			0 = non-snow	188,544	80,768
	10	Fog/smoke	1 = fog/smoke	2,087	885
			0 = non-fog/smoke	205,893	88,248
	11	Severe wind	1 = severe wind	737	357
			0 = non-severe wind	207,243	88,776
	12	Sleet/hail	1 = sleet/hail	783	365
			0 = non-sleet/hail	207,197	88,768
	(Traffic control device)		The traffic control device present at the site of the crash		
	13	Signal device	1 = signal device	20,745	8,905
			0 = non-signal device	187,235	80,228
	14	Stop sign	1 = stop sign	5,272	2,281
			0 = non-stop sign	202,708	86,852
	15	Yield sign	1 = yield sign	1,094	477
			0 = non-yield sign	206,886	88,656
	(Other vehicles involvement)		Whether the crash involved at least one following vehicle		
	16	Commercial vehicle (or bus)	1 = commercial vehicle	5,886	2,568
			0 = non-commercial vehicle	202,094	86,565
	17	Emergency vehicle	1 = emergency vehicle	1,690	691
			0 = non- emergency vehicle	206,290	88,442
	18	Off road vehicle	1 = off-road vehicle	302	98
			0 = non-off-road vehicle	207,678	89,035
	19	Snowmobile	1 = snowmobile	99	41
			0 = non-snowmobile	207,881	89,092
	20	Pedestrian	1 = pedestrian	1,3222	575
			0 = non-pedestrian	206,658	88,558
	21	Bicyclist	1 = bicyclist	695	270
			0 = non-bicyclist	207,375	88,863
	22	Farm vehicle	1 = farm vehicle	208	92
			0 = non-farm vehicle	207,772	89,041
	23	Motorcycle	1 = motorcycle	2,115	886
			0 = non-motorcycle	205,865	88,247
	24	Train	1 = train	8	7
			0 = non-train	207,972	89,126
Temporal factor	(Season)		The season in which the crash occurred		
	25	Spring	1 = spring (March, April, May)	44,683	19,030
			0 = non-spring	163,297	70,103
	26	Summer	1 = summer (June, July, August)	44,010	18,819
			0 = non-summer	163,970	70,314
	27	Fall	1 = fall (September, October, November)	59,951	25,505
			0 = non-fall	148,029	63,628
	(Day of week)		The day of the week on which the crash occurred		
	28	Weekday	1 = weekday (Monday, Tuesday, Wednesday, Thursday, Friday)	159,213	67,973
			0 = weekend (Saturday, Sunday)	48,767	21,160
	(Time of day)		The hour at which the crash occurred		
	29	Peak	1 = peak (6–9 am, 4–7 pm)	78,546	33,553
			0 = non-peak	129,434	55,580
	30	Night	1 = night (8 pm–5 am)	49,061	20,949
			0 = non-night	158,919	68,184

Table 4
Injury severity levels of crashes represented in dataset.

Injury severity level		Train	Test	Total
Level 1	Fatal injury	715 (0.34%)	306 (0.34%)	1,021 (0.34%)
Level 2	Incapacitating injury	3,076 (1.48%)	1317 (1.48%)	4,393 (1.48%)
Level 3	Non-incapacitating injury	9,427 (4.53%)	4039 (4.53%)	13,466 (4.53%)
Level 4	Possible injury	18,947 (9.11)	8,119 (9.11%)	27,066 (9.11%)
Level 5	No injury	175,815 (84.53%)	75,352 (84.54%)	251,167 (84.54%)
Overall		207,980	89,133	297,113

2017). Initially, three categories of contributing factors including driver, environmental, and temporal factors were extracted from the database (i.e., alcohol, drug, age, distraction, area of road at crash, weather condition, traffic control device, other vehicle involvement, crash month, day of week, and time of day). These factors were then converted into 30 binary attributes. Table 3 provides detailed information on the 30 attributes. Records that included missing values (52.6% of the original dataset), such as the “unknown” and “not entered” categories were not included in the final dataset.

A total of 297,113 crashes (including 207,980 – 70% for training and 89,133–30% for testing, selected randomly) were used in this study. The original dataset provided five levels of crash severity based on the most severe injury level of any driver/passengers involved in the crash. Table 4 shows the description of injury severity of crashes. Level 1 represents fatal injury crashes which account for the smallest portion (0.34%). Levels 2 and 3 indicate incapacitating and non-incapacitating injury crashes, respectively, which account for 1.48% and 4.53% of all crashes. Level 4 represents possible injury crashes which account for 9.11% of all crashes. Level 5 indicates no-injury crashes with the largest portion of 84.54%. In addition to the original five-class classification, we train two additional classification models, one with 3 classes, and one with two classes (binary classification). Clustering the original 5-class response variable to two and three classes enables us to (*i*) assess whether such clustering allows us to output better predictions and especially obtain better sensitivity results for the more severe classes, and (*ii*) compare our study with the literature, as different studies in the literature are conducted with different numbers of classes. For 3-class classification, the original five classes were categorized into three classes, with the first class containing levels 1 and 2, the second class containing levels 3 and 4, and the third class containing level 5. For the binary classification, we used levels 1 and 2 as the first class and levels 3,4, and 5 as the second class.

4. Methodology

In this study, five classification models were used for training the data. This section provides a summary of these models, elaborates on algorithms used for training, and discusses parameter values for the final models, where applicable. In the rest of this section, we introduce several methods used to treat the imbalanced structure of the dataset and investigate whether we can improve the predictive power of the five models using bootstrap aggregation. The programming language R has been used in training the models in the following sections. Under each model, we specify the package that has been used.

