Transportation Research Part C 118 (2020) 102708

Contents lists available at ScienceDirect

Transportation Research Part C

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

A real-time crash prediction fusion framework: An imbalance-aware strategy for collision avoidance systems

Zouhair Elamrani Abou Elassad[a,*], Hajar Mousannif[a], Hassan Al Moatassime[b]

[a] LISI Laboratory, Computer Science Department, FSSM, Cadi Ayyad University, Morocco
[b] OSER Research Team, Computer Science Department, FSTG, Cadi Ayyad University, Morocco

ARTICLE INFO

Keywords:
Real-time crash prediction
Machine learning
Fusion framework
Driving behavior
SMOTE

ABSTRACT

Real-time traffic crash prediction has been a major concern in the development of Collision Avoidance Systems (CASs) along with other intelligent and resilient transportation technologies. There has been a pronounced progress in the use of machine learning models for crash events assessment by the transportation safety research community in recent years. However, little attention has been paid so far to evaluating real-time crash occurrences within information fusion systems. The main aim of this paper is to design and validate an ensemble fusion framework founded on the use of various base classifiers that operate on fused features and a Meta classifier that learns from base classifiers' results to acquire more performant crash predictions. A data-driven approach was adopted to investigate the potential of fusing four real-time and continuous categories of features namely physiological signals, driver maneuvering inputs, vehicle kinematics and weather covariates in order to systematically identify the crash strongest precursors through feature selection techniques. Moreover, a resampling-based scheme, including Bagging and Boosting, is conducted to generate diversity in learner combinations comprising Bayesian Learners (BL), k-Nearest Neighbors (kNN), Support Vector Machine (SVM) and Multilayer Perceptron (MLP). To ensure that the proposed framework provide powerful and stable decisions, an imbalance-learning strategy was adopted using the Synthetic Minority Oversampling TEchnique (SMOTE) to address the class imbalance problem as crash events usually occur in rare instances. The findings show that Boosting depicted the highest performance within the fusion scheme and can accomplish a maximum of 93.66% F1 score and 94.81% G-mean with Naïve Bayes, Bayesian Networks, k-NN and SVM with MLP as the Meta-classifier. To the best of our knowledge, this work presents the first attempt at establishing a fusing framework on the basis of data from the four aforementioned categories and fusion models while accounting for class imbalance. Overall, the method and findings provide new insights into crash prediction and can be harnessed as a promising tool to improve intervention efforts related to traffic intelligent transportation systems.

1. Introduction

Traffic accidents has been widely recognized as one of the most major concerns that encounters societies nowadays, resulting in great loss of lives and economic damages. The World Health Organization (WHO, 2017) reports that 1.35 million people die in road traffic crashes every year, and a further 20–50 million are injured or disabled worldwide. As such, understanding under what

* Corresponding author.
E-mail address: z.elamrani@uca.ma (Z. Elamrani Abou Elassad).

https://doi.org/10.1016/j.trc.2020.102708
Received 25 November 2019; Received in revised form 6 June 2020; Accepted 23 June 2020
Available online 10 July 2020

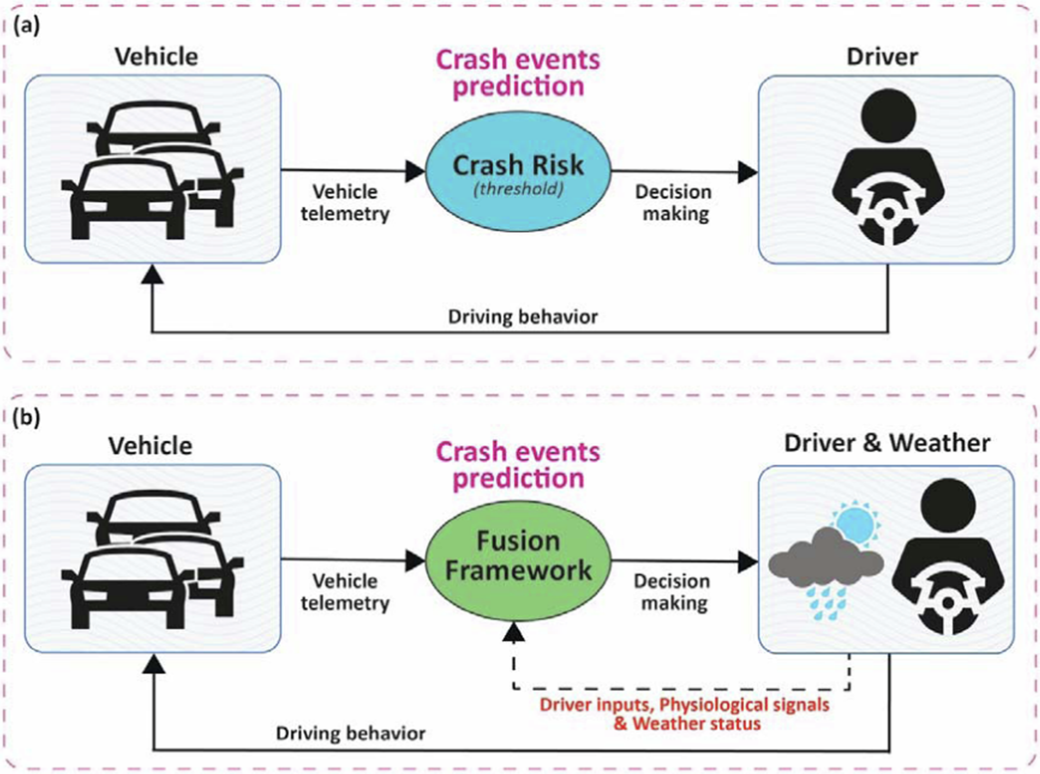

Fig. 1. Design of CAS without/with driver inputs, physiological signals and weather status investigation.

circumstances road crashes occur and which factors lead to the likelihood of a car accident, would have a substantial influence on developing efficient policy interventions in order to avoid accidents from arising. Accordingly, a variety of sophisticated systems have been developed to reduce the frequency and severity of traffic crashes. One actively involved safety technology, Collision Avoidance Systems (CASs) are implemented to predict impending accidents, ensure warnings to drivers or perform autonomous decisions (Seiler et al., 1998). Empirical research and market feedbacks have proven the efficiency of CASs in promoting productive driving conduct, as well as drivers' personal estimates (Cui et al., 2018; Hoffenson et al., 2013; Jamson et al., 2008). Most of crash predictions in real-time driver assistance systems are based on information gathered consistently from loop detectors and radars sensors (Hassan and Abdel-Aty, 2013; Seiler et al., 1998; Yu and Abdel-Aty, 2013a). This data entail multiple vehicle kinematics such as speed and Time-to-Collision (TTC), which are then operated by the algorithms embedded in an on-board electric system. The algorithms evaluate the crash risk level or collision likelihood, and determine whether, when and how to apply the preliminarily interventions (e.g. alert or stop). In this regard and from a systemic standpoint, the CAS is a close-loop system including drivers' behavior. As can be seen in Fig. 1(a), vehicle telemetry during the driving process are continuously collected then forwarded to the algorithms which assess the current risk level; decisions are made afterwards in attempt to take the appropriate interventions. However, the existing CAS scheme and its crash analysis rarely examine the relation between drivers' state along with weather status and crash possibilities. Therefore, to further boost the predictability of crash events, one promising procedure is to systemically evaluate the situational hazards of vehicle, environment and driver.

Traffic crash is a complex paradigm, affected by various contributing factors such as the driver state and environmental factors (Aljanahi et al., 1999). Hazardous traffic conditions and unsafe driving behaviors have been examined in numerous previous studies in order to characterize road crashes and develop efficient real-time traffic management strategies (Ahmed and Abdel-Aty 2012; Elamrani Abou Elassad et al., 2020b; Shi et al. 2018). To achieve a performant crash analysis, the identification of pre-accident risk exposure is crucial as the likelihood of crash events is typically related to multiple traffic flow metrics including, amongst others, driver physiological state and inputs, the vehicle telemetry and weather conditions; crash prediction models can be then harnessed to proactively promote potential countermeasures and improve traffic flow management. Based on the above-mentioned assessments, a more elaborate design is proposed in Fig. 1(b) in which the driver's inputs, physiological state and weather conditions have been also underlined through a machine learning fusion framework. It is critical to note that during the development of crash prediction strategies, the choice of the adopted data and modeling techniques along with the followed steps to construct these models have a huge impact on the resulting performance measures. As such, with the development of machine learning strategies, the consolidation of data from diverse sources with distinct properties plays a crucial role in improving the solution of the issue at hand. Many categories of data are employed in crash analysis in an attempt to designate the beneficial information for the assessment. In this direction, physiological signals along with driver inputs, vehicle kinematics and weather status have been utilized to identify crash strongest precursors in the development of crash prediction paradigms. Vehicle telemetry and driver input responses have been used extensively in the investigation of crash and near-crash events (Ba et al., 2016; Perez et al., 2017). On the other hand, it was found that more than 1.25 million accidents are caused yearly due to weather conditions (21% of all vehicle crashes), leading to about 418,000 injuries (19% of crash injuries), and nearly 5000 casualties (16% of all casualties) (FHWA, 2016). Weather conditions have been evaluated in crash examination procedures and found to be highly influential (Elamrani Abou Elassad et al., 2020c; Madanat and Liu, 1995; Wang et al., 2015). On top of that, the Heart Rate Variability (HRV) physiological signal has been also used in the analysis as it is a common non-invasive indicator that can dynamically reflect the accumulation of mental workload (Chen et al., 2017).

Within this context, machine learning models have proven to be highly effective in predicting forthcoming events and have reported satisfying results in many transportation systems (Elamrani Abou Elassad et al., 2020a). In recent years, the multiple classifier ensemble system (MCS), an advanced machine learning technique, has been widely adopted in the development of new strategies for obtaining effective and higher performance in predictions; in effect, the optimization of a set of fairly straightforward learners seems more practical than tuning the structure of a single complex learner (Barak and Sadegh, 2016). In practice, three essential key points are vital for constructing an effective MCS model: performance of individual learners, diversity among learners, and the selection of the fusion methods that will be utilized (Britto et al., 2014; Cruz et al., 2018); the design of this combination provides an enhanced precision with single predictors and exclude the uncorrelated errors made by individual learners on different sections of the dataset. Within this context, choosing a variety of different classifiers with inherited diversity among them promote the well attainment of the established MCS. In fact, learners that are effective in distinct scopes are assumed to be diverse as the whole purpose of fusing multiple learners is to set up a certain stability to compensate for the weaknesses and limitations of individual classifiers. The techniques that foster diversity approaches are typically sorted as implicit and explicit methods (Haghighi et al., 2011). Implicit techniques are indirect and often use different strategies such as different weight initializations and different training data in order to create diversity. Bagging, as an implicit method, randomly divides the training set into several subsets with replacement to train each individual learner so that diversity could be established (Breiman, 1996). Unlike implicit techniques, explicit methods are explicit and generally aim to tune certain characteristics during performing diversity. Boosting is one type of explicit diversity approaches that directly manipulate the training data distributions based on the knowledge gained by previous learner to make some sort of diversity in the combination procedure. Additionally, the problem of class imbalance should be considered in crash prediction strategies. Crash related observations generally generate imbalanced input spaces since the target classes are not equally represented which leads to imbalances causing a bias toward the majority class given that modeling classifiers prioritize the class with the higher number of observations causing an over-prediction of the this class (Fernández et al., 2009). Handling the issue of imbalanced dataset is a challenging procedure for which practitioners are seeking to enhance and harness different technologies. To this end, Synthetic Minority Oversampling Technique (SMOTE), deemed as one of the most powerful re-sampling algorithms, was presented by (Chawla et al., 2002) to solve the imbalance issue by producing synthetic instances from the minor class and have been applied at the data level of this work to rebalance the training sets. It's true that over-generalization may occur when SMOTE is adopted which may increase the overlapping between classes, and there have been other sampling methods such as Adaptive Synthetic sampling and Generative Adversarial Network; However, it was found that these synthetic oversampling approaches may generate incorrect or unnecessary samples in some scenarios (Zhang et al., 2018). Moreover, an extensive research have proven that SMOTE has a better efficiency than under-sampling and over-sampling techniques (Batuwita and Palade, 2013; Kaur and Gosain, 2018; Nguyen et al., 2011). To our knowledge, there has been a limited interest, if not at all, at adopting an imbalance-learning formulation in a fusion architecture for crash prediction investigation.

In the present paper, in which we aim to examine the prediction performance for crash events, we propose a fusion design framework that supports the combination of several dissimilar classifiers working on fused information. First, four categories of data, namely physiological signals, driver input responses, vehicle kinematics and weather conditions are collected, fused and processed through the two feature selection techniques Random Forest and Principal Component Analysis along with SMOTE technique. Then, cross-validation is conducted on the input space and an optimum number of learners' sets is induced from the dataset generating a pool of base learners. Followed by the characterization of a classifier selection strategy in which the performance of the learners on the dataset is assessed and the efficient combinations are chosen for the fusion stage. Finally, in the fusion stage, Bagging and Boosting are carried out to the selected learners' sets of the preceding stage and one fusion method as a Meta-classifier learns from their predictions to produce the final prediction. To the best of our knowledge, little to no research has explored the impact of extensive information fusion incorporating the HRV physiological signal, driver maneuvering inputs, vehicular telemetry and various weather patterns within a fusing strategy for crash prediction. The contributions of the paper are summarized as follows:

i. Designing an advanced crash prediction ensemble fusion framework for Collision Avoidance Systems (CAS).
ii. Acquisition of multiple categories of features for information fusion.
iii. Information fusion processing through approved approaches for crash analysis.
iv. Implementing various diversity techniques in order to ensure more performant predictions.
v. Establishing a base classifier selection procedure by considering the performance of combined learners.
vi. Developing a fusion strategy to generate new predictions and comparing them with the outcomes of the selected classifiers sets.

The remainder of this study is organized as follows. Section 2 discusses related work focusing on previous efforts in crash events classification along with a brief review on simulator-based data acquisition and fusion in ensembles schemes. Section 3 covers the proposed methodology. Section 4 explains the experimental design. Section 5 lists the obtained results. Section 6 contains the discussion. Finally, Section 7 conclusions with future scopes of the present study are mentioned.

2. Related work

2.1. Real-time crash classification

Safety assessment studies have adopted multiple machine learning techniques to unravel the influencing factors of crashes and identify patterns in risky traffic situations in an attempt to improve operations and safety management. Different studies have investigated various methodologies and suggested different models to predict crash occurrences; disaggregate data have been employed and crash occurrence is assessed as a binary variable having two outcomes, namely crash and non-crash (Ahmed and Abdel-Aty, 2012; Basso et al., 2018), whereas (Lee et al., 2002) presented an aggregated linear crash prediction model based on crash and traffic data before and after the crash. The widely implemented machine learning based techniques for real time crash analysis have been deemed very helpful to characterize general relatioships between accident occurrences and coexisting features or contemporary scenarios;. Accordignly, binary logistic models are frequently developed (Theofilatos, 2017; Yu and Abdel-Aty, 2013b). Other non-traditional techniques have also been followed Support vector machine (Dong et al., 2015; Yu and Abdel-Aty, 2013a), k-nearest neighbors (Ali et al., 2019; Theofilatos et al., 2019) and Neural Networks (Liu et al., 2018; Wang et al., 2019b). Recently, more research focused on using Bayesian Learners such as Naïve Bayes, Bayesian Networks and Tree Augmented Naive Bayes (Hossain and Muromachi, 2012; Kwon et al., 2015; Mujalli et al., 2016). With respect to feature selection, multiple modelling techniques have been adopted such as the classification and regression tree (CART) model (Yu and Abdel-Aty, 2013a), random multinomial logit model (Hossain and Muromachi, 2012) and random forest (Siddiqui et al., 2012) which have been widely used for feature selection in crash investigations. The current knowledge of approaches employed in this field have been compared and summarized (Halim et al., 2016); challenges and opportunities for assessing the impact of various influecing factors on the crash occurrence prediction and unsafe driving patterns have also been adressed. Even though previous research have been found to be capable of predicting real-time crash frequency in different set ups, diverse modeling approaches result in different prediction performances, there is still much to be investigated, especially for acquiring better knowledge of detailed pre-crash precursors and conditions for better proactive safety management.