4.1. Classification models

4.1.1. Logistic regression

The logistic regression is a special case of the generalized linear model (GLM), which generalizes the ordinary linear regression by allowing the linear model to be related with a response variable that follows the exponential family via an appropriate link function. When the response variable is binomially distributed with parameter p_i, the

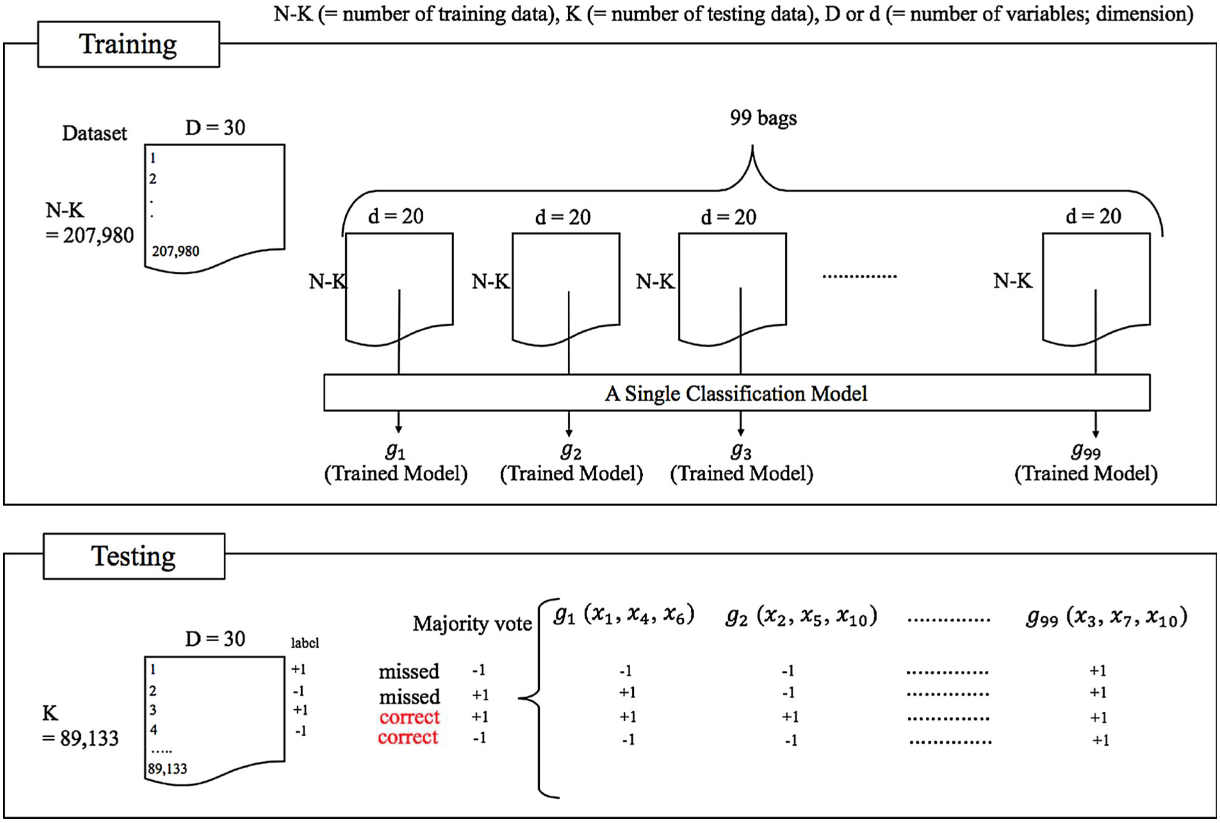

Fig. 1. Bagging process.

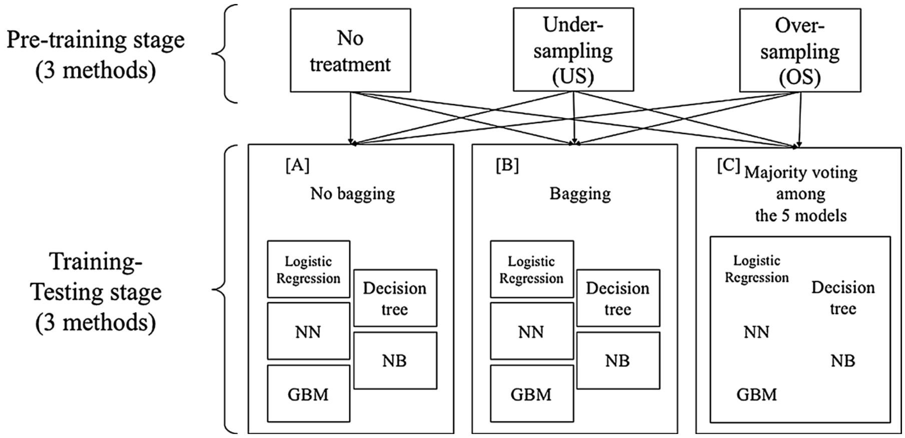

Fig. 2. Learning process.

Table 5a
5-class Classification Performance (%) and computation time (s) by Treatments and Methods [A] and [B].

		Method [A]: No bagging			Method [B]: Bagging		
Treatments[1]		No	US	OS	No	US	OS
Multinomial logistic regression	Overall accuracy	84.65	1.44	0.35	84.66	0.35	0.35
	Class 1	3.55	0	100	3.22	100	100
	Class 2	4.43	100	0	2.72	0	0
	Class 3	10.55	0	0	10.37	0	0
	Class 4	1.16	0	0	1.02	0	0
	Class 5	99.55	0	0	99.60	0	0
	G-mean	7.19	0	0	6.21	0	0
	Time (s)	*23.4*	*1.33*	*39.17*	*2,127.63*	*352.12*	*5,694.35*
Decision tree	Overall accuracy	84.37	64.44	82.10	84.69	52.83	56.63
	Class 1	0	60.00	60.00	5.48	54.84	40.00
	Class 2	0	14.54	12.36	5.83	23.25	28.93
	Class 3	0	7.13	0	9.24	12.03	11.95
	Class 4	0	21.82	0	1.11	36.30	40.22
	Class 5	100	73.09	96.85	99.63	57.35	61.39
	G-mean	0	25.08	0	7.99	31.68	[2]**32.11**
	Time (s)	*3.37*	*0.25*	*134.70*	*79.43*	*3.48*	*288.42*
GBM	Overall accuracy	84.71	56.17	57.83	84.57	54.19	55.04
	Class 1	7.74	50.65	44.52	0	52.90	40.96
	Class 2	6.07	23.87	27.29	3.42	24.41	28.53
	Class 3	9.86	9.51	8.75	7.24	9.14	9.76
	Class 4	1.21	36.99	39.73	0.35	41.38	41.90
	Class 5	99.6	61.38	63.04	99.74	58.55	59.44
	G-mean	8.90	30.43	30.55	0	30.99	30.95
	Time (s)	*52.29*	*6.55*	*193.02*	*3,891.42*	*3,311.83*	*67,438.66*
NN	Overall accuracy	84.65	0.35	83.88	9.41	0.35	0.946
	Class 1	100	100	28.39	4.83	100	0
	Class 2	1.48	0	6.53	18.74	0	6.68
	Class 3	12.69	0	1.53	5.42	0	8.65
	Class 4	0.57	0	0.28	95.49	0	96.48
	Class 5	99.56	0	99.07	0	0	0
	G-mean	10.13	0	6.01	0	0	0
	Time (s)	*14.97*	*1.46*	*16.01*	*2,127.63*	*863.32*	*4,444.24*
NB	Overall accuracy	84.54	57.86	57.19	84.66	67.11	66.58
	Class 1	11.29	54.19	54.84	5.48	51.61	53.87
	Class 2	8.71	19.67	22.47	5.36	12.44	12.90
	Class 3	8.70	9.34	8.97	8.87	3.54	2.76
	Class 4	1.46	34.26	36.17	1.07	25.07	25.95
	Class 5	99.38	63.74	62.71	99.63	76.17	75.46
	G-mean	10.44	29.34	30.19	7.74	21.26	20.65
	Time (s)	*1.41*	*0.30*	*3.76*	*793.33*	*248.31*	*3,135.09*
Mean (SD) of G-means		7.3 (4.3)	17.0 (15.6)	13.4 (15.7)	4.4 (4.1)	16.8 (15.9)	16.7 (15.9)
Increase (%) of G-means from no treatment		N/A	131.5	82.1	N/A	282.5	281.5

Note: [1] No = No treatment, US = Under-sampling, OS = Over-sampling. [2] The highest G-mean under all treatment-method-model combinations.

Table 5b
5-class Classification Performance (%) computation time (s) by Treatments and Method [C].

		Method [C]: Majority voting		
Treatments		No	US	OS
All 5 models	Overall accuracy	84.7	0.42	61.55
	Class 1	3.98	94.83	65.44
	Class 2	5.72	6.14	16.32
	Class 3	10.09	0	4.91
	Class 4	1.19	0	33.87
	Class 5	99.57	0	68.25
	G-mean	7.71	0	26.10
Increase (%) of G-means from no treatment		N/A	−100	238.6

logit function is used for the link function. The logistic regression establishes the relationship between the response variable $Y = (y_1, \ldots, y_n)$ given $p = (p_1, \ldots, p_n)$ and a set of k predictors, $X = (x_1, \ldots, x_k)$:

$$y_i|p_i \sim Bernoulli(p_i), \quad i = 1, \ldots, n$$

$$\text{logit}(p_i) = \log\left(\frac{p_i}{1 - p_i}\right) = \beta_0 + \beta_1 x_{1i} + \cdots + \beta_k x_{ki}$$

By solving for $\beta = (\beta_1, \ldots, \beta_k)$ using the maximum likelihood method, the probability of occurrence of crash severity level i, denoted by p_i, can be estimated (Dobson and Barnett, 2008).

$$p_i = \frac{\exp\{\beta_0 + \beta_1 x_{1i} + \cdots + \beta_k x_{ki}\}}{1 + \exp\{\beta_0 + \beta_1 x_{1i} + \cdots + \beta_k x_{ki}\}}$$

Based on the estimated p_i, crash severity level can be predicted.

The logistic regression can be binomial or multinomial. Multinomial logistic regression (or multinomial logit) is used for the multi-class response variables. The multinomial regression with the multinomial response variable, $Y = (y_1, \ldots, y_n)$ and multiple predictor variables, $X = (x_1, \ldots, x_k)$, consists of $C-1$ non-overlapping logit models. The probability of occurrence of crash severity level c (i.e., c^{th} logistic model), p_{ic}, can be estimated.

$$p_{ic} = P(y_i = c|X) = \frac{\exp\{\beta_{c0} + \beta_{c1} x_{1i} + \cdots + \beta_{ck} x_{ki}\}}{1 + \sum_{c=1}^{C-1} \exp\{\beta_{c0} + \beta_{c1} x_{1i} + \cdots + \beta_{ck} x_{ki}\}}, \quad p, \ c < C$$

Based on the estimated p_{ic}, crash severity level can be predicted. - In the program R, the 'h2o' function is used.

4.1.2. Decision tree and random forest

Classification and regression tree (CART) is a nonparametric decision tree learning method for classification and regression (Breiman et al., 1984). In CART, a decision tree is formulated by a collection of rules with respect to the variables in the training dataset. First, a rule for splitting the data at each node is selected. Next, with this rule, data at a given node are divided into two subsets. The same procedure is implemented recursively to each nested node. Splitting stops either

Table 6a
3-class Classification Performance (%) and computation time (s) by Treatments and Methods [A] and [B].