On another aspect, the pattern of class distributions in the dataset is a crucial characterization. Real-time crash analysis is an imbalanced classification issue as crash events typically occur in rare instances tending to be underrepresented in the input space. When it concerns a binary classification problem, a dataset is deemed to be imbalanced when the number of instances in one class is substantially greater than the other one. The class with more instances is called the major class while the one with fairly fewer data points is designated as the minor class. The case-control proportion has been assumed to varry from 1:1 to 1:5 (Roshandel et al., 2015), which means that for every crash situation, 1 to 5 non-crash cases were considered. For instance (Yu et al., 2019; Yu and Abdel-Aty, 2013b) used a 1:4 ratio, whereas in the work of (Ahmed et al., 2012), 301 crash cases and 880 non-crash cases were anlyzed, which is a roughly 1:3 ratio. Higher crash to non-crash propotions have been also employed but only rarely; (Wang et al., 2013) utilized a 1:10 ratio using a dataset comprising 125 crash and 1250 randomly selected non-crash cases, while (Xu et al., 2013) adopted a 1:20 ratio based on 794 crash cases and 15,880 non-crash cases. Within this context, the imbalanced dilemma is also regularly perceived in a large range of classification problems (Haixiang et al., 2017) such as software defect prediction, medical evaluation of rare disease, network intrusion detection, etc. There are two essential elucidation to deal with class imbalance challenge: (i) Cost-sensitive learning assigns weights to the target classes in an attempt to incite learners to give more emphasis to the minority class (Pazzani et al., 1994); (ii) Re-sampling the input space by means of over-sampling and under-sampling. SMOTE (Chawla et al., 2002), held to be one of the most effective re-sampling techniques, have been applied at the data level of this work to rebalance the training sets. In particular, SMOTE have been shown to be successfuly thanks to its ability to generate larger and less specific decision regions, to deal with noisy and sparse datasets (Parsa et al., 2020). As such, relevant work has employed SMOTE technique in resampling class distrubutions for crash analysis (Ke et al., 2018; Kitali et al., 2019; Wang et al., 2019b).

2.2. Simulator-based data acquisition

Using driving simulator studies in the field of transportation research has been increasing in the last several years as they imitate the driving behavior in a safe environment, with the main gain of possessing full empirical control over conditions and the capability to examine multiple design designs (Elamrani Abou Elassad and Mousannif, 2019). There are several instructed simulator-based practices to evaluate risky behavior leading to crash occurrences. Four key modeling metrics have been used in the literature to assess crash events, but rarely fused together. These include: physiological signals, driver inputs, vehicle kinematics and weather conditions.

Physiological features have been viewed as the most accurate measures to monitor driver's vigilance level based on information including brain waves and heart rate variability amongst others (Ba et al., 2017; Wang et al., 2010); while some of these features is considered intrusive as they collect data using ponderous devices that may distract the driver or perturb the driving conduct, here we utilize a non-invasive wrist band to compute the HRV which have been broadly adopted in similar investigations being an efficient workload metric that provides continuous basic data about the autonomic nervous system (Backs et al., 2003; Chen et al., 2017). In terms of driver maneuvering inputs and vehicle telemetry, scholars aimed to employ these metrics for crash monitoring in extensive experimental simulations (Ba et al., 2017, 2016; Rosey and Auberlet, 2014). They generally enable an analysis of driving performance by exploring the aptitudes of the driver while conducting the vehicle and have the benefit to be non-intrusive, real-time, continuous, and robust; they integrate inputs like steering angle, throttle/brake pedal positions, vehicle speed and acceleration, lane position deviation and others. Vehicle telemetry features have been grouped into two essential classes (Aghaei et al., 2016) (i) vehicle reaction to driver input (e.g. turn, jerk) and (ii) vehicle status as regards the surroundings (e.g. time to collision, time to lane exit).

On the other hand, albeit there are many research evaluating the impact of weather status in predicting crash instances (Theofilatos et al., 2019; Zeng et al., 2017), the prediction of crash events in multiple weather patterns have not been extensively explored. Moreover, the majority of research adopting weather variables in crash analysis are based on data gathered from accidents police reports, which could be sensitive to inaccuracies as the recorded conditions may be what the person filling the crash report observed and not the actual weather status at the moment of accident (Naik et al., 2016). Hence, simulator induced weather patterns have the advantage of being real-time, controllable and reliable, which motivates our intention regarding this type of features. As such, and building on the aforementioned analysis, a more effective crash assessment with the goal of providing drivers and road managers with more reliable information that fused with various computational techniques is needed.

2.3. Fusion in ensembles learning

Fusion methods pertain to the techniques used to acquire classifier ensembles. A suitable fusion method is the one that can harness the strength of particular learners and optimally consolidate their results to provide the final decision of the system (Woźniak et al., 2014). Machine learning fusion techniques are famous fusers that exploit the performance of individual learners as training data and then implement a meta-classifier that learns from the decisions of a learners set to procure better performance (Ferreiro et al., 2011). An instrumental factor in the development of MCS based architecture is the diversity concept (Tsymbal et al., 2005). In general, diversity can be accomplished by eliciting alterations in learner's parameters (e.g., error penalty and kernel parameters for SVM) (Windeatt, 2005), learner types (e.g., employing ensemble members with distinct types of classifiers) and learners' training input space (e.g., based on data resampling procedures like Bagging and Boosting) (Kuncheva, 2004). K-fold cross validation, which is a popular method for data-partitions overlap reduction (Krogh and Vedelsby, 1994), along with Bagging and Boosting have been adopted in this work in order to achieve diversity along with developing different learners (Bayesian Learners, kNN, SVM and MLP).

Relevant scholars aimed to predict crash occurrences based on hybrid methods. A mixed method of random forest and the matched case-control logistic mode was constructed in Hassan and Abdel-Aty (2013) in order to predict the reduced visibility-related crashes on freeways with real-time traffic flow data. (Sun and Sun, 2016) proposed a hybrid SVM classifier in combination with a k-mean clustering technique for crash analysis, a validation of approach transferability has been conducted as well. Another approach using Dynamic Time Warping and Hidden Markov Models in association with the built-in accelerometer in mobile phones have been developed to detect and report car accidents in (Aloul et al., 2015). A novel ensemble technique for crash prediction based on road geometric alignments and traffic with variable selection techniques have been proposed in the work of (Wu et al., 2019). Also, (Yu et al., 2019) explored the heterogeneous influencing factors and the causal relationships between crash occurrence and microscopic traffic flow variables in a hybrid modeling approach with latent class logit and path analysis. Ensemble-learning in general can efficiently strengthen models' decision, generalizability, and robustness over a single model (Krawczyk et al., 2017). For this previous research, however, gaps still exist in terms of methods and analytical results. For instance, none of them has developed an MCS-based framework and compared the performance of different learners sets for crash prediction. Also, investigation on the basis of physiological signals combined with weather conditions and the other of features have rarely explored. Moreover, an imbalance aware strategy accounting for crash and no-crash events distributions in the datasets was not established. One of the primary goals of this work is to address these research gaps.

As related research has shown that adopting hybrid methodologies in crash occurrences assessment is still in its infancy, employing an MCS fusion approach with a meta-classifier is relatively limited. Conventionally, a three-level strategy for constructing a substantiated fusion system have been defined in the literature (Gravina et al., 2017; Kanjo et al., 2018): (i) the “Data Dimension” fusion with the goal of collecting numerous information inputs from various sources to supplement each other (ii) “Feature Dimension” fusion is conducted during data preprocessing to find the suitable set of variables for the classification. (iii) “Decision Dimension” fusion, which intends to combine the outcomes of multiple techniques to enhance decision making. All these methodological levels have been respected in the development of our proposed MCS framework, different features from distinct source have been acquired and processed with variable selection techniques, further, several machine learning algorithms are used and fused for better results. The motivation of this research is to continue extend systematic advances in crash prediction, which can aid governmental agencies to build powerful safety policies, and ultimately, proactively promote traffic safety strategies.

3. Proposed methodology

This work aims to provide an effective crash prediction approach based on multi-classifier ensemble system, using four categories of features for information fusion, namely physiological signals, driver input responses, vehicle kinematics and weather conditions. This section describes the methodology used to capture various features, employing machine learning classifiers, pre- and post-processing of the data and the adopted feature selection techniques.

Represented as an integrated framework, this proposal involves data acquisition during driving trials in the first stage. For data collection, a non-invasive smartwatch was used to record the physiological signals of Heart Rate Variability (HRV), whereas driver inputs, vehicle kinematics and weather conditions were obtained from the driving simulator. Once the data is recorded the standard preprocessing procedure is applied. Accident event probability is considered to be generally small which will lead to have a highly imbalanced classification problem, SMOTE technique was adopted to handle the issue of imbalanced class distribution. The next step of the framework is to reduce the dimensions of the input signal. This promotes speeding up the classification and it also reduces

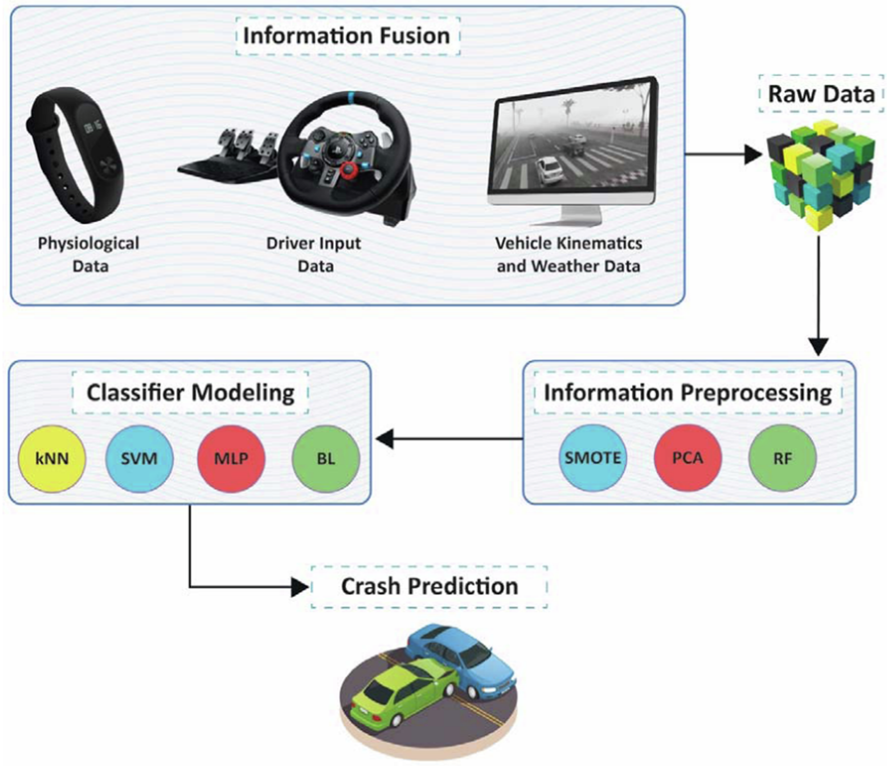

Fig. 2. Abstract overview of the proposed framework.

computational complexity and error of estimates. For this, two feature selection techniques are utilized, namely, Principal Component Analysis (PCA) and Random Forest (RF) techniques. The framework's following step comprises of adopting fusing methods in developing a multi-classifier ensemble system. The goal of this combination strategy is to benefit from increased performance with appropriate single classifiers and exclude errors made by individual classifiers on different portions of input space. Various base classifiers including SVM, MLP, kNN and BL have been adopted as ensemble learners, and two methods are employed: Bagging and Boosting in order to create distinctions in classifiers' training dataset. Adopting several feature selection and classification techniques allow the results' generalization and also the identification of the best effective technique for the problem at hand.

The parameters of SVM are optimized using Particle Swarm Optimization (PSO) algorithm to enhance the classification performance. The proven efficiency of the PSO, along with its search approach simplicity and ease of implementation, make PSO technique more advantageous for developing model selection than other evolutionary algorithms (Escalante et al., 2009; Guo et al., 2008). Finally, in the fusion step, Bagging and Boosting are applied to the diverse base classifiers and one fusion algorithm as a Meta-classifier learns from their predictions to present the final predictions. Fig. 2 depicts the complete process for the proposed framework.

3.1. Information fusion

Information fusion is the consolidation of data from diverse sources with distinct properties. In crash investigation, there is a growing trend to examine the pre-crash risk exposure as the likelihood of crash occurrences is typically related to multiple traffic flow metrics, therefore, physiological signals along with driver inputs, vehicle kinematics and weather status have been utilized to identify crash strongest precursors in the development of crash prediction algorithms. Heart rate variability (HRV) is a frequent non-invasive measure to monitor the activities of the autonomic nervous system. HRV can powerfully reflect the mental workload growth which makes it a well-suited physiological estimator for crash analysis. Alternatively, vehicle telemetry such as speed and yaw angle, driver inputs like steering wheel position and pedal positions which have proven to hold a crucial impact on the identification of crash and near-crash events have also been endorsed in this study. Furthermore, several weather scenarios namely clear, heavy fog and heavy rain have been simulated during the experiments and found to be highly influential. These categories of features were fused to supplement each other, then information pre-processing was carried out in order to cope imbalanced class distributions.

3.2. Feature selection

After pre-processing, feature selection is an essential step for classification models aiming to reduce the dimensionality of the input space variables as some of them may correlate with each other or not hold any meaningful impact on crash events. Thus, incorporating a large amount of features without any approach of variable selection can cause higher error of estimates (Kan et al., 2019). Therefore, it is essential to examine the dataset in an attempt to identify crash strongest precursors. The feature selection techniques adopted in this paper are listed as follows.

3.2.1. Random forest (RF)

RF is a machine-learning method that consists of an ensemble of randomized classification and regression trees (Breiman, 2001). RF models have been widely used for feature selection in crash investigations (Siddiqui et al., 2012; Yu et al., 2019). The configuration of each tree of the RF is made through a random sampling with replacement of cases which serves to grow the tree, followed by a selection of a sample among all the variables, which is then used to split the nodes. A major advantage of RF is its ability to capture complex and non-linear relationships between predictors and outcomes (Brokamp et al., 2017). Both the Gini index and classification accuracy of out-of-bag (OOB) data, are frequently utilized by RF model to evaluate feature importance. The importance of a variable in a tree is estimated in its capability to minimize an impurity index of nodes when exploited as a split feature. In this paper, we measured the Gini index to assess features importance and to select the most effective variables for building our models.