		Method [A]: No bagging			Method [B]: Bagging		
Treatments[1]		No	US	OS	No	US	OS
Multinomial logistic regression	Overall accuracy	85.08	84.37	1.79	84.37	1.79	1.79
	Class 1	7.33	0	100	0	100	100
	Class 2	8.33	0	0	0	0	0
	Class 3	99.32	100	0	100	0	0
	G-mean	18.24	0	0	0	0	0
	Time (s)	*9.04*	*1.60*	*7.71*	*1,280.90*	*1,148.03*	*2,641.23*
Decision tree	Overall accuracy	84.94	60.04	63.85	85.14	61.39	63.22
	Class 1	0	53.07	54.63	8.27	58.02	40.06
	Class 2	6.22	40.69	32.65	7.51	38.50	67.21
	Class 3	99.66	63.36	69.15	99.51	65.22	63.22
	G-mean	0	51.53	49.78	18.35	52.62	[2]**55.42**
	Time (s)	*25.43*	*0.91*	*71.50*	*39.31*	*3.45*	*108.72*
GBM	Overall accuracy	85.15	64.82	63.32	85.11	61.84	61.15
	Class 1	7.21	61.22	56.20	8.14	58.77	57.51
	Class 2	7.69	57.46	38.64	7.74	39.28	41.01
	Class 3	99.51	39.78	67.52	99.44	65.61	64.53
	G-mean	17.67	51.92	52.73	18.43	53.30	53.39
	Time (s)	*28.44*	*7.67*	*69.92*	*3,463.98*	*2,315.44*	*25,488.59*
NN	Overall accuracy	85.09	1.79	85.01	84.37	1.79	84.39
	Class 1	6.70	100	7.27	0	100	0
	Class 2	8.69	0	7.73	0	0	0.17
	Class 3	99.28	0	99.33	99.99	0	99.99
	G-mean	17.95	0	17.74	0	0	0
	Time (s)	*59.08*	*6.78*	*79.98*	*3,046.21*	*1,088.49*	*3,088.44*
NB	Overall accuracy	84.98	62.99	65.04	84.37	70.73	67.65
	Class 1	14.79	59.02	58.96	0	47.79	52.00
	Class 2	7.63	31.91	31.00	0	19.72	20.29
	Class 3	99.16	68.18	70.76	100	79.59	75.76
	G-mean	22.37	50.45	50.57	0	42.17	43.08
	Time (s)	*1.47*	*1.33*	*1.67*	*1,628.60*	*259.02*	*1,726.52*
Mean (SD) of G-means		15.2 (8.7)	30.8 (28.1)	34.2 (24.0)	7.4 (10.1)	29.6 (27.4)	30.4 (28.1)
Increase (%) of G-means from no treatment		N/A	101.9	124.1	N/A	302.6	312.9

Note: G-means higher than 65% in Method [A] are underlined. [1] No = No treatment, US = Under-sampling, OS = Over-sampling. [2] The highest G-mean under all treatment-method-model combinations.

Table 6b
3-class Classification Performance (%) computation time (s) by Treatments and Method [C].

		Method [C]: Majority voting		
Treatments		No	US	OS
All 5 models	Overall accuracy	85.14	53.98	61.40
	Class 1	6.89	59.64	59.27
	Class 2	8.19	46.07	36.54
	Class 3	99.43	55.16	65.52
	G-mean	17.77	53.32	52.16
Increase (%) of G-means from no treatment		N/A	200.1	193.5

when CART detects that there is no more gain to be obtained by growing the tree deeper, or when some pre-determined criterion as the stopping rule is satisfied. Given the defined branches and nodes of the tree, each response variable falls into one terminal node. This tree structure can be used for prediction. A random forest (RF) is a bagged decision tree. RF trains multiple decision trees on bootstrapped data from the original training data, following the bootstrap aggregation method. Each tree casts a vote for the response variable. The final decision is made by taking the majority vote (Friedman, 2001). - In the program *R*, the package ‘rpart’ is used.

4.1.3. Gradient boosting model (GBM)

A gradient boosting machine trains a number of weak learners sequentially, where each weak learner is trained based on the error of the previous weak learner to obtain a strong learner. In this study, we used 50 shallow decision trees as our weak learners (Friedman, 2002). - In the program *R*, the package ‘h2o’ is used.

4.1.4. Neural network (NN)

(Artificial) neural network is a learning algorithm inspired by the learning mechanism in the human brain. A neural network is constructed by three types of layers, namely input, hidden, and output layers. In a neural network, it is important to determine the appropriate number of hidden layers and the number of neurons (or nodes) in each hidden layer, since this structure severely affects the number of parameters, and hence the generalization behavior of NN (Ripley, 2007; Heaton, 2008). Based on several rule-of-thumb methods (e.g., the number of hidden nodes should be between the size of the input and output layers (Heaton, 2008)), a structure containing a single-hidden-layer with 10 nodes was used in the current study. - In the program *R*, the package ‘nnet’ is used.

4.1.5. Naïve Bayes classifier (NB)

A Naïve Bayes classifier is a model based on the Bayes’ theorem, under the assumption of conditional independence among features given the label (Zhang, 2004). Due to the conditional independence assumption on the features, the conditional distribution of the class variable over the features can be written as:

$$p(C_k x_1, \ldots, x_n) \propto p(C_k) \prod_{i=1}^{n} p(x_i C_k)$$

where C_k is the k th possible outcome, and $\boldsymbol{x} = (x_1, \ldots, x_n)$ represents a vector of n independent features. For parameter estimation, the maximum a posteriori method is used. In the current study, the likelihood distribution is assumed to be a Bernoulli distribution. - In the program *R*, the package ‘h2o’ is used.

Table 7a
2-class Classification Performance (%) and computation time (s) by Treatments and Methods [A] and [B].

		Method [A]: No bagging			Method [B]: Bagging		
Treatments[1]		No	US	OS	No	US	OS
Logistic regression	Overall accuracy	94.07	93.94	93.93	93.67	85.76	6.34
	Class 1	18.36	17.37	11.90	0.26	38.10	100
	Class 2	99.19	99.12	99.49	99.99	88.98	0
	G-mean	42.67	41.49	34.41	5.10	58.22	0.00
	Time (s)	*3.68*	*1.48*	*5.37*	*1,409.88*	*392.75*	*1,322.13*
Decision tree	Overall accuracy	94.04	92.11	92.11	93.66	93.71	93.71
	Class 1	15.79	35.25	35.25	0	5.67	5.67
	Class 2	99.34	95.96	95.96	100	99.67	99.67
	G-mean	39.61	58.16	58.16	0.00	23.77	23.77
	Time (s)	*26.49*	*2.51*	*52.96*	*230.91*	*187.78*	*427.38*
GBM	Overall accuracy	94.12	94.04	94.10	94.10	89.33	90.70
	Class 1	18.43	13.86	17.26	14.04	42.36	39.89
	Class 2	99.24	99.46	99.30	99.52	92.51	94.14
	G-mean	42.77	37.13	41.40	37.38	[2]**62.60**	61.28
	Time (s)	*38.47*	6.58	*22.15*	*1,805.61*	1,992.74	9,617.57
NN	Overall accuracy	93.90	93.76	94.01	94.15	93.74	93.66
	Class 1	6.56	3.43	18.04	15.16	1.87	0
	Class 2	99.81	93.76	99.15	99.49	99.95	100
	G-mean	25.59	17.93	42.29	38.84	13.67	0.00
	Time (s)	23.35	*3.48*	26.24	2,459.44	*1,398.15*	2,733.98
NB	Overall accuracy	93.90	93.87	93.89	93.97	93.78	93.60
	Class 1	11.09	10.49	9.90	9.00	8.19	8.54
	Class 2	99.50	99.52	99.57	99.72	99.57	99.36
	G-mean	33.22	32.31	31.40	29.96	28.56	29.13
	Time (s)	*1.43*	*1.34*	*1.62*	*786.46*	*409.52*	*1,314.67*
Mean (SD) of G-means		36.8 (7.4)	37.4 (14.6)	41.5 (10.4)	22.3 (18.4)	37.4 (21.8)	22.8 (25.3)
Increase (%) of G-means from no treatment		N/A	1.7	12.9	N/A	67.9	2.6

Note: G-means higher than 65% in Method [A] are underlined. [1] No = No treatment, US = Under-sampling, OS = Over-sampling. [2] The highest G-mean under all treatment-method-model combinations.

Table 7b
2-class Classification Performance (%) computation time (s) by Treatments and Method [C].

		Method [C]: Majority voting		
Treatments		No	US	OS
All 5 models	Overall accuracy	94.14	90.95	91.67
	Class 1	17.58	39.13	37.87
	Class 2	99.32	94.46	95.31
	G-mean	41.79	60.80	60.08
Increase (%) of G-means from no treatment		N/A	45.5	43.8

4.2. Treating data imbalance

Machine learning techniques can be typically viewed as heuristic methods to solve optimization problems with the objective of minimizing the classification error on a training dataset. When working with imbalanced data, therefore, the traditional techniques that aim at maximizing training accuracy are not adequate, and measures need to be taken to put all classes on the same standing. Since the dataset used in this study has imbalanced classes with only 0.34% of the records pertaining to fatal accidents, two treatments, namely under-sampling and over-sampling, are used

Assume we have C classes, where n_C denotes the number of samples in class $c \in \{1, ..., C\}$. Also assume that class IDs are ordered in an ascending order of the class size, i.e., $n_1 \leq n_2 \leq ... \leq n_C$. In under-sampling, we randomly select n_1 samples from classes 2 to C, without replacement. In over-sampling, we randomly select n_C samples from classes1 to $C-1$, with replacement.

4.3. Ensemble methods

Ensemble methods are machine learning algorithms that train a set of classifiers and use a weighted vote of the trained models when conducting classification. Ensemble methods help reduce the variance of out-of-sample error and therefore are helpful in avoiding overfitting. In this paper, we consider two ensemble methods, namely majority voting and bootstrap aggregation.

In the case of majority voting, we train a set of different models using various machine learning techniques and take the class with the highest number of votes as the final classification result of the ensemble method. We assume all models in the ensemble to have the same weight.

Bootstrap aggregating (also called bagging) is an ensemble meta-algorithm to learn from multiple classifiers, by re-sampling the training data with replacement and using majority voting on the classification results from the new (or bootstrapped) samples (Breiman, 1996). Fig. 1 shows how we have used bagging in this study. In the training stage, ninety-nine additional datasets (or 99 bags) were randomly re-sampled from the original training dataset (D = 30 variables, N - K = 207,980 training data points). In each bag, the number of data points was identical to the original training dataset, but the number of variables was reduced (d = 20). For each single classification model (i.e., each of the 5 machine learning models), the following procedure was followed: first, 99 bags were trained. Next, in the testing stage, the 99 trained models were used to classify the test data (K = 89,133). Next, the ultimate classification result was determined using majority voting. Finally, the classification results were compared to the actual labels on the test set to determine model accuracy. For example, in binary classification, if more than 49 out of 99 models classify the first data point in the testing set as '-1' and the original label of that element is '$+1$', this prediction is considered as a 'miss'. Once all the K test points are processed through this majority voting logic, the classification accuracy is calculated (based on the number of 'missed' and 'correct' classifications that exist among the total data elements).

5. Results

Fig. 2 shows the learning process used in this study. In the pre-training stage, three treatments (including no treatment) for the imbalanced data were applied to the training set. In the training-testing stage, three methods were used. First, with the three datasets created in the pre-training stage, five classification models were used to classify the fatality of motor vehicle crash injury (method [A]: No bagging). Second, a bagging approach was used for all the 5 models (method [B]: Bagging, meaning that bagging has been applied to each single model, and repeated for all 5 models). Third, we used majority voting on all the 5 models (method [C]: Majority voting among selected models). All executions regarding the learning and treatment processes were performed using R version 3.3.1. The computer specifications used in this study are: Intel® Xeon ® Processor E31230 CPU at 3.2 GHz, 32.0 GB RAM.