3.2.2. Principal component analysis (PCA)

The PCA is a dimension reduction technique that finds a linear transformation of the input features creating projections of the original variables to a new variable space, the new features are called the principal components (Wold et al., 1987). The new features are sorted in reference to their eigenvalues in order to reduce the dimensions; the variables with larger variance are selected (Jolliffe, 2010). To measure the PCA of a given input space, first, d dimensional mean vector is computed from the input variables, followed by calculating the scatter matrix which results in computing the covariance matrix. Next, by sorting eigen-vector and Eigenvalues acquired from the covariance matrix, the components with maximum variance are captured. Using predefined threshold value, k out of p principal components are adopted. In this paper, 0.8 is employed as a k value to hold 80% information of the initial input data. Endorsing this measure, the data is then plotted on k dimensions.

3.3. Information pre-processing

Crash investigation commonly requires dealing with unbalanced data sets due to unequal outcome classes distribution. This sort of imbalances produce a bias toward the majority class, since modeling classifiers prioritize the class with the higher number of instances leading to an over-prediction of this class (Fernández et al., 2009). Pre-processing is designed to tackle this issue by balancing class representations in the data set. In this work, the synthetic minority over-sampling technique (SMOTE) was applied, presented by (Chawla et al., 2002).

SMOTE generates synthetic minority samples in accordance with random intervals between current minority instances instead of duplicating existing minority cases. The technique first finds the k-nearest neighbors of each minority case, following the recommendation of (Chawla et al., 2002). The value of k was set to 5. Next, on the basis of the required over-sampling, multiple iterations are carried out in which one neighbor is randomly chosen from the k-nearest neighbors. Then the difference between the instance in process and its neighbor is computed. This difference is multiplied by a random number between 0 and 1. Lastly, the new synthetic instances are included in the data set and appointed to the minority class. In contrast with the random oversampling technique which randomly duplicates the original minority instances to rebalance the input space (Haibo and Garcia, 2009). SMOTE redress balance without data replication, as such, the over-fitting issue can be prevented (Fernández et al., 2009; Gao et al., 2011). Typically, the renown and effectiveness of SMOTE methodology emanate from three foundations: computational efficacy, straightforwardness and notable performance (Haibo and Garcia, 2009; Sun et al. 2009). In order to increase the base classifiers' performance, the SMOTE technique is applied to handle imbalance issue by oversampling only on the training data, then evaluate the models on the stratified but non-transformed test set in order to ensure that the test set does not contain "synthetic" examples. Fig. 3 depicts an illustration of the SMOTE algorithm.

3.4. Multi-classifier system (MCS)

The multiple classifier system (MCS) is a machine learning technique that has been proven to give higher performance results comparing to using single classifiers. The main issue with constructing a MCS is the way classifiers are selected to form an ensemble, and how to fuse the individual decisions of the base classifiers into a single decision; In practice, three requirements are critical for building an effective MCS model: performance of base classifiers, variety among classifiers, and the selection of the fusion method (meta-classifier) that will be utilized. The goal of this fusion scheme is to gain improved performance with proper single classifiers and to exclude the uncorrelated classifier-specific errors (Tsai et al. 2014). The MCS is typically structured into three major steps (Britto et al., 2014); generation, selection, and fusion discussed as follows.

3.4.1. Classifier generation

The generation stage aims to produce a set of base classifiers contains the most competent candidates for the ensuing classifier choosing and integration phases. Combining decision outcomes of base classifiers in which analysis boundaries are broadly distinct is deemed to be a prime element. To accomplish this, establishing variety among classifiers is needed. A dissimilarity methodology focuses on training the base classifiers on diverse input partial spaces in order to generate a collection of distinct but complementary

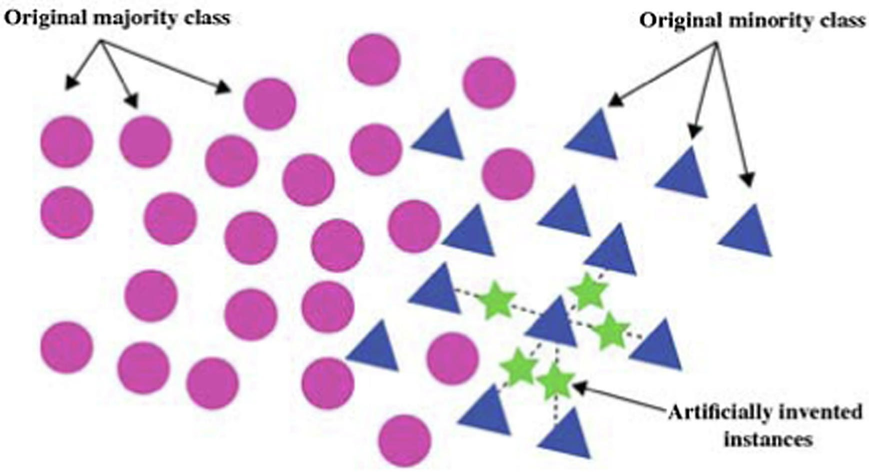

Fig. 3. Illustration of the SMOTE procedure.

classifiers. To establish variations in classifiers' training dataset, two methodologies are adopted: Bagging and Boosting, each of which is described in detail below.

3.4.1.1. Bagging. Bootstrap aggregation, simply known as Bagging, is an ensemble-based technique and one of the most intuitive and straightforward methods to develop with remarkably high performance. It's designed to enhance the stability of machine learning algorithms as it minimizes the variance of the model, without raising the bias. In the Bagging approach, the classifier is trained on different training datasets that are created by bootstrap method (Breiman, 1996). Assuming that a training set includes N instances while C denotes the class outcome such that $\{c_1, \cdots, c_n\}$ its potentially states. Bagging constructs a set of m classifiers. For each one of them, the original given input space is randomly resampled with replacement into a bootstrapped replica, these formed samples are independent and used for the classifier training procedure. When the training is done, the final results are then aggregated via a proper technique, such as majority voting. Pseudo code for the Bagging method is provided as follows:

Algorithm (*Bagging*).

Input:
Training dataset T with labels $c_i \in \phi = \{c_1, \cdots, c_n\}$ representing N possible classes
Base learning algorithm L
Integer R specifying number of learning rounds

Process:
For r = 1,2, …, R:
$T_r = Bootsrap(T)$ % Generate a bootstrapped replica randomly from T
$h_r = L(T_r)$ % Train a base learner from the bootstrapped sample
end

Output:
$H(x) = argmax_{c \in C} \sum_{r=1}^{R} l(c = h_r(x))$% the value of $l(\alpha)$ is 1 if α is true
% and 0 otherwise

3.4.1.2. Boosting. Boosting is a highly effective and commonly used ensemble classifier. Unlike bagging, each individual base learner in boosting is trained on different training sets in a sequential way. Boosting generates base learner: the first one, L_1, is trained by a random sample of the training data. As for the second one L_2, half of its training data is correctly classified by L_1 and the other half is misclassified by L_1. Finally, the third classifier $L_3$is trained on samples that are misclassified by both L_1 and L_2. The $w_l = \{w_l^1, \cdots, w_l^n\}$ depicts the weight distribution over tuples in iteration r and is equally allocated in the first iteration. In each iteration r, boosting maintains the weights on the training sample i designated as w_r^i so that the learner algorithm will watch out for the wrongly classified instances in the training set in subsequent iterations. The base learning algorithm finds the most appropriate classifier h_r and assigns an importance measure $\alpha_r$to it, set as:

$$\alpha_r = \frac{1}{2}\ln\left(\frac{1 - e_r}{e_r}\right)$$

where e_r represents the mean squared error (MSE) for h_r. Even though there are various boosting algorithms, we adopt the AdaBoost method presented by (Freund and Schapire, 1997), which is the most popular boosting algorithm. Pseudo code for the Bagging method is provided as follows:

Algorithm (*AdaBoost*).

Input:
Training dataset T with labels $c_i \in \phi = \{c_1, \cdots, c_n\}$ representing N possible classes
Base learning algorithm L
Integer R specifying number of learning rounds

Process:
$w_l = \{w_l^1, \cdots, w_l^n\}$For r = 1,2, …, R:
Determine the weight allocation w_r
Train a base learner from T using distribution T_r: $h_r = L(T, T_r)$
Compute: $e_r = MSE$ for h_r and $\alpha_r = \frac{1}{2}\ln\left(\frac{1-e_r}{e_r}\right)$
Update the distribution weight set: $w_{r+1}^i = normlize(w_r^i * e^{-\alpha_r})$
end

Output:
The combined learner $H = \sum_{r=1}^{R} \alpha_r h_r$

3.4.2. Classifier selection

On top of dissimilarity in handling the dataset, the performance of the base learners is another substantial factor in classifier selection and can ensure the successful construction of the MCS. The selection stage aims to identify group of learners (fusion set) that enhance the classification performance in fusion. Several selection criteria have been suggested in the literature. The individual performance is a universal metric for selecting the most effective classifiers as it has proven to be reliable and efficient (Ruta and Gabrys, 2005; Thomas et al., 2018). A variety of frequently used performance measures are used to evaluate the quality of the classification models. Recall, also known as true positive rate (TPR) or sensitivity, is defined as the proportion of correctly classified positives (i.e. crash events correctly classified). Since the primary goals focus is to correctly predict the rare events of the accident class, recall is a particularly substantial metric of classifier performance in this case. Precision on the other hand is a measure of accuracy outlining the relevance ratio of the predicted instances, i.e. percentage of truly predicted events from all predicted events.

$$Recall = \frac{TP}{TP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

where the True Positive (TP) indicates the number of crash occurrences correctly classified, and False Positive (FP) indicates the number of non-crash events incorrectly classified as crash-events. False Negative (FN) indicates the number of crash events incorrectly classified as non-crash events, and True Negative (TN) indicates the number of non-crash occurrences correctly classified. F1 score, is a highly informative measure as it considers both precision and recall measures, which makes it very suitable for imbalanced classification (Qian et al., 2014; Sun et al., 2018); it's deemed to be a special measure that conveys the balance between the precision and recall in order to find an effective and efficient trade-off. Another useful metric is G-mean, which is considered as a metric of stability between correct classification of positive class and negative class viewed independently. It is usually adopted in order to resist the imbalances in the dataset (Kubat et al., 1997). As this is a class imbalance problem, both F1 score and G-mean have been calculated to evaluate base classifiers, they are described as follows:

$$F1\ \ score = 2 * \frac{precision * recall}{precision + recall}$$

$$Gmean = \sqrt{\frac{TP}{TP + FN} * \frac{TN}{FP + TN}}$$

First, a 10-fold cross-validation is conducted to produce diversity among base learners. K-fold cross-validation has been recognized for its susceptibility to yield minimal bias and variance in contrast with the other validation methods, including the leave-one-out method, and has been widely adopted in the evaluation of classification performance (Kohavi and Kohavi, 1995). In order to increase the base classifiers' performance, the SMOTE technique is applied to handle imbalance issue and prevent over-fitting by oversampling only on the training data, The proper way to apply rebalancing strategies is to address the imbalance issue by oversampling only the training set while the test set is left intact, (Elamrani Abou Elassad et al., 2020b; Makond et al., 2015; Ramentol et al., 2012; Wang et al., 2014), this way none of the information in the test data is being used to create synthetic observations making the evaluation more realistic. The SMOTE technique is applied within every cross-validation round; the models are trained on the redressed training fold afterwards, then, the final performance of a classifier corresponds to the average over the cross-validation iterations. All potential sets of learners are built then acquired by the fusion method along with their output decisions and ground truth of the instances. For each group of classifiers, the F1 score and G-mean metrics are computed. If they are higher than 80%, the corresponding set is picked; if not, it will be taken away from the set of learners. The elaborated methodology adopts two of highly powerful performance measures to ensure an imbalance-aware strategy, and also considers diversity by endorsing cross-validation in order to generate variations in the training dataset, thus it analyzes the effectiveness of the predictor and reduces the overlapping of input space partition. The complete selection stage is outlined as follows:

Classifier selection algorithm	
Input:	
	Pool of base learners: $P = \{l_1, \cdots, l_n\}$ Training dataset: *Train_set* Testing dataset: *Test_set*
Process:	

	$Set_{learners} = \varnothing$% Initialize the set of selected classifiers
	Apply 10-fold cross validation on *Train_set* – *[*Fig. 5 *(Step 4)]*
	
	For each $l_i \in P$ do:
	Perform SMOTE technique within each training fold – *[*Fig. 5 *(Step 4)]*
	Train l_i by the redressed training fold – *[*Fig. 5 *(Step 4 & 5)]*
	End for
	
	For k = 1: C_j^N do:
	Create Set_k with j learners taken from P without replacement*) – [Best learners based on* Tables 2,3,4 & 5*]*
	Apply 10-fold cross validation on *Test_set* – *[*Fig. 5 *(Step 6)]*
	Generate the Set_k learners' outputs and ground truth – *[*Fig. 5 *(Step 7)]*
	Pass them down to the fusion method (Meta-classifier) – *[*Fig. 5 *(Step 8)]*
	if (Average F1 score and Average G-mean) ≥ 80% - *[*Fig. 5 *(Step 9 & 10)]*
	$Set_{learners} = Set_{learners} \cup Set_k$ - *[*Fig. 5 *(Step 11 & 12)]*
	End if
	End for

Output:	
	Set of selected base learners: $Set_{learners}$ – *[*Fig. 5 *(Step 12)* + Table 6*]*

3.4.3. Fusion design

Various diversity techniques were presented for establishing and training of classifiers during the generation stage. SMOTE and cross-validation were used in the selection step and the additional algorithms bagging and boosting will be adopted in this final fusion stage which is expected to minimize the overall error estimates and improve classification performance. The results of the base classifiers have been fused using a meta-classifier trained on top of the different learners' outputs. The underlying concept is that the meta-classifier learns how properly the classifiers train data by combining the predictions in order to detect patterns and enhance prediction efficiency. Suppose a set of base classifiers Set_k combining learners $l_1, \cdots, l_n$. Every l base learner of a set of classifiers Set_k is trained with i^{th} feature tuple $X_i = \{\varphi_1, \cdots, \varphi_f\}$ where φ_m is the m^{th} feature of the vector X_i. Afterwards, the classifiers are tested and their outcome predictions are obtained and denoted as $y_i^l = f_{Set_k}(X_i)$. Individual predictions by the classifiers along with the ground truth G_i of the i^{th} sample form the vector Y_i. The final prediction for i^{th} tuple is expressed as follows:

$$P_i = \mathscr{L}(Y_i, G_i) \quad i = 1, 2, \cdots, I$$

Fig. 4 shows the generalized process of the fusion stage. Given an appropriate meta-classifier and ensemble of base classifiers, the samples utilized to train the base learners should not be used to train the meta-classifier so that overfitting can be prevented. Accordingly, at large, the base classifiers are trained solely adopting the training input space and the meta-classifier is trained based on the validation input space (Rokach, 2010). The detailed fusion framework scheme of crash prediction is illustrated in Fig. 5.