Tables 5a and 5b show performance results of the 5-class classification obtained through the learning process shown in Fig. 2, as well as computation time of each process. The tables include overall accuracy, each class's accuracy, and G-mean for the three treatments and three training-testing methods. Time for finding the optimal parameters using cross-validation is not included in the computational time. In the majority voting, we do not specify the learning time, because its prediction is based on the votes from 5 classifiers which are already trained. The 5-class classification performance was the highest (i.e., G-mean = 32.1) when bagging was used with decision trees and over-sampling treatment on the data.

The 3-class and 2-class classifications were also performed. The highest performance for 3-class classification (G-mean 55.4%; see Tables 6a and 6b), was obtained with bagged decision trees (i.e.,

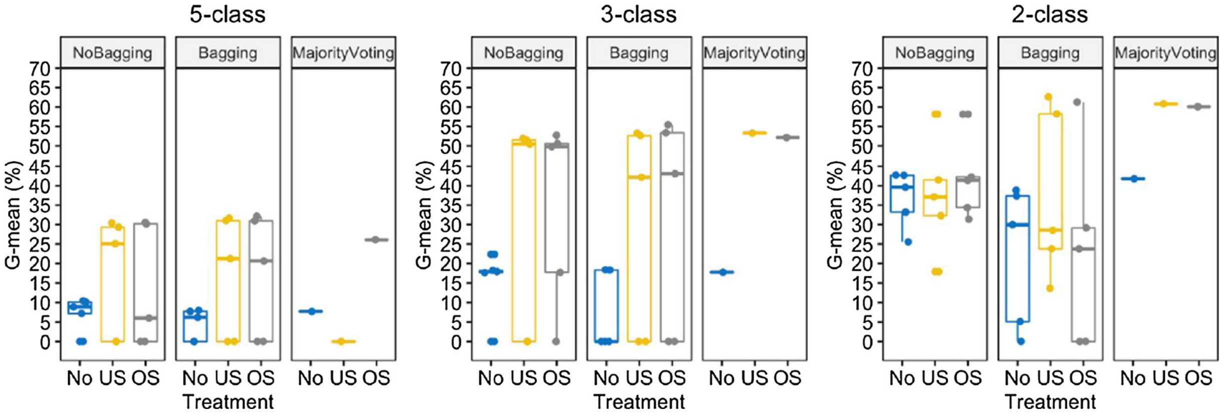

Fig. 3. G-means by treatments and training-testing methods.

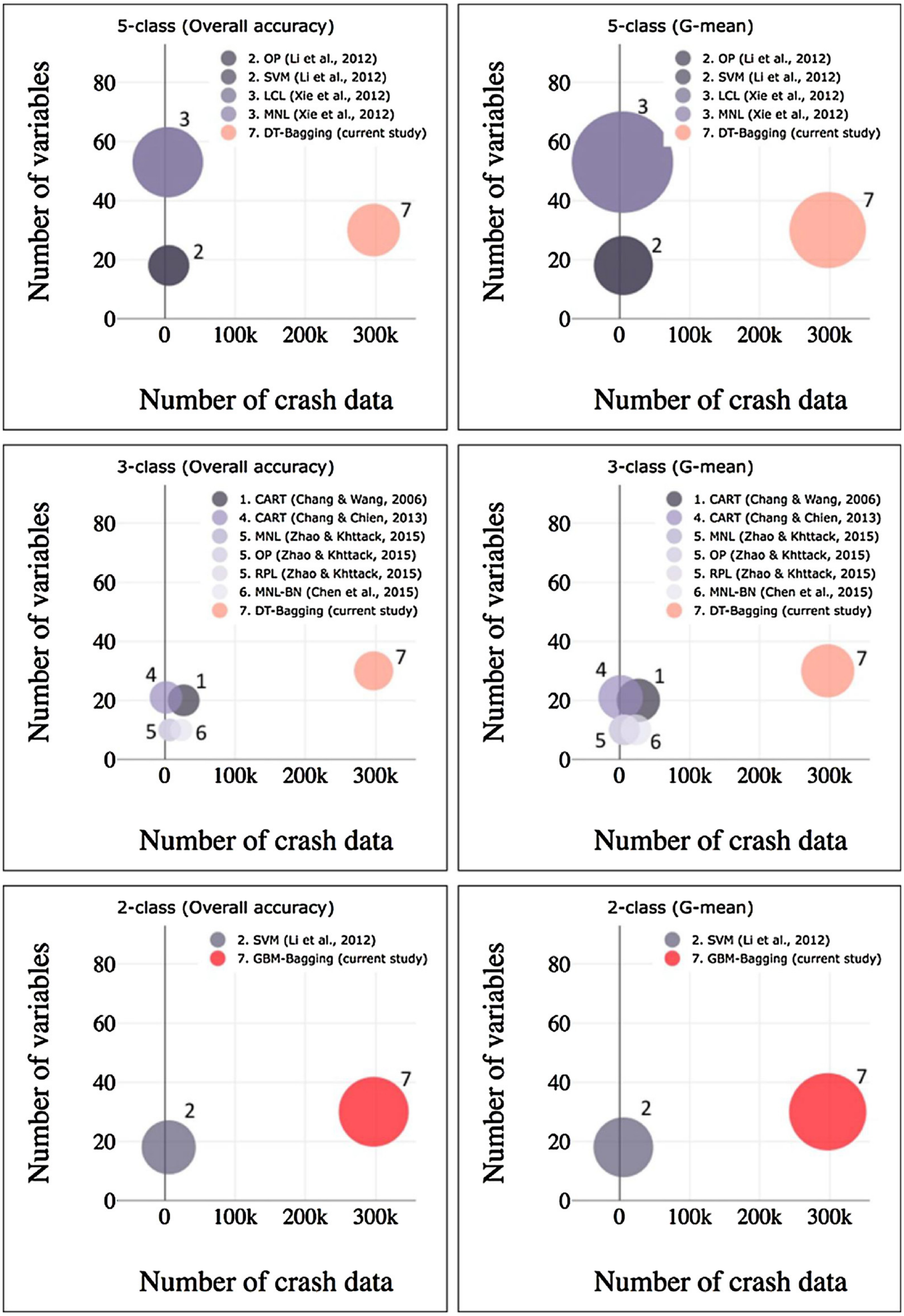

Fig. 4. Overall accuracy and G-mean comparisons by the numbers of crash data and variables.

random forest) and the over-sampling treatment, similar to the 5-class classification. In the 2-class classification, the GBM and under-sampling conditions produced the highest classification result (G-mean 62.6%; see Tables 7a and 7b).