4. Experiment design

4.1. Experiment setup and subjects

A total of 113 participants (86 males, 27 female) between the ages of 20 and 42 (M = 27.5; SD = 1.15) volunteered to participate in this experiment. All participants had a full driver's license and had been driving for at least a year. Average years of driving experience ranged from 1 to 24 years (M = 11.75; SD = 3.64). All were in a good shape, and had (corrected to) normal vision. With respect to the given information about the study's general intentions, all participants gave informed consent form about data recording of their driving data. The study was carried out using a fixed-based driving simulator located at the University of Cadi Ayyad (UCA) facility. Simulator driving studies confer a significant advantage of imitating conduct in a safe environment with a full

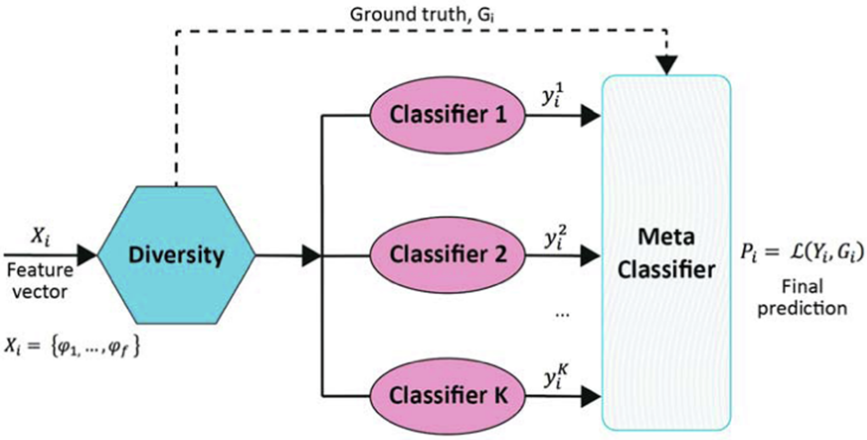

Fig. 4. Generalized process of the fusion stage.

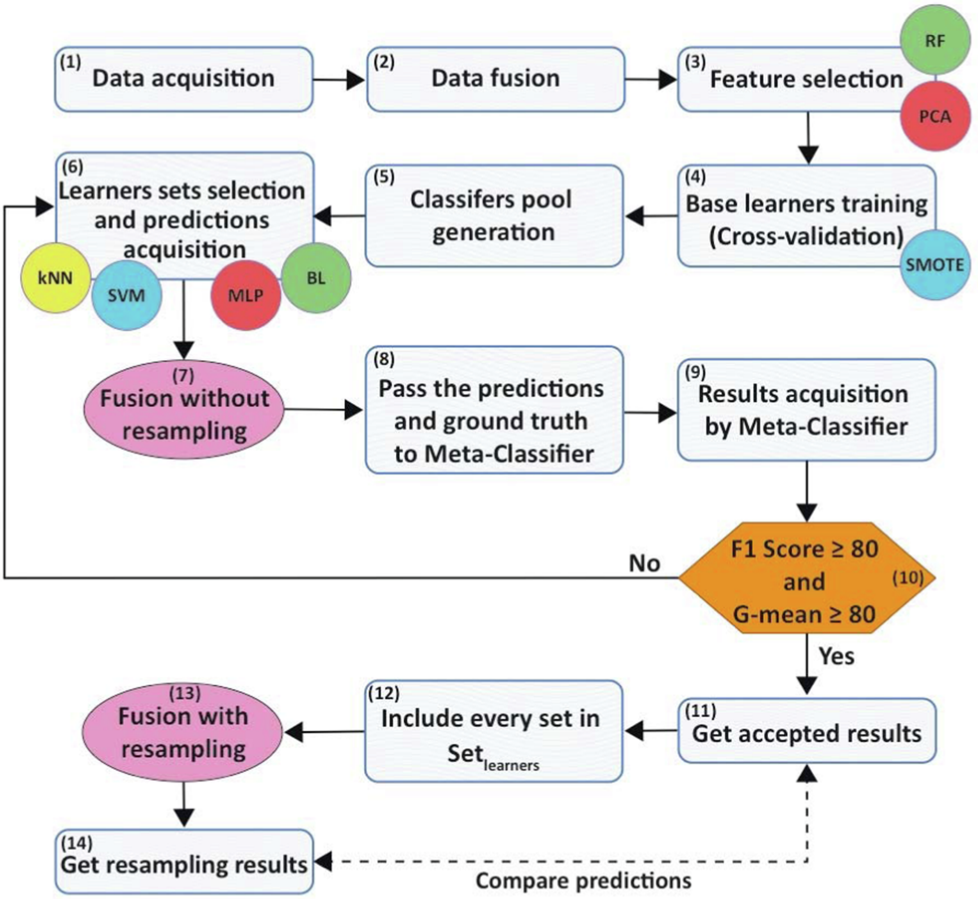

Fig. 5. Comprehensive illustration of the proposed fusion framework.

empirical control over driving conditions including all types of weather, ground, and traffic (Elamrani Abou Elassad and Mousannif, 2019). The driving simulation was run through the Project Cars 2 simulator by (Slightly Mad Studios) using a DELL XPS running on Windows 10. Computational calculations were performed on the same platform and on a 2015 MacBook Pro with an i7 2.8 GHz chip, 16 GB RAM and SSD hard drive. Participants viewed the simulation on a 27-inch LCD monitor with a resolution of 1920 × 1080 pixels, and heard auditory via a surround speaker system. The computer was fitted with state-of-the-art driving simulator driving force GT27 Logitech® incorporating Racing Wheel set (steering wheel, accelerator pedal, and brake pedal) with the adjustable

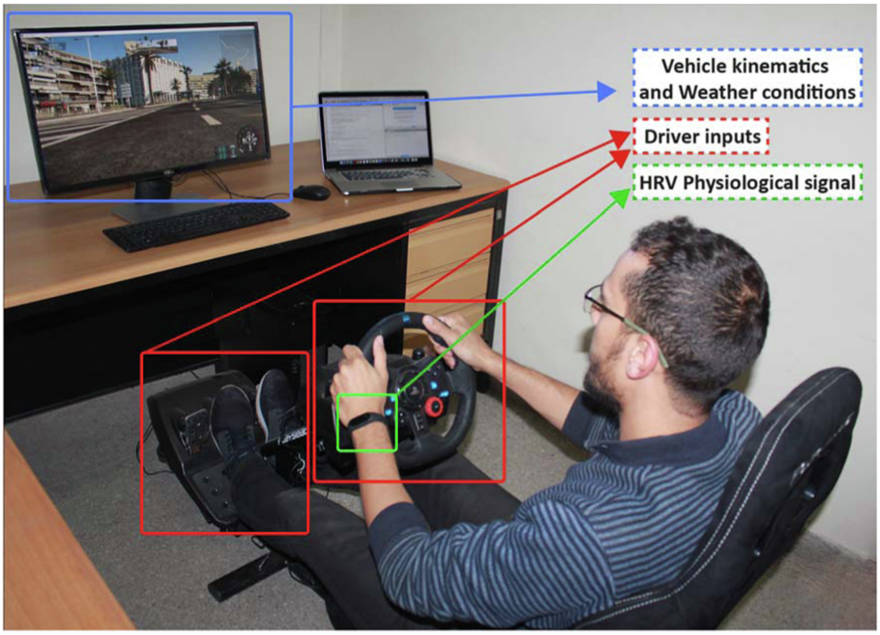

Fig. 6. Experimental setup of the desktop driving simulator at UCA.

Logitech Evolution® Playseat. Simulations were conducted with automatic transmission mode; thus, gear shifter was not needed, and had a 900-degree wheel rotation with advanced force feedback mechanism to make driving experience as realistic as possible. Fig. 6 illustrates the hardware setup.

4.2. Procedure and data acquisition

The drivers were given essential information about the experimental procedure before they were appointed to a quiet laboratory to virtually drive the vehicle; each participant navigated experimental session drives on a virtual two-lane urban road of 23.75 km length which required about 16 min to complete when the speed limits are kept. The driving scenario was performed in daylight and under three different sequential weather conditions: clear, heavy fog, heavy rain that can be seen is Fig. 7, and aimed to simulate various intricacies and aspects that real-world driving entails in order to explore the impact of the factors on driving behavior and to collect enough raw data before the crash. The adopted protocol had similar traffic conditions and the identical number of outer events for all participants. Upon arrival, participants read through and signed an informed consent form to indicate their agreement to participate in the experiment and completed a questionnaire assessing their demographic characteristics and recent activities. The experimental session consisted of two separate visits to the simulator where drivers were instructed to drive as they usually do in a real driving situation and follow the traffic rules. The first visit was a 15-minutes practice drive for the participants to familiarize themselves with the simulator, whereas in the second visit which is the main trail, drivers navigated the vehicle in a highway road consisting of four different virtual weather factors along with several hazard scenarios located along the route such as a surrounding road user (e.g. another vehicle). This set of weather and terrain characteristics serve to provide various levels of difficulty while maneuvering the vehicle along the driving route. Within this context, when developing the study, the effects of repeated crash-imminent events within a single experimental session were indeed a concern. These effects could result in drivers anticipating crash events and driving more cautiously. The virtual simulator minimized these effects by randomly generating dissimilar driving situations and by including many outer events where it appeared that a road user would pose a threat but ultimately did not promoting the participants to scan the layout more thoroughly and drive cautiously. Therefore, some outer events generated by the simulator were designed to imitate possibly crash conditions in which an event did not materialize, thereby making the driving process less predictable. On another note, the adopted simulation software controlled the generated road users in a way that some of them were not necessarily visible the entire time. In some scenarios, external vehicles are hidden behind representative buildings and landmarks influencing the participant's sight view, then the systems vehicles are triggered to begin moving in order to become visible for the subject's vehicle, as a result, the participant would not be able to observe the incursion from the start. Post-drive interviews revealed that the driving scenario was highly realistic and the participants did not feel like they were experiencing repeated conduct situations nor driving an obstacle course.

Data were continuously recorded throughout each drive with a sampling frequency of 60 Hz through. The simulator collects, through UDP protocol, records of driver inputs (e.g. throttle/brake pedal position, steering wheel position), vehicle dynamics (e.g. speed, yaw angle), weather condition at the time of the accident namely clear, heavy fog or heavy rain. Furthermore, a MI band was used to record the HRV signal. The band was paired with a Xiaomi Android Mobile device running an application implemented for

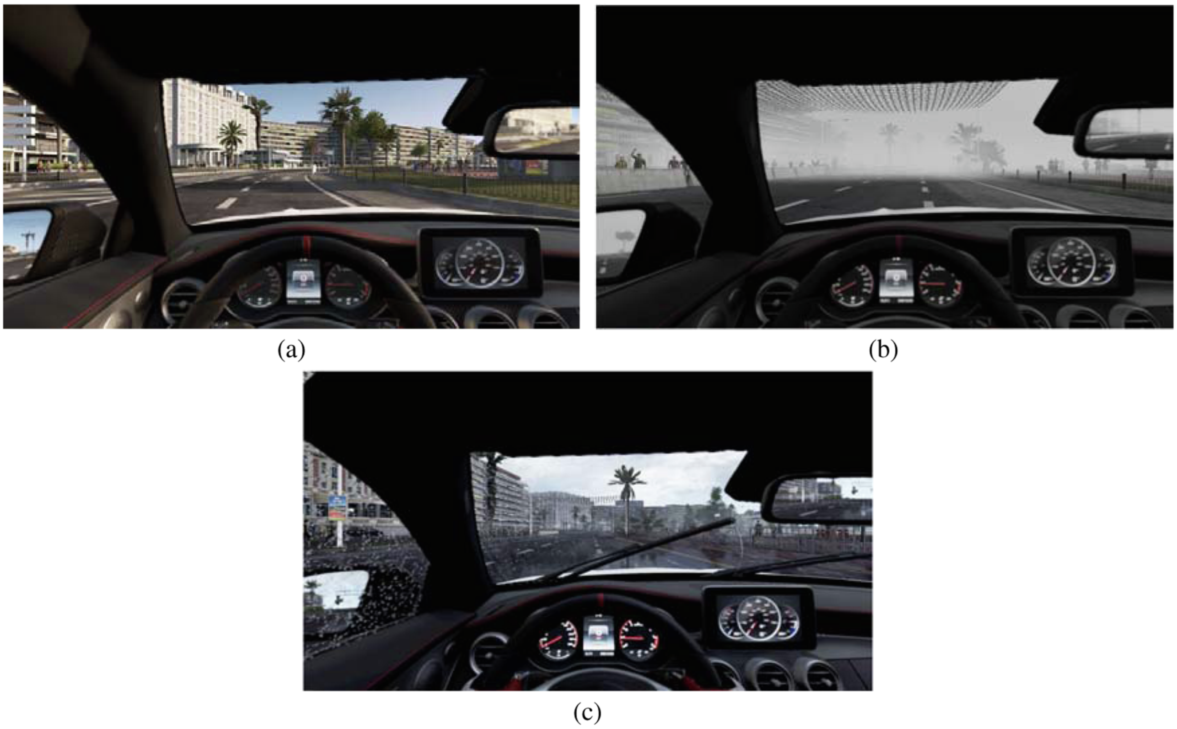

Fig. 7. Weather patterns for driving scenarios: (a) clear, (b) heavy fog, (c) heavy rain.

the experiment which recorded the HRV data. Time to Collision (TTC) to the nearest road users was computed and included within the vehicle kinematics. The TTC feature value was calculated on the basis of the method presented in the work of (Ward et al., 2015), which involved multiple contexts of unconstrained vehicle activity. These are four categories of metrics that may affect traffic safety. Table 1 summarizes the grouping and definitions for all the features acquired during the driving simulations. The dependent variable is crash occurrence, coded as a binary variable with a value of 1 if a crash was identified and 0 if not. Apart from the categorical feature of weather season, all variables are with continuous values, some of which are graphically displayed in Fig. 8. A thorough and comprehensive data screening that includes cleaning and consistency checks is executed to secure data operability and validity for the analysis.

In general, if the original data of the input vector possess a high degree of dispersion, larger parameter values will occupy the learning process of the adopted models which is likely to affect the prediction performance (Akbari et al., 2012), though a data normalization operation has been applied to transform our dataset into a normal distribution in order to reduce data variability, whereas missing values were replaced by the respective mean. During the trials, participants who had reported simulator sickness during either the training or experimental sessions were stopped immediately, and their driving records were excluded in dataset. In total, 1026 samples were extracted from the raw information. During the driving trials, 101 crashes were committed by 97 drivers out of 113 subjects. Only four drivers experienced more than one crash. With Reference to the previous research of the intervention time (Werneke and Vollrath, 2013; Yan et al., 2015), we retrieved the 12 s length data segments, from 16 s to 4 s prior to the crashes, as the crash data to validate the patness of the suggested prediction strategy. In parallel, 925 information segments of 12 s duration were randomly extracted from all 113 drivers' raw data as the non-crash observations, which did not overlap with the any crash instances.

5. Experimental results

In crash prediction, statistical techniques have been found to suffer from low data standards and require large extent of historical data (Wang et al., 2019a). Another limitation of statistical analysis is due to the underperformance in handling features with a high number of categories (Tabachnick and Fidell, 2013). In reaction to these limitations, machine learning models have demonstrated to outperform statistical analysis in predicting forthcoming events and have stated favorable outcomes in many transportation systems (Lee et al., 2018; Schlögl et al., 2019). In this work, Multilayer Perceptron (MLP), Support Vector Machines (SVM) and Bayesian Network algorithms are used in assessing crash collisions. Artificial neural networks (ANN) are abstract computational techniques inspired by biological neural networks and are efficient and applicable for predicting the relation between dependent and independent parameters. The performance of ANN prediction is highly affected by its structure which is comprised of an input layer, hidden layers and an output layer (Basheer and Hajmeer, 2000). MLP is one of the types of ANN. In training, forward propagation and backward propagation are used repeatedly to calibrate all the network weights. As for SVM, a technique developed by (Vapnik, 1995), is a non-probabilistic binary linear classifier that can be used to solve a classification problem by constructing optimal

Table 1
Definitions of all the acquired features during driving simulations.

Category	Feature	Definition
Vehicle Kinematics	Speed	Magnitude of vehicle's velocity.
	Lateral Gravity	Lateral gravitational force
	Longitudinal Gravity	Longitudinal gravitational force
	Vertical Gravity	Vertical gravitational force
	Yaw Angle	Angle between a vehicle's longitudinal axis and its line of travel.
	Drift Angle	Angle during a turn between car's orientation and its velocity direction.
	Spin Angle	Angle between the direction in which a wheel is pointing and the direction in which it is actually traveling.
	Revolutions Per Minute (RPM)	Number of rotations of the vehicle's engine crankshaft.
	Tires Temperature	Overall temperature of vehicle's tires
	Oil Temperature	Temperature of the engine's oil
	Oil Pressure	Pressure of the engine's oil
	Water Temperature	Vehicle's water temperature
	Water Pressure	Vehicle's water pressure
	Fuel level	Vehicle's fuel level
	Elevation	Vehicle's elevation measure from the road surface
	Driven Distance	Vehicle's traveling distance
	Time-To-Collision	The time it would take to reach the given collision point.
	Tyre wear FL	Front left tyre wear
	Tyre wear FR	Front right tyre wear
	Tyre wear RL	Rear left tyre wear
	Tyre wear RR	Rear right tyre wear
	Tyre slip FL	Relative motion between the front left tyre and the road surface.
	Tyre slip FR	Relative motion between the front right tyre and the road surface.
	Tyre slip RL	Relative motion between the rear left tyre and the road surface.
	Tyre slip RR	Relative motion between the rear right tyre and the road surface.
	Brakes Temperature	Overall temperature of vehicle's brakes.
	Type	Vehicle's type
Driver Inputs	Throttle	Accelerator pedal position
	Brake	Brake pedal position
	Steering	Steering wheel angle
	Clutch	Clutch pedal position
	Gear	Gear shift position
Environmental Conditions	Weather Season	(1) Clear, (2) Heavy Fog and (3) Heavy Rain
	Track Temperature	Road surface temperature.
	Ambient Temperature	Air temperature of the environment.
	Road Type	Rural or Urban
Physiological Signal	Heart Rate Variability	Indicative of autonomic regulation of the heart rate

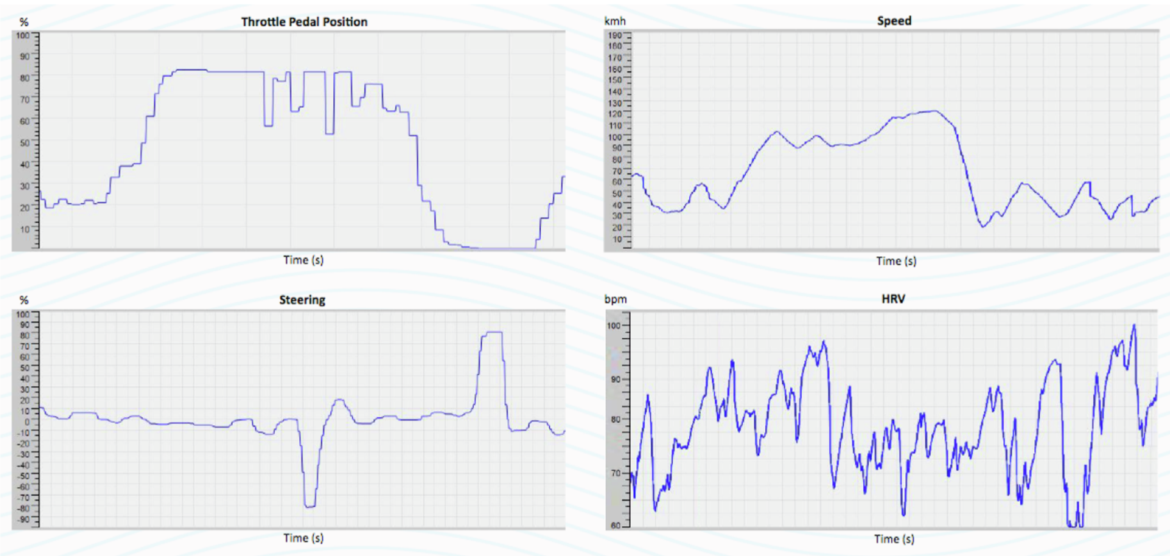

Fig. 8. Graphical visualization of raw data readings.

separating hyperplane in a manner that the margin is maximized so that SVM has good generalization ability. SVM has been proven to be an efficient and robust algorithm for binary classification problems, and it has been found to demonstrate proportionate or superior performance than other statistical and machine learning methods (Chih-Wei et al., 2003; Kecman, 2005). The k-nearest neighbor (k-NN) is a simple and effective technique for objects classification based on the nearest training examples in the input space (Altman, 1992). It is a non-parametric classification technique, that is employed when there is little or no prior knowledge about the distribution of the input space; it transforms instances to a metric space the distance function between a validation tuple and the learning tuples is computed, then a test instance with reference the most common class in its k-nearest training samples is classified. Lastly, BL which is are probabilistic graphical model, consisting of nodes and arcs, where nodes represent the variables in the system and arcs represent the conditional probability distributions between nodes derived from empirical data in the form of a directed acyclic graph, have illustrated robust effectiveness in assembling information across complex networks in many application domains, such as data fusion and engineering decision-making (Dahll, 2000), and enable examining the marginal and conditional dependencies for the issue as provided by the existing data (Koller and Friedman, 2009). In this study, several Bayesian learners such as Naïve Bayes, Augmented Bayesian Networks, Tree Augmented Naive-Bayes and others have been utilized for crash analysis.

The data have been preprocessed first, then, the aforementioned classifiers are adopted individually and combined in order to grasp the effectiveness in establishing the MCS. Crash prediction have been thoroughly examined taking into account several aspects, namely, information heterogeneity, classifier type, variable selection, and the effect of pre-processing procedure. That is, normalization was applied to scale the data within a certain range to minimize bias and to ensure that they receive equal attention within the base learners. The weather categorical feature has been mapped using binary encoding, where each feature is represented by n-features with n is the number of categories the feature can have. Also, a ratio between 1% and 2% of existing missing data have been imputed to overcome the influence of ignoring instances. On another note, two feature selection techniques (PCA and RF) were adopted for feature selection in this study. As revealed in Tables 2–5, the effect of the preprocessing using these approaches is obvious in the result since all classifiers depicted inferior performance without using feature selection, while when employing the appropriate technique, the classification effectiveness improves significantly. The best performance is achieved using the RF adopted as feature selection to identify the crash strongest precursors, the selected variables are displayed in Fig. 9.

Table 2 displays the averaged classification performance of various BL. Lazy Bayesian Rules was the least effective with 66.15% F1 score and 67.84% G-mean, followed by Naïve Bayes, while the best results were given using Bayesian Networks and Augmented Bayesian. Table 3 shows the averaged classification performance of SVM using multiple kernels. During the parameterization process there are four parameters which should be carefully adjusted. The *C* parameter, known as a regularization parameter or penalty factor, is set to avoid models over-fitting, and adjusts the trade-off between training errors and margins. For RBF, Sigmoid and Linear kernels, the degree of non-linearity is controlled by the gamma γ parameter while for the Polynomial and Sigmoid kernels, *d* is the polynomial degree and *r* is the bias term. Whereas the tolerance hyper-parameter ε determines the margin within which the error is neglected. Maximum classification F1 score of 83.26% and G-mean of 82.90% is obtained using RBF kernel. Table 4 depicts the averaged outcomes of kNN. The best results were 78.17% of F1 score and 79.30% of G-mean observed using 5 as number of neighbors. Tables 5, shows the averaged prediction performance of MLP using different number of hidden layers. Maximum F1 score of 82.87% and G-mean of 83.19% are achieved using ten hidden layers. In accordance, to define fusion methods, the algorithms that depict higher prediction performance are chosen as meta-classifiers. The optimal configuration for every classifier has been selected to be used in the next stage.

The average F1 score and G-mean are calculated for learners sets with 4 classifiers while conducting 10-fold cross-validation. The chosen sets are identified as the best ensembles whose performance are above the specified threshold and then adopted in the fusion scenario with diversity techniques Bagging and Boosting. Table 6 shows the performance results for each set, the results reveal that although some of algorithms did not individually display a high performance (e.g., Naïve Bayes and Lazy Bayesian Rules), they could

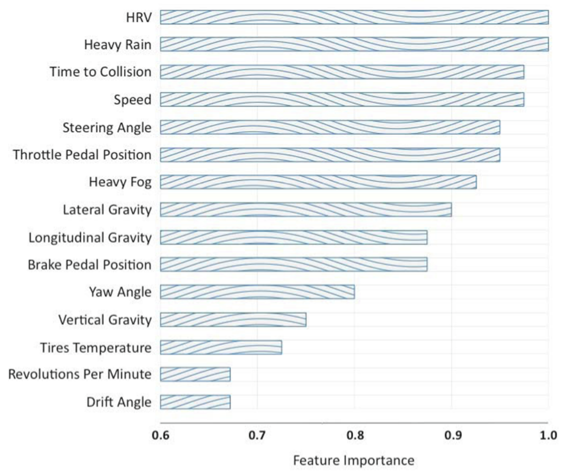

Fig. 9. Feature selection illustration using Random forest.

Table 2
Bayesian learners crash prediction performance.

Classifier	Without feature selection		With PCA		With RF	
	F1 score (%)	G-mean (%)	F1 score (%)	G-mean (%)	F1 score (%)	G-mean (%)
Naïve Bayes (NB)	67.23	69.03	69.80	70.51	70.67	71.18
Bayesian Networks (BN)	75.17	75.92	78.77	77.80	80.32	81.65
Augmented Bayesian Networks (ABN)	76.10	78.05	78.50	79.55	81.03	81.79
Tree Augmented Naive-Bayes (TAN)	75.21	76.07	77.02	76.13	77.38	78.07
Lazy Bayesian Rules (LBR)	62.36	62.42	65.19	64.94	66.15	67.84

Table 3
SVM crash prediction performance.

SVM kernel	Parameters' values					Without feature selection		With PCA		With RF	
	C	γ	r	d	ε	F1 score (%)	G-mean (%)	F1 score (%)	G-mean (%)	F1 score (%)	G-mean (%)
Linear	14	–	–	–	0.03	79.02	80.00	81.11	80.10	81.17	81.02
Polynomial	8	0.07	0.93	5	0.05	74.53	74.10	76.96	77.20	78.69	78.10
RBF	6	0.04	–	–	0.08	80.20	80.83	82.16	82.55	83.26	82.90
Sigmoid	11	0.09	0.08	–	0.15	76.50	76.30	77.09	76.37	80.05	78.43

Table 4
kNN crash prediction performance.

k	Without feature selection		With PCA		With RF	
	F1 score (%)	G-mean (%)	F1 score (%)	G-mean (%)	F1 score (%)	G-mean (%)
1	51.19	53.50	53.90	56.76	56.23	60.08
3	64.30	64.47	67.05	64.32	68.11	66.34
5	73.00	73.83	75.40	74.11	78.17	79.30
10	70.09	69.25	73.33	72.66	74.20	74.46

Table 5
MLP crash prediction performance.

Hidden layers	Hidden neurons	Without feature selection		With PCA		With RF	
		F1 score (%)	G-mean (%)	F1 score (%)	F1 score (%)	G-mean (%)	F1 score (%)
2	5	55.43	57.28	58.03	61.11	61.55	64.44
5	5	67.30	66.00	72.09	72.15	76.02	76.00
10	5	78.80	80.17	81.10	81.50	82.87	83.19
15	5	70.70	68.55	75.71	74.05	78.79	77.63

achieve an adequate level of performance in an ensemble model. In the last stage, two resampling techniques (Bagging and AdaBoost) are conducted separately for each selected set of learners (Table 6), and the final results for crash predictions are depicted in Table7. The highest prediction performance is evaluated as 93.56% and 94.86% for F1 score and G-mean respectively, which is achieved by the kNN, SVM, Bayesian Networks and Naïve Bayes with MLP as the fusion algorithm and AdaBoost as the resampling technique. In accordance of results in Table 7, AdaBoost broadly demonstrated effective performance in increasing the prediction measurement metrics of ensembles in comparison with Bagging; Still, the performance of some ensembles with weak base learners was not successfully enhanced using the resampling approaches. As can be seen, almost all the findings depict performance over 80% for both F1 score and G-mean. Moreover, the highest results (performance over 90%) have been acquired once as SVM meta-classifier in the combination of MLP, kNN, Augmented Bayesian Networks and Bayesian Networks for both Bagging and AdaBoost, and also when using MLP as meta-classifier for SVM, Naïve Bayes, Tree Augmented Naive-Bayes and Bayesian Networks with AdaBoost as a resampling technique and with Bagging for kNN, Naïve Bayes, Bayesian Networks and SVM. We have also provided in Table 8 the detailed statistics of F1 score, G-mean as well as AUC values for each selected set with resampling to deeply investigate the estimation performance for crash prediction.

Fig. 10 is plotted to further outline the performance with respect to the adopted resampling techniques (Bagging and AdaBoost) measured in F1 score, G-mean and Accuracy for both crash and non-crash events. As evidenced, when it comes to crash prediction, the performance based on Bagging depicts higher values with medians around 85% for all the performance metrics; In terms of AdaBoost, F1 score and Accuracy exhibited superior results with medians around 87% whereas G-mean achieved a median value of

Table 6
Average F1 score and G-mean of selected sets for crash prediction (without resampling).

Set ID	Base learners				Fusion method	F1 score (%)	G-mean (%)
1	Augmented Bayesian Networks	Tree Augmented Naive-Bayes	SVM	kNN	Bayesian Networks	80.11	82.22
2	Tree Augmented Naive-Bayes	Naïve Bayes	SVM	MLP	Bayesian Networks	84.37	84.89
3	MLP	kNN	SVM	Augmented Bayesian Networks	Bayesian Networks	83.08	84.02
4	kNN	SVM	Bayesian Networks	Naïve Bayes	Augmented Bayesian Networks	81.55	81.00
5	Bayesian Networks	Lazy Bayesian Rules	MLP	SVM	Augmented Bayesian Networks	82.22	84.19
6	MLP	Bayesian Networks	kNN	Tree Augmented Naive-Bayes	SVM	85.10	84.64
7	kNN	Naïve Bayes	Augmented Bayesian Networks	Lazy Bayesian Rules	SVM	80.90	81.55
8	MLP	Naïve Bayes	Tree Augmented Naive-Bayes	Bayesian Networks	SVM	84.33	86.20
9	MLP	kNN	Augmented Bayesian Networks	Bayesian Networks	SVM	85.78	85.96
10	SVM	Naïve Bayes	kNN	Bayesian Networks	MLP	85.66	83.37
11	Bayesian Networks	Tree Augmented Naive-Bayes	Augmented Bayesian Networks	kNN	MLP	83.19	82.60
12	kNN	Naïve Bayes	Bayesian Networks	SVM	MLP	85.80	82.75

Table 7
Average F1 score and G-mean of selected sets for crash prediction (with resampling).