6. Discussion

Fig. 3 displays the range for G-mean values of 5-, 3-, and 2-class classifications for the three methods [A- No Bagging], [B- Bagging], and [C- Majority Voting], and under 3 different data imbalance treatments discussed in the previous section. The first observation is that reducing the number of classes from 5-class to 3- and 2-class produces better performance, which can be also seen in previous literature (e.g., Lachiche and Flach, 2003; Tax and Duin, 2002). The results of 5- and 3-class classifications show the similar pattern. In 5- and 3-class classifications, when applying under- and over-sampling treatments to the imbalanced data with no bagging (i.e., method [A]) and bagging (i.e., method [B]) methods, the results have higher performance and vary widely based on the selected model, compared to no treatment. Bagging with over-sampling produces the best results in both 5- and 3-class classifications, where the algorithm producing the best G-mean is the decision tree. Bagging with under-sampling produces the best results in 2-class classifications, where the algorithm producing the best G-mean is GBM. Majority voting (method [C]) produces relatively high G-mean values with under- and/or over-sampling. From these observations, a general conclusion can be made that ensemble methods can be effective in treating imbalanced data when combined with either under-sampling or over-sampling treatments.

Fig. 4 shows the classification results (i.e., overall accuracy and G-mean) obtained from current and previous studies, by numbers of crash data and variables, based on all three categories of multi-class classification. In the 5-class classification, the current study does not outperform the result from Xie et al. (2012)' latent class logit model. However, the current study is based on a significantly larger and more recent dataset. In the 3-class classification, the current study matches the best results in the literature (i.e., CART model from Chang and Chien (2013)), with a higher number of variables and a larger dataset. In 2-class classification, our study outperforms the best classification results in the literature, obtained from Li et al. (2012)'s SVM model.

7. Limitations

Although the hybrid approach proposed in this study performed well in classifying motor vehicle injury severity levels, there are several factors that can allow for training classification models with higher predictive power. Specifically, the explanatory variables used in the current study could be expanded when used in the context of semi-autonomous vehicle systems. Although we extracted thirty variables from driver-related, environmental, and temporal conditions surrounding recorded accidents, most of which can be measured using sensors deployed on automated vehicles, other variables that can help improve the classification accuracy in semi-autonomous vehicle systems are missing. Examples include the vehicle's health status, an anomaly in driver performance, and the state of the surrounding traffic.

In this study, we used over-sampling and under-sampling to balance our crash injury severity dataset as a way to put all classes on the same footing when assessing the performance of our classification models. This allowed us to prevent the larger class from dominating the analysis. An alternative approach is to alter the cost function, by assigning higher costs to misclassification of more severe crashes. In this approach, the actual crash cost can be incorporated into the training model. As a future research direction, we will apply a crash costs-based approach (such as the one proposed in Iranitalab and Khattak (2017)), where we use the Actual Overall Crash Costs, Predicted Overall Crash Costs, Overall Prediction Error, and Specific Prediction Error, to train models that are informed by crash costs.

8. Conclusions

The purpose of this research is to train classification models that can predict the level of motor vehicle crash injury severity with both high accuracy and sensitivity rates at the same time. Paying special attention to sensitivity rates allows us to consider the harsher consequences of more severe accidents when training motor vehicle crash severity classification models. Such models can be used for two purposes: (i) identifying factors that contribute the most in the injury severity of motor vehicle crashes through conducting sensitivity analysis, and (ii) real-time prediction of crash injury severity in semi-autonomous vehicles. In the latter case, the models can inform the decision by the autonomous entity to take over the control from the human driver if the models predict severe accidents under the human control.

This study uses 5 machine learning algorithms, 2 treatments for balancing the originally imbalanced crash data, and 2 ensemble methods for classification of motor vehicle crash severity level, with a large (~300 K) crash dataset. Among all the trained models, bagging applied to decision trees (also known as random forests), combined with the over-sampling treatment, provided the highest classification performance in the 5- and 3-class classifications. In the 2-class classification, GBM with bagging and under-sampling produced the best classification outcome. It is also observed that bagging applied to all trained models, although not producing the best G-mean values, but produces results that are more robust, with smaller variance.

Acknowledgments

We would like to thank Dr. Carol Flannagan and Helen Spradlin from the University Michigan Transportation Research Institute's Center for the Management of Information for Safe and Sustainable Transportation (CMISST), and Sydney Smith and Amanda Heinze from Michigan State Police's Criminal Justice Information Center, for providing us with the crash data from the Michigan Traffic Crash Reporting System (TCRS).

References

Abdelwahab, H., Abdel-Aty, M., 2001. Development of artificial neural network models to predict driver injury severity in traffic accidents at signalized intersections. Transport. Res. Record 1746, 6–13. https://doi.org/10.3141/1746-02.

Blincoe, L., Miller, T., Zaloshnja, E., Lawrence, B., 2015. The Economic and Societal Impact of Motor Vehicle Crashes, 2010 (Revised). Report No. DOT HS 812 013. National Highway Traffic Safety Administration, Washington, DC.

Breiman, L., 1996. Bagging predictors. Mach. Learn. 24 (2), 123–140. https://doi.org/10.1007/BF00058655.

Breiman, L., Friedman, J., Olshen, R.A., Stone, C.J., 1984. Classification and Regression Trees. CRC press.