Set ID	Base learners				Fusion method	Resampling Technique	F1 score (%)	G-mean (%)
1	Augmented Bayesian Networks	Tree Augmented Naive-Bayes	SVM	kNN	Bayesian Networks	Bagging	81.57	82.72
						AdaBoost	83.29	84.12
2	Tree Augmented Naive-Bayes	Naïve Bayes	SVM	MLP	Bayesian Networks	Bagging	79.73	80.03
						AdaBoost	85.79	87.50
3	MLP	kNN	SVM	Augmented Bayesian Networks	Bayesian Networks	Bagging	85.16	85.11
						AdaBoost	88.10	87.44
4	kNN	SVM	Bayesian Networks	Naïve Bayes	Augmented Bayesian Networks	Bagging	80.72	80.06
						AdaBoost	81.82	83.31
5	Bayesian Networks	Lazy Bayesian Rules	MLP	SVM	Augmented Bayesian Networks	Bagging	87.08	88.20
						AdaBoost	86.65	85.92
6	MLP	Bayesian Networks	kNN	Tree Augmented Naive-Bayes	SVM	Bagging	85.10	84.64
						AdaBoost	87.02	86.77
7	kNN	Naïve Bayes	Augmented Bayesian Networks	Lazy Bayesian Rules	SVM	Bagging	80.09	80.33
						AdaBoost	80.10	80.40
8	MLP	Naïve Bayes	Tree Augmented Naive-Bayes	Bayesian Networks	SVM	Bagging	84.37	86.56
						AdaBoost	83.72	83.22
9	MLP	kNN	Augmented Bayesian Networks	Bayesian Networks	SVM	Bagging	88.80	89.90
						AdaBoost	91.34	92.07
10	SVM	Naïve Bayes	Tree Augmented Naive-Bayes	Bayesian Networks	MLP	Bagging	85.50	85.55
						AdaBoost	90.19	91.92
11	Bayesian Networks	Tree Augmented Naive-Bayes	Augmented Bayesian Networks	kNN	MLP	Bagging	84.51	83.68
						AdaBoost	87.97	85.06
12	kNN	Naïve Bayes	Bayesian Networks	SVM	MLP	Bagging	90.08	92.70
						AdaBoost	93.56	94.81

86%. The worst performance is obtained when none of the resampling methods is performed with median values of 83% for the F1 score, G-mean and Accuracy. In regards to non-crash events, F1 score for both Bagging and AdaBoost, G-mean using Bagging and Accuracy using AdaBoost have achieved higher values with medians around 88%, while G-mean adopting AdaBoost depicted the highest performance with a median over 92%. The least performant results were acquired when the resampling methods are not employed (e.g. 86% for Accuracy). In addition to the aforementioned observations, in contrast to non-crash outcomes, the no-resampling F1 score, G-mean and Accuracy for crash prediction all have their medians nearly in the centers of the boxes, which implies that the values of these models are symmetrically distributed around the medians. Moreover, their values in general have less variation than Bagging and AdaBoost, since they have proportionately shorter boxes and tighter ranges of values.

A comparison between our system and other crash assessment studies is presented in Table 9. Several hybrid models that have been conducted in crash were compared based on modeling metrics including Driver-inputs (DM), Vehicle-based (VM), Physiological-based (PM) and Environmental-based (EM), base learners, feature selection, hybrid model, class imbalance and level of performance in terms of F1 score and G-mean.

6. Discussion

A comparison of the results depicted in Tables 6 and 7 demonstrates the major contribution of adopting AdaBoost as a resampling strategy on performance improvement in crash prediction for the proposed CAS architecture. Apart from the ensemble kNN, Naïve Bayes, Augmented Bayesian Networks and Lazy Bayesian Rules with SVM fusion method, AdaBoost improved the F1 score and G-mean of the selected ensembles. The reason may be the weakness of Naïve Bayes and Lazy Bayesian Rules algorithms (in our dataset) used as base learners, whose effectiveness could not be enhanced even with Boosting.

The findings obtained indicate that the proposed fusion framework utilizing the strongest individual learners has led to more performant predictions, still, this is not always applicable. For instance, the F1 Score and G-mean of LBR are 66.15% and 67.84% respectively, but in fusion with Bayesian Networks, SVM and MLP algorithms, the outcomes improved to 86.65% for F1 score and 85.92% for G-mean based on AdaBoost technique and to 87.08% for F1 score 88.20% for G-mean in Bagging; the same thing goes for the ensemble kNN, Naïve Bayes and Augmented Bayesian Networks where LBR's performance increased to 80.09% for F1 score and 80.33% for G-mean when Bagging is implemented and to 80.10% for F1 score 88.40% for G-mean by means of AdaBoost. One reason for this may be that the impairment of some models can be covered by other learners adopted in the fusion framework which shows complementary performance when they are associated in this way.

Among the different classification models employed in this work, the kNN and Tree Augmented Naive-Bayes expressed different behavior. Although their individual performances for both F1 score and G-mean were improved by the fusion strategy, the results revealed that the fusion system without resampling gave better outcomes. Values of 78.17% and 79.30% for F1 score and G-mean respectively have been found using kNN individually, while Tree Augmented Naive-Bayes resulted in 77.38% F1 score and 78.07% G-mean. However, the ensembles with Naïve Bayes, SVM and MLP resampled by Bagging for Tree Augmented Naive-Bayes led to a

Table 8
Descriptive statistics with resampling of F1-score, G-mean and AUC for crash prediction selected sets.

Fusion method	Base learners	Resampling technique	F1 score (%)				G-mean (%)				AUC (%)			
			Median	Std. dev.	Min	Max	Median	Std. dev.	Min	Max	Median	Std. dev.	Min	Max
BN	ABN + TAN	Bagging	81.55	0.10	71.81	86.02	82.75	0.08	70.10	89.91	73.29	0.01	60.50	79.49
	+	Adaboost	83.32	0.03	70.14	85.55	84.15	0.12	71.59	90.00	77.13	0.51	69.47	86.30
	SVM + kNN													
BN	TAN + NB	Bagging	79.70	0.05	58.50	84.10	79.98	0.01	60.25	83.37	82.10	0.24	70.00	89.78
	+	Adaboost	85.79	0.04	72.10	90.06	87.47	0.01	77.00	91.52	80.09	0.13	65.33	88.50
	SVM + MLP													
BN	MLP + kNN	Bagging	85.19	0.11	70.14	89.76	85.15	0.19	72.23	89.20	76.65	0.08	61.26	84.02
	+	Adaboost	88.08	0.14	80.44	91.20	87.40	0.21	72.62	89.98	87.53	0.05	79.04	92.01
	SVM + ABN													
ABN	kNN + SVM	Bagging	80.70	0.08	69.97	89.34	80.10	0.13	72.31	88.81	55.44	0.12	45.55	67.29
	+	Adaboost	81.83	0.12	73.38	84.66	83.35	0.10	75.18	86.15	69.81	0.20	63.62	76.63
	BN + NB													
ABN	BN + LBR	Bagging	87.01	0.02	79.10	90.09	88.17	0.04	72.39	94.16	79.97	0.01	70.01	90.00
	+	Adaboost	86.62	0.01	78.12	90.01	85.92	0.09	76.61	91.04	79.98	0.01	60.20	91.05
	MLP + SVM													
SVM	MLP + BN	Bagging	85.11	0.04	71.74	89.30	84.65	0.05	75.04	89.09	90.01	0.15	78.91	93.34
	+	Adaboost	86.98	0.06	75.50	96.15	86.79	0.06	76.70	88.51	91.26	0.06	85.35	93.99
	kNN + TAN													
SVM	kNN + NB	Bagging	80.14	0.17	71.11	88.22	80.30	0.11	68.18	87.72	83.73	0.03	73.00	89.87
	+	Adaboost	80.13	0.03	72.07	87.72	80.37	0.02	71.11	86.24	75.55	0.33	68.22	89.03
	ABN + LBR													
SVM	MLP + NB	Bagging	84.32	0.05	77.22	91.50	86.55	0.01	73.67	91.15	70.02	0.27	58.19	79.55
	+	Adaboost	83.68	0.18	74.60	89.16	83.23	0.02	75.45	85.64	88.89	0.17	73.46	92.25
	TAN + BN													
SVM	MLP + kNN	Bagging	88.77	0.16	74.85	93.91	89.91	0.16	77.76	92.12	82.20	0.19	70.25	89.97
	+	Adaboost	91.38	0.10	80.30	95.02	92.06	0.12	80.05	94.09	90.64	0.01	82.77	93.85
	ABN + BN													
MLP	SVM + NB	Bagging	85.47	0.05	76.02	87.00	85.51	0.04	70.17	93.42	89.90	0.20	80.08	92.18
	+	Adaboost	90.21	0.04	82.55	96.45	91.95	0.05	83.30	95.80	91.11	0.18	81.45	94.93
	TAN + BN													
MLP	BN + TAN	Bagging	84.53	0.14	73.16	88.80	83.70	0.11	71.92	87.65	79.49	0.09	62.21	90.00
	+	Adaboost	88.00	0.07	79.60	90.91	85.05	0.15	75.50	90.00	85.60	0.56	77.06	92.02
	ABN + kNN													
MLP	kNN + NB	Bagging	90.10	0.09	82.43	95.15	92.71	0.14	84.21	95.10	92.22	0.02	82.01	95.10
	+	Adaboost	93.55	0.02	83.88	96.20	94.82	0.01	83.83	95.94	93.05	0.04	82.90	95.86
	BN + SVM													

79.73% F1 score and 80.03% G-mean, yet without resampling creation, values of 84.37% F1 score and 84.89% G-mean were achieved. Also for kNN, 80.72% F1 score and 80.06% G-mean were obtained in ensembles with SVM, Bayesian Networks and Naïve Bayes when conducting Bagging resampling, but higher values of 81.55% F1 score and 81.00% G-mean were accomplished when not using any resampling technique. The superiority of AdaBoost in improving performance in comparison with Bagging demonstrates its powerful consistency with the suggested prediction procedure. The fusion results of some sets of learners are not reported because of the low prediction results.

As reported above, AdaBoost considerably outperformed the Bagging method. A crucial question to ask is why AdaBoost provided better results and what precisely distinguish AdaBoost from Bagging. One reason for this superb efficiency could be its powerful ability in enlarging the separation margin between the output classes, which plays an essential part in limiting the generalization error of the generated combined learner. Also, Bagging is not suitable choice is some base learners have a high bias. In such cases, AdaBoost is recommended which takes a step ahead and exclude the effect of a high bias present in the baseline model. Another reason for the major outperformance of AdaBoost may be related to its high effectiveness in low noise cases. The findings reveal that AdaBoost is more sensitive to the presence of outliers in the dataset particularly when a well-established pre-processing is conducted leading to a decrease of such deficiencies.

7. Conclusion

Traffic safety management is judged to be a major concern worldwide. In view of the countless health issues, economic losses and fatalities resulting from traffic crashes, a more thorough examination that aims to reduce accidents and improve road safety using

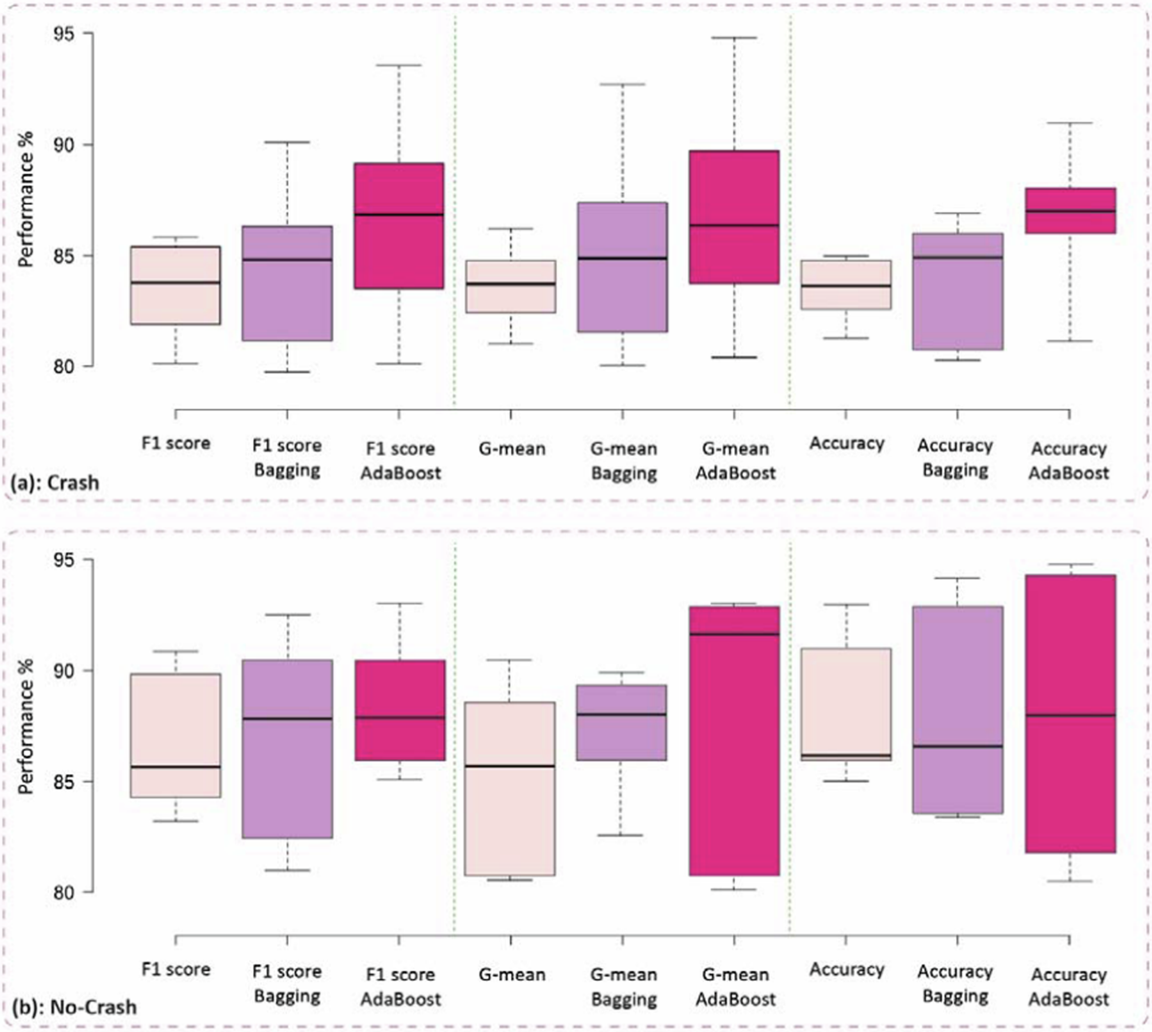

Fig. 10. Crash and No-Crash performance box plots of F1 score, G-mean and Accuracy for the triptych (no resampling, Bagging-resampled and AdaBoost-resampled).

Table 9
Comparison of the suggested crash prediction fusion framework versus other studies.

Paper	Modeling features				Base learners	Feature selection	Hybrid model	Class imbalance	Performance (%)
	VM	DM	PM	EM					
Sun and Sun (2016)	√			√	SVM, K-means	RF	Hybrid	–	79.10 (G-mean) 41.30 (Recall)
Park et al. (2018)	√			√	–	–	Stochastic gradient boosted decision trees	–	94.00 (F1 score)
Liu et al. (2018)	√			√	CART, ANN, SVM	Multiple Factor Analysis	Hybrid	–	72.00 (F1 score)
Kitali et al. (2019)	√			√	Logistic Regression	BN	Hybrid	SMOTE	87.6 (F1 score)
Osman et al. (2019)	√				–	–	AdaBoost	–	99.00 (F1 score)
Zhou et al. (2019)	√			√	SVM	CART	Hybrid	SMOTE	61.00 (G-mean)
Current Article	√	√	√	√	BL, kNN, SVM, MLP	Fusion-based PCA, RF with AdaBoost	Fusion	SMOTE	93.56 (F1 score) 94.81 (G-mean)
Current Article	√	√	√	√	BL, kNN, SVM, MLP	Fusion-based PCA, RF with Bagging	Fusion	SMOTE	90.08 (F1 score) 92.70 (G-mean)

effective and highly performant real-time crash prediction models is imperative. Previous research that have focused on this topic using machine learning models have proven to be highly powerful and have provided satisfying decisions in many transportation systems. Collision Avoidance Systems (CASs), which are designed to forecast imminent accidents, have evidenced their effectiveness in enhancing driving safety. Still, the current CAS design rarely investigates the correlation between crash occurrences and both drivers' state along with weather covariates. For this reason, the purpose of this study is addressing this limitation by developing a crash prediction strategy based on several machine learning techniques and to identify the strongest precursors leading to crash occurrences within a fusion framework incorporating driver inputs, physiological status and vehicle kinematics during different weather patterns.

In this paper, a study of fusion models on the basis of multiple diversity learners namely Bayesian Learners, kNN, SVM and MLP, is presented for crash prediction. Empirical trials were conducted on a driving simulator and four distinct categories of features, including the Heart Rate Variability physiological signal, vehicle kinematics, driver maneuvering inputs and weather conditions, were acquired, fused and processed through approved approaches for crash analysis. Random Forest and Principal Component Analysis were employed for feature selection. Moreover, to cope with the imbalanced data set, the SMOTE technique was used to rebalance the target training data. Bagging and Boosting were applied as resampling algorithms for generating a pool of learners. The performance of diversity models with various sets of learners in a fusion system was compared in reference to the values of F1 score and G-mean as these metrics are deemed to be highly informative considering both precision and recall measures, thus taking the class-balance issue into account. The highest scores were obtained using Boosting which persistently outperformed Bagging in most of the combination established in relation to the individual learners applied. Nearly all of the fusion procedures produced numerically substantial increase in performance over the best individual learners.

This paper opens up new directions in terms of combining classifiers, both in information fusion and fusion models' development. Hence, although the above conclusions have evidenced the effectiveness of crash prediction with physiological signals, driver inputs, vehicle kinematics and weather conditions, and despite the fact that simulator studies ensure a suitable and adjustable environment with the major gain of imitating the driving conduct in a safe environment, the driving simulator is not a complete substitute for the real-world driving experiences. A feasible solution would be to drive similar trials with different simulator setups and to support the simulation outcomes by carrying out fairly identical trails on real-world contexts. Moreover, prior to field application, there is a need to conduct additional research for testing the transferability of the models using data collected based on various driving conditions. In the future, potential directions of this study may include but are not limited to:

1. Expanding the simulations design to include more candidates and additional conduct timing.
2. Optimizing the parameters of classifiers using metaheuristics algorithms to improve the prediction performance.
3. Investigate various undersampling and oversampling techniques to deal with class imbalance problem such ADASYN or Generative Adversarial Network.
4. Broaden the ground truth labels in that dataset to include specific types of crashes such rear-ends and side-impacts.
5. Using additional complicated metrics such as Electroencephalography (EEG) and visual-based behaviors could be highly advantageous.
6. Customizing the proposed framework for a particular crash avoidance/warning system.

Acknowledgment

This research was jointly supported by (1) the Moroccan Ministry of Equipment, Transport and Logistics, and (2) the Moroccan National Center for Scientific and Technical Research (CNRST).

References

Ahmed, M.M., Abdel-Aty, M., Yu, R., 2012. Assessment of Interaction of Crash Occurrence, Mountainous Freeway Geometry, Real-Time Weather, and Traffic Data. Transp. Res. Rec. J. Transp. Res. Board 2280, 51–59. https://doi.org/10.3141/2280-06.

Ahmed, M.M., Abdel-Aty, M.A., 2012. The viability of using automatic vehicle identification data for real-time crash prediction. IEEE Trans. Intell. Transp. Syst. 13, 459–468. https://doi.org/10.1109/TITS.2011.2171052.

Akbari, S., Simonson, C.J., Besant, R.W., 2012. Application of neural networks to predict the transient performance of a Run-Around Membrane Energy Exchanger for yearly non-stop operation. Int. J. Heat Mass Transf. 55, 5403–5416. https://doi.org/10.1016/j.ijheatmasstransfer.2012.04.033.

Ali, E.M., Ahmed, M.M., Wulff, S.S., 2019. Detection of critical safety events on freeways in clear and rainy weather using SHRP2 naturalistic driving data: Parametric and non-parametric techniques. Saf. Sci. https://doi.org/10.1016/j.ssci.2019.01.007.

Aljanahi, A.A., Rhodes, A., Metcalfe, A., 1999. Speed, speed limits and road traffic accidents under free flow conditions. Accid. Anal. Prev. 31, 161–168. https://doi.org/10.1016/S0001-4575(98)00058-X.

Aloul, F., Zualkernan, I., Abu-Salma, R., Al-Ali, H., Al-Merri, M., 2015. IBump: Smartphone application to detect car accidents. Comput. Electr. Eng. 23, 66–75. https://doi.org/10.1016/j.compeleceng.2015.03.003.

Altman, N.S., 1992. An introduction to kernel and nearest-neighbor nonparametric regression. Am. Stat. 46, 175–185. https://doi.org/10.1080/00031305.1992.10475879.

Ba, Y., Zhang, W., Chan, A.H.S., Zhang, T., Cheng, A.S.K., 2016. How drivers fail to avoid crashes: A risk-homeostasis/perception-response (RH/PR) framework evidenced by visual perception, electrodermal activity and behavioral responses. Transp. Res. Part F Traffic Psychol. Behav. 43, 24–35. https://doi.org/10.1016/j.trf.2016.09.025.

Ba, Y., Zhang, W., Wang, Q., Zhou, R., Ren, C., 2017. Crash prediction with behavioral and physiological features for advanced vehicle collision avoidance system. Transp. Res. Part C Emerg. Technol. 74, 22–33. https://doi.org/10.1016/j.trc.2016.11.009.

Backs, R.W., Lenneman, J.K., Wetzel, J.M., Green, P., 2003. Cardiac measures of driver workload during simulated driving with and without visual occlusion. Hum. Factors 45, 525–538. https://doi.org/10.1518/hfes.45.4.525.27089.

Barak, S., Sadegh, S.S., 2016. Forecasting energy consumption using ensemble ARIMA-ANFIS hybrid algorithm. Int. J. Electr. Power Energy Syst. 82, 92–104. https://

doi.org/10.1016/j.ijepes.2016.03.012.
Basheer, I., Hajmeer, M., 2000. Artificial neural networks: fundamentals, computing, design, and application. J. Microbiol. Methods 43, 3–31. https://doi.org/10.1016/S0167-7012(00)00201-3.
Basso, F., Basso, L.J., Bravo, F., Pezoa, R., 2018. Real-time crash prediction in an urban expressway using disaggregated data. Transp. Res. Part C Emerg. Technol. 86, 202–219. https://doi.org/10.1016/j.trc.2017.11.014.
Batuwita, R., Palade, V., 2013. Class Imbalance Learning Methods for Support Vector Machines, in: Imbalanced Learning. John Wiley & Sons, Inc., Hoboken, NJ, USA, pp. 83–99. https://doi.org/10.1002/9781118646106.ch5.
Breiman, L., 2001. Random forests. Mach. Learn. 45, 5–32. https://doi.org/10.1023/A:1010933404324.
Breiman, L., 1996. Bagging predictors. Mach. Learn. 24, 123–140. https://doi.org/10.1007/BF00058655.
Britto, A.S., Sabourin, R., Oliveira, L.E.S., 2014. Dynamic selection of classifiers – a comprehensive review. Pattern Recognit. 47, 3665–3680. https://doi.org/10.1016/j.patcog.2014.05.003.
Brokamp, C., Jandarov, R., Rao, M.B., LeMasters, G., Ryan, P., 2017. Exposure assessment models for elemental components of particulate matter in an urban environment: A comparison of regression and random forest approaches. Atmos. Environ. 151, 1–11. https://doi.org/10.1016/J.ATMOSENV.2016.11.066.
Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P., 2002. SMOTE: Synthetic Minority Over-sampling Technique. J. Artif. Intell. Res. 16, 321–357. https://doi.org/10.1613/jair.953.
Chen, L. Lan, Zhao, Y., Ye, P. Fei, Zhang, J., Zou, J. Zhong, 2017. Detecting driving stress in physiological signals based on multimodal feature analysis and kernel classifiers. Expert Syst. Appl. 85, 279–291. https://doi.org/10.1016/j.eswa.2017.01.040.
Chih-Wei, H., Chih-Chung, C., Chih-Jen, L., 2003. A practical guide to support vector classification. Theory Cult. Soc. https://doi.org/10.1177/02632760022050997.
Cruz, R.M.O., Sabourin, R., Cavalcanti, G.D.C., 2018. Dynamic classifier selection: Recent advances and perspectives. Inf. Fusion 41, 195–216. https://doi.org/10.1016/j.inffus.2017.09.010.
Cui, L., Hu, J., Park, B.B., Bujanovic, P., 2018. Development of a simulation platform for safety impact analysis considering vehicle dynamics, sensor errors, and communication latencies: Assessing cooperative adaptive cruise control under cyber attack. Transp. Res. Part C Emerg. Technol. 97, 1–22. https://doi.org/10.1016/j.trc.2018.10.005.
Dahll, G., 2000. Combining disparate sources of information in the safety assessment of software-based systems. Nucl. Eng. Des. 195, 307–319. https://doi.org/10.1016/S0029-5493(99)00213-7.
Dong, N., Huang, H., Zheng, L., 2015. Support vector machine in crash prediction at the level of traffic analysis zones: Assessing the spatial proximity effects. Accid. Anal. Prev. 82, 192–198. https://doi.org/10.1016/j.aap.2015.05.018.
Elamrani Abou Elassad Z., Mousannif, H., 2019. Understanding Driving Behavior: Measurement, Modeling and Analysis. Adv. Intell. Syst. Comput. 5. https://doi.org/10.1007/978-3-030-11928-7_41.
Elamrani Abou Elassad, Z., Mousannif, H., Al, H., Karkouch, A., 2020a. Engineering Applications of Artificial Intelligence The application of machine learning techniques for driving behavior analysis : A conceptual framework and a systematic literature review. Eng. Appl. Artif. Intell. 87, 103312. https://doi.org/10.1016/j.engappai.2019.103312.
Elamrani Abou Elassad, Z., Mousannif, H., Al Moatassime, H., 2020b. Class-imbalanced crash prediction based on real-time traffic and weather data: A driving simulator study. Traffic Inj. Prev. 1–8. https://doi.org/10.1080/15389588.2020.1723794.
Elamrani Abou Elassad, Z., Mousannif, H., Al Moatassime, H., 2020c. Towards analyzing crash events for novice drivers under reduced-visibility settings: a simulator study. In: Proceedings of the 3rd International Conference on Networking, Information Systems & Security, NISS2020. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3386723.3387849.
Escalante, H.J., Montes, M., Sucar, L.E., 2009. Particle swarm model selection. J. Mach. Learn. Res. 10, 405–440.
Fernández, A., del Jesus, M.J., Herrera, F., 2009. Hierarchical fuzzy rule based classification systems with genetic rule selection for imbalanced data-sets. Int. J. Approx. Reason. 50, 561–577. https://doi.org/10.1016/J.IJAR.2008.11.004.
Ferreiro, S., Sierra, B., Irigoien, I., Gorritxategi, E., 2011. Data mining for quality control: Burr detection in the drilling process. Comput. Ind. Eng. 60, 801–810. https://doi.org/10.1016/j.cie.2011.01.018.
FHWA, 2016. How Do Weather Events Impact Roads? – FHWA Road Weather Management [WWW Document]. URL https://ops.fhwa.dot.gov/weather/q1_roadimpact.htm (accessed 7.25.19).
Freund, Y., Schapire, R.E., 1997. A decision-theoretic generalization of on-line learning and an application to boosting. J. Comput. Syst. Sci. 55, 119–139. https://doi.org/10.1006/jcss.1997.1504.
Gao, M., Hong, X., Chen, S., Harris, C.J., 2011. A combined SMOTE and PSO based RBF classifier for two-class imbalanced problems. Neurocomputing 74, 3456–3466. https://doi.org/10.1016/J.NEUCOM.2011.06.010.
Gravina, R., Alinia, P., Ghasemzadeh, H., Fortino, G., 2017. Multi-sensor fusion in body sensor networks: State-of-the-art and research challenges. Inf. Fusion 35, 1339–1351. https://doi.org/10.1016/j.inffus.2016.09.005.
Guo, X.C., Yang, J.H., Wu, C.G., Wang, C.Y., Liang, Y.C., 2008. A novel LS-SVMs hyper-parameter selection based on particle swarm optimization. Neurocomputing 3211–3215. https://doi.org/10.1016/j.neucom.2008.04.027.
Haghighi, M.S., Vahedian, A., Yazdi, H.S., 2011. Creating and measuring diversity in multiple classifier systems using support vector data description. Appl. Soft Comput. J. 11, 4931–4942. https://doi.org/10.1016/j.asoc.2011.06.006.
Haibo, H., Garcia, E.A., 2009. Learning from imbalanced data. IEEE Trans. Knowl. Data Eng. 21, 1263–1284. https://doi.org/10.1109/TKDE.2008.239.
Haixiang, G., Yijing, L., Shang, J., Mingyun, G., Yuanyue, H., Bing, G., 2017. Learning from class-imbalanced data: Review of methods and applications. Expert Syst. Appl. 73, 220–239. https://doi.org/10.1016/J.ESWA.2016.12.035.
Halim, Z., Kalsoom, R., Bashir, S., Abbas, G., 2016. Artificial intelligence techniques for driving safety and vehicle crash prediction. Artif. Intell. Rev. 46, 351–387. https://doi.org/10.1007/s10462-016-9467-9.
Hassan, H.M., Abdel-Aty, M.A., 2013. Predicting reduced visibility related crashes on freeways using real-time traffic flow data. J. Safety Res. 45, 29–36. https://doi.org/10.1016/j.jsr.2012.12.004.
Hoffenson, S., Frischknecht, B.D., Papalambros, P.Y., 2013. A market systems analysis of the U.S. Sport Utility Vehicle market considering frontal crash safety technology and policy. Accid. Anal. Prev. 50, 943–954. https://doi.org/10.1016/j.aap.2012.07.021.
Hossain, M., Muromachi, Y., 2012. A Bayesian network based framework for real-time crash prediction on the basic freeway segments of urban expressways. Accid. Anal. Prev. 45, 373–381. https://doi.org/10.1016/j.aap.2011.08.004.
Jamson, A.H., Lai, F.C.H., Carsten, O.M.J., 2008. Potential benefits of an adaptive forward collision warning system. Transp. Res. Part C Emerg. Technol. 16, 471–484. https://doi.org/10.1016/j.trc.2007.09.003.
Jolliffe, I.T., 2010. Principal Component Analysis. Springer.
Kan, H.J., Kharrazi, H., Chang, H.-Y., Bodycombe, D., Lemke, K., Weiner, J.P., 2019. Exploring the use of machine learning for risk adjustment: A comparison of standard and penalized linear regression models in predicting health care costs in older adults. PLoS One 14, e0213258. https://doi.org/10.1371/journal.pone.0213258.
Kanjo, E., Younis, E.M.G., Sherkat, N., 2018. Towards unravelling the relationship between on-body, environmental and emotion data using sensor information fusion approach. Inf. Fusion 40, 18–31. https://doi.org/10.1016/j.inffus.2017.05.005.
Kaur, P., Gosain, A., 2018. Comparing the Behavior of Oversampling and Undersampling Approach of Class Imbalance Learning by Combining Class Imbalance Problem with Noise. Springer, Singapore, pp. 23–30. https://doi.org/10.1007/978-981-10-6602-3_3.
Ke, J., Zhang, S., Yang, H., Chen, X. (Michael), 2018. PCA-based missing information imputation for real-time crash likelihood prediction under imbalanced data. Transp. A Transp. Sci. 0, 1–24. https://doi.org/10.1080/23249935.2018.1542414.
Kecman, V., 2005. Support Vector Machines – An Introduction. Springer, Berlin, Heidelberg, pp. 1–47. https://doi.org/10.1007/10984697_1.
Kitali, A.E., Alluri, P., Sando, T., Wu, W., 2019. Identification of Secondary Crash Risk Factors using Penalized Logistic Regression Model. Transp. Res. Rec. https://doi.