Castro, M., Paleti, R., Bhat, C.R., 2013. A spatial generalized ordered response model to examine highway crash injury severity. Accid. Anal. Prev. 52, 188–203. https://doi.org/10.1016/j.aap.2012.12.009.

Chang, L.Y., Chien, J.T., 2013. Analysis of driver injury severity in truck-involved accidents using a non-parametric classification tree model. Saf. Sci. 51 (1), 17–22. https://doi.org/10.1016/j.ssci.2012.06.017.

Chang, L.Y., Wang, H.W., 2006. Analysis of traffic injury severity: an application of non-parametric classification tree techniques. Accid. Anal. Prev. 38 (5), 1019–1027. https://doi.org/10.1016/j.aap.2006.04.009.

Chen, C., Zhang, G., Tarefder, R., Ma, J., Wei, H., Guan, H., 2015. A multinomial logit model-Bayesian network hybrid approach for driver injury severity analyses in rear-end crashes. Accid. Anal. Prev. 80, 76–88. https://doi.org/10.1016/j.aap.2015.03.036.

Chen, C., Zhang, G., Qian, Z., Tarefder, R.A., Tian, Z., 2016. Investigating driver injury severity patterns in rollover crashes using support vector machine models. Accid. Anal. Prev. 90, 128–139. https://doi.org/10.1016/j.aap.2016.02.011.

Dobson, A.J., Barnett, A., 2008. An Introduction to Generalized Linear Models. CRC press.

Eluru, N., Bhat, C.R., Hensher, D.A., 2008. A mixed generalized ordered response model for examining pedestrian and bicyclist injury severity level in traffic crashes. Accid. Anal. Prev. 40 (3), 1033–1054. https://doi.org/10.1016/j.aap.2007.11.010.

Farchi, S., Camilloni, L., Rossi, P.G., Chini, F., Borgia, P., Guasticchi, G., 2007. Home injuries mortality: sensitivity and specificity analysis of different data sources and operative definitions. Accid. Anal. Prev. 39 (4), 716–720. https://doi.org/10.1016/j.aap.2006.11.002.

Friedman, J.H., 2001. Greedy function approximation: a gradient boosting machine. Ann. Stat. 29 (5), 1189–1232.

Friedman, J.H., 2002. Stochastic gradient boosting. Comput. Stat. Data Anal. 38 (4), 367–378. https://doi.org/10.1016/S0167-9473(01)00065-2.

Heaton, J., 2008. Introduction to Neural Networks with Java. Heaton Research, Inc.

Iranitalab, A., Khattak, A., 2017. Comparison of four statistical and machine learning methods for crash severity prediction. Accid. Anal. Prev. 108, 27–36. https://doi.org/10.1016/j.aap.2017.08.008.

Kubat, M., Holte, R., Matwin, S., 1997. Learning when negative examples abound. Mach. Learn. ECML-97 1224, 146–153. https://doi.org/10.1007/3-540-62858-4_79.

Lachiche, N., Flach, P.A., 2003. Improving accuracy and cost of two-class and multi-class probabilistic classifiers using ROC curves. Proceedings of the 20th International Conference on Machine Learning (ICML-03). pp. 416–423.

Lee, C., Li, X., 2015. Predicting driver injury severity in single-vehicle and two-vehicle crashes with boosted regression trees. Transport. Res. Record 2514, 138–148. https://doi.org/10.3141/2514-15.

Li, Z., Liu, P., Wang, W., Xu, C., 2012. Using support vector machine models for crash injury severity analysis. Accid. Anal. Prev. 45, 478–486. https://doi.org/10.1016/j.aap.2011.08.016.

National Center for Statistics and Analysis, 2015. Motor Vehicle Crashes: Overview. Traffic Safety Facts Research Note. Report No. DOT HS 812 318). National Highway Traffic Safety Administration, Washington, DC 2016.

Office of Highway Safety Planning, 2017. Michigan Traffic Crash Facts. (Accessed 16 June 2017). http://www.michigantrafficcrashfacts.org.

Parikh, R., Mathai, A., Parikh, S., Sekhar, G.C., Thomas, R., 2008. Understanding and using sensitivity, specificity and predictive values. Indian J. Ophthalmol. 56 (1), 45.

Ripley, B.D., 2007. Pattern Recognition and Neural Networks. Cambridge university press.

Tax, D.M., Duin, R.P., 2002. Using two-class classifiers for multiclass classification. Pattern Recognition, 2002. Proceedings. 16th International Conference on (Vol. 2, pp. 124-127). IEEE.

Xie, Y., Zhao, K., Huynh, N., 2012. Analysis of driver injury severity in rural single-vehicle crashes. Accid. Anal. Prev. 47, 36–44. https://doi.org/10.1016/j.aap.2011.12.012.

Xu, C., Tarko, A.P., Wang, W., Liu, P., 2013. Predicting crash likelihood and severity on freeways with real-time loop detector data. Accid. Anal. Prev. 57, 30–39. https://doi.org/10.1016/j.aap.2013.03.035.

Yu, R., Abdel-Aty, M., 2013. Utilizing support vector machine in real-time crash risk evaluation. Accid. Anal. Prev. 51, 252–259. https://doi.org/10.1016/j.aap.2012.11.027.

Zhang, H., 2004. The optimality of naive bayes. Proceedings of the 17th International Florida Artificial Intelligence Research Society Conference, American Association for Artificial Intelligence 1 (2), 562–567.

Zhao, S., Khattak, A., 2015. Motor vehicle drivers' injuries in train–motor vehicle crashes. Accid. Anal. Prev. 74, 162–168. https://doi.org/10.1016/j.aap.2014.10.022.

Zhu, X., Srinivasan, S., 2011. Modeling occupant-level injury severity: an application to large-truck crashes. Accid. Anal. Prev. 43 (No. 4), 1427–1437. https://doi.org/10.1016/j.aap.2011.02.021.