org/10.1177/0361198119849053.
Kohavi, R., Kohavi, R., 1995. A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. pp. 1137–1143.
Koller, D., Friedman, N., 2009. Probabilistic Graphical Models: Principles and Techniques - Adaptive Computation and Machine Learning. The MIT Press.
Krawczyk, B., Minku, L.L., Gama, J., Stefanowski, J., Woźniak, M., 2017. Ensemble learning for data stream analysis: A survey. Inf. Fusion 37, 132–156. https://doi.org/10.1016/j.inffus.2017.02.004.
Krogh, A., Vedelsby, J., 1994. Neural network ensembles, cross validation and active learning. pp. 231–238.
Kubat, M., Kubat, M., Matwin, S., 1997. Addressing the curse of imbalanced training sets: one-sided selection. Proc. Fourteenth Int. Conf. Mach. Learn. 179–186.
Kuncheva, L.I., 2004. Combining Pattern Classifiers. Wiley-Interscience New York, NY, USA.
Kwon, O.H., Rhee, W., Yoon, Y., 2015. Application of classification algorithms for analysis of road safety risk factor dependencies. Accid. Anal. Prev. 75, 1–15. https://doi.org/10.1016/j.aap.2014.11.005.
Lee, C., Saccomanno, F., Hellinga, B., 2002. Analysis of crash precursors on instrumented freeways. Transp. Res. Rec. J. Transp. Res. Board 1784, 1–8. https://doi.org/10.3141/1784-01.
Lee, D., Derrible, S., Pereira, F.C., 2018. Comparison of four types of artificial neural network and a multinomial logit model for travel mode choice modeling. Transp. Res. Rec. J. Transp. Res. Board 2672, 101–112. https://doi.org/10.1177/0361198118796971.
Liu, J., Boyle, L.N., Banerjee, A.G., 2018. Predicting interstate motor carrier crash rate level using classification models. Accid. Anal. Prev. 120, 211–218. https://doi.org/10.1016/j.aap.2018.06.005.
Madanat, S., Liu, P.-C., 1995. A prototype system for real-time incident likelihood prediction. ITS-IDEA Progr. Proj. Final Rep.
Makond, B., Wang, K.-J., Wang, K.-M., 2015. Probabilistic modeling of short survivability in patients with brain metastasis from lung cancer. Comput. Methods Programs Biomed. 119, 142–162.
Mujalli, R.O., López, G., Garach, L., 2016. Bayes classifiers for imbalanced traffic accidents datasets. Accid. Anal. Prev. 88, 37–51. https://doi.org/10.1016/j.aap.2015.12.003.
Naik, B., Tung, L.W., Zhao, S., Khattak, A.J., 2016. Weather impacts on single-vehicle truck crash injury severity. J. Safety Res. 58, 57–65. https://doi.org/10.1016/j.jsr.2016.06.005.
Nguyen, H.M., Cooper, E.W., Kamei, K., 2011. Borderline over-sampling for imbalanced data classification. Int. J. Knowl. Eng. Soft Data Paradig. 3, 4. https://doi.org/10.1504/IJKESDP.2011.039875.
Osman, O.A., Hajij, M., Bakhit, P.R., Ishak, S., 2019. Prediction of near-crashes from observed vehicle kinematics using machine learning. Transp. Res. Rec. J. Transp. Res. Board, 036119811986262. https://doi.org/10.1177/0361198119862629.
Park, H., Haghani, A., Samuel, S., Knodler, M.A., 2018. Real-time prediction and avoidance of secondary crashes under unexpected traffic congestion. Accid. Anal. Prev. 112, 39–49. https://doi.org/10.1016/j.aap.2017.11.025.
Parsa, A.B., Movahedi, A., Taghipour, H., Derrible, S., Mohammadian, A. (Kouros), 2020. Toward safer highways, application of XGBoost and SHAP for real-time accident detection and feature analysis. Accid. Anal. Prev. 136. https://doi.org/10.1016/j.aap.2019.105405.
Pazzani, M., Merz, C., Murphy, P., Ali, K., Hume, T., Brunk, C., 1994. Reducing Misclassification Costs. In: Mach. Learn. Proc. 1994, pp. 217–225. https://doi.org/10.1016/b978-1-55860-335-6.50034-9.
Perez, M.A., Sudweeks, J.D., Sears, E., Antin, J., Lee, S., Hankey, J.M., Dingus, T.A., 2017. Performance of basic kinematic thresholds in the identification of crash and near-crash events within naturalistic driving data. Accid. Anal. Prev. 103, 10–19. https://doi.org/10.1016/j.aap.2017.03.005.
Qian, Y., Liang, Y., Li, M., Feng, G., Shi, X., 2014. A resampling ensemble algorithm for classification of imbalance problems. Neurocomputing 143, 57–67. https://doi.org/10.1016/j.neucom.2014.06.021.
Ramentol, E., Caballero, Y., Bello, R., Herrera, F., 2012. SMOTE-RSB*: a hybrid preprocessing approach based on oversampling and undersampling for high imbalanced data-sets using SMOTE and rough sets theory. Knowl. Inf. Syst. 33, 245–265.
Rokach, L., 2010. Ensemble-based classifiers. Artif. Intell. Rev. 33, 1–39. https://doi.org/10.1007/s10462-009-9124-7.
Rosey, F., Auberlet, J.M., 2014. Driving simulator configuration impacts drivers' behavior and control performance: An example with studies of a rural intersection. Transp. Res. Part F Traffic Psychol. Behav. 27, 99–111. https://doi.org/10.1016/j.trf.2014.09.009.
Roshandel, S., Zheng, Z., Washington, S., 2015. Impact of real-time traffic characteristics on freeway crash occurrence: Systematic review and meta-analysis. Accid. Anal. Prev. 79, 198–211. https://doi.org/10.1016/j.aap.2015.03.013.
Ruta, D., Gabrys, B., 2005. Classifier selection for majority voting. Inf. Fusion 6, 63–81. https://doi.org/10.1016/j.inffus.2004.04.008.
S. Aghaei, A., Donmez, B., Liu, C.C., He, D., Liu, G., Plataniotis, K.N., Chen, H.-Y.W., Sojoudi, Z., 2016. Smart driver monitoring: when signal processing meets human factors: in the driver's seat. IEEE Signal Process. Mag. 33, 35–48. https://doi.org/10.1109/MSP.2016.2602379.
Schlögl, M., Stütz, R., Laaha, G., Melcher, M., 2019. A comparison of statistical learning methods for deriving determining factors of accident occurrence from an imbalanced high resolution dataset. Accid. Anal. Prev. 127, 134–149. https://doi.org/10.1016/j.aap.2019.02.008.
Seiler, P., Song, B., Hedrick, J.K., 1998. Development of a collision avoidance system. SAE Trans. 107, 1334–1340.
Shi, X., Wong, Y.D., Li, M.Z.F., Chai, C., 2018. Key risk indicators for accident assessment conditioned on pre-crash vehicle trajectory. Accid. Anal. Prev. 117, 346–356. https://doi.org/10.1016/j.aap.2018.05.007.
Siddiqui, C., Abdel-Aty, M., Huang, H., 2012. Aggregate nonparametric safety analysis of traffic zones. Accid. Anal. Prev. 45, 317–325. https://doi.org/10.1016/J.AAP.2011.07.019.
Slightly Mad Studios, n.d. Project CARS - The Ultimate Driver Journey [WWW Document]. URL https://www.projectcarsgame.com/ (accessed 6.23.19).
Sun, J., Lang, J., Fujita, H., Li, H., 2018. Imbalanced enterprise credit evaluation with DTE-SBD: Decision tree ensemble based on SMOTE and bagging with differentiated sampling rates. Inf. Sci. (Ny) 425, 76–91. https://doi.org/10.1016/j.ins.2017.10.017.
Sun, Jie, Sun, Jian, 2016. Real-time crash prediction on urban expressways: Identification of key variables and a hybrid support vector machine model. IET Intell. Transp. Syst. 10, 331–337. https://doi.org/10.1049/iet-its.2014.0288.
Sun, Y., Wong, A.K.C., Kamel, M.S., 2009. Classification of imbalanced data: a review. Int. J. Pattern Recognit. Artif. Intell. 23, 687–719. https://doi.org/10.1142/S0218001409007326.
Tabachnick, B.G., Fidell, L.S., 2013. Using Multivariate Statistics, sixth ed. Pearson Education, Boston.
Theofilatos, A., 2017. Incorporating real-time traffic and weather data to explore road accident likelihood and severity in urban arterials. J. Safety Res. 61, 9–21. https://doi.org/10.1016/j.jsr.2017.02.003.
Theofilatos, A., Chen, C., Antoniou, C., 2019. Comparing machine learning and deep learning methods for real-time crash prediction. Transp. Res. Rec. https://doi.org/10.1177/0361198119841571.
Thomas, P., Bril El Haouzi, H., Suhner, M.C., Thomas, A., Zimmermann, E., Noyel, M., 2018. Using a classifier ensemble for proactive quality monitoring and control: The impact of the choice of classifiers types, selection criterion, and fusion process. Comput. Ind. 99, 193–204. https://doi.org/10.1016/j.compind.2018.03.038.
Tsai, C.F., Hsu, Y.F., Yen, D.C., 2014. A comparative study of classifier ensembles for bankruptcy prediction. Appl. Soft Comput. J. 24, 977–984. https://doi.org/10.1016/j.asoc.2014.08.047.
Tsymbal, A., Pechenizkiy, M., Cunningham, P., 2005. Diversity in search strategies for ensemble feature selection. Inf. Fusion 6, 83–98. https://doi.org/10.1016/j.inffus.2004.04.003.
Vapnik, V.N., 1995. The Nature of Statistical Learning Theory. Springer.
Wang, C., Quddus, M.A., Ison, S.G., 2013. The effect of traffic and road characteristics on road safety: A review and future research direction. Saf. Sci. https://doi.org/10.1016/j.ssci.2013.02.012.
Wang, C., Xu, C., Dai, Y., 2019a. A crash prediction method based on bivariate extreme value theory and video-based vehicle trajectory data. Accid. Anal. Prev. 123, 365–373. https://doi.org/10.1016/j.aap.2018.12.013.
Wang, J., Kong, Y., Fu, T., 2019b. Expressway crash risk prediction using back propagation neural network: A brief investigation on safety resilience. Accid. Anal. Prev. 124, 180–192. https://doi.org/10.1016/j.aap.2019.01.007.

Wang, J., Xu, W., Gong, Y., 2010. Real-time driving danger-level prediction. Eng. Appl. Artif. Intell. 23, 1247–1254. https://doi.org/10.1016/j.engappai.2010.01.001.
Wang, K.-J., Makond, B., Chen, K.-H., Wang, K.-M., 2014. A hybrid classifier combining SMOTE with PSO to estimate 5-year survivability of breast cancer patients. Appl. Soft Comput. 20, 15–24.
Wang, L., Shi, Q., Abdel-Aty, M., 2015. Predicting crashes on expressway ramps with real-time traffic and weather data. Transp. Res. Rec. J. Transp. Res. Board 2514, 32–38. https://doi.org/10.3141/2514-04.
Ward, J.R., Agamennoni, G., Worrall, S., Bender, A., Nebot, E., 2015. Extending Time to Collision for probabilistic reasoning in general traffic scenarios. Transp. Res. Part C Emerg. Technol. 51, 66–82. https://doi.org/10.1016/j.trc.2014.11.002.
Werneke, J., Vollrath, M., 2013. How to present collision warnings at intersections? - A comparison of different approaches. Accid. Anal. Prev. 52, 91–99. https://doi.org/10.1016/j.aap.2012.12.001.
WHO, 2017. Road traffic injuries [WWW Document]. URL https://www.who.int/en/news-room/fact-sheets/detail/road-traffic-injuries (accessed 7.18.19).
Windeatt, T., 2005. Diversity measures for multiple classifier system analysis and design. Inf. Fusion 6, 21–36. https://doi.org/10.1016/j.inffus.2004.04.002.
Wold, S., Esbensen, K., Geladi, P., 1987. Principal component analysis. Chemom. Intell. Lab. Syst. 2, 37–52. https://doi.org/10.1016/0169-7439(87)80084-9.
Woźniak, M., Graña, M., Corchado, E., 2014. A survey of multiple classifier systems as hybrid systems. Inf. Fusion 16, 3–17. https://doi.org/10.1016/j.inffus.2013.04.006.
Wu, P., Meng, X., Song, L., 2019. A novel ensemble learning method for crash prediction using road geometric alignments and traffic data. J. Transp. Saf. Secur. 1–19. https://doi.org/10.1080/19439962.2019.1579288.
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.
Yan, X., Zhang, Y., Ma, L., 2015. The influence of in-vehicle speech warning timing on drivers' collision avoidance performance at signalized intersections. Transp. Res. Part C Emerg. Technol. 51, 231–242. https://doi.org/10.1016/j.trc.2014.12.003.
Yu, R., Abdel-Aty, M., 2013a. 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.
Yu, R., Abdel-Aty, M., 2013b. Investigating the different characteristics of weekday and weekend crashes. J. Safety Res. 46, 91–97. https://doi.org/10.1016/j.jsr.2013.05.002.
Yu, R., Zheng, Y., Abdel-Aty, M., Gao, Z., 2019. Exploring crash mechanisms with microscopic traffic flow variables: A hybrid approach with latent class logit and path analysis models. Accid. Anal. Prev. 125, 70–78. https://doi.org/10.1016/j.aap.2019.01.022.
Zeng, Z., Zhu, W., Ke, R., Ash, J., Wang, Y., Xu, J., Xu, X., 2017. A generalized nonlinear model-based mixed multinomial logit approach for crash data analysis. Accid. Anal. Prev. 99, 51–65. https://doi.org/10.1016/J.AAP.2016.11.008.
Zhang, Y., Li, X., Gao, L., Wang, L., Wen, L., 2018. Imbalanced data fault diagnosis of rotating machinery using synthetic oversampling and feature learning. J. Manuf. Syst. 48, 34–50.
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. Sustain. 11, 1–14. https://doi.org/10.3390/su11051327.