ANALYSIS OF ROADWAY FATAL ACCIDENTS USING ENSEMBLE-BASED META-CLASSIFIERS

Waheeda Almayyan

Computer Information Department, Collage of Business Studies, PAAET, Kuwait

ABSTRACT

In the past decades, a lot of effort has been put into roadway traffic safety. With the help of data mining, the analysis of roadway traffic data is much needed to understand the factors related to fatal accidents. This paper analyses Fatality Analysis Reporting System (FARS) dataset using several data mining algorithms. Here, we compare the performance of four meta-classifiers and four data-oriented techniques known for their ability to handle imbalanced datasets, entirely based on Random Forest classifier. Also, we study the effect of applying several feature selection algorithms including PSO, Cuckoo, Bat and Tabu on improving the accuracy and efficiency of classification. The empirical results show that the Threshold selector meta-classifier combined with over-sampling techniques results were very satisfactory. In this regard, the proposed technique has gained a mean overall Accuracy of 91% and a Balanced Accuracy that varies between 96% to 99% using 7-15 features instead of 50 original features.

KEYWORDS

Data mining; roadway traffic safety; Imbalanced Data; Meta-classifiers; Ensemble classifier; Data Sampling.

1. INTRODUCTION

Nowadays, traffic road accidents are considered a major problem that confronts people health all around the world. The figures from World Health Organization about global road accident fatalities showed that fatalities were approximately 1.35 million people annually worldwide [1]. Road traffic injuries are now the leading killer of people aged 5-29 years. The cost of traffic accidents is estimated to be 3% of Gross Domestic Product worldwide and more than 90% of road traffic deaths that occurs in low-and middle-income countries. Analysing the severity of accidents is the key of improving the road safety [2]. Recent roadway safety studies have focused on identifying the contributing factors that impact accident severity. Nevertheless, several risk causes are waiting to be discovered or analysed [3].

A number of publications have examined the use of data mining methods in many aspects [4]. Data mining can be defined as a process that uses a variety of data analysis tools to discover patterns and relationships in data that may be used to make valid predictions. Data mining tools are based on highly automated search procedures. Basically, it overcomes the weaknesses of traditional techniques that operate under the assumption that data are distributed normally or according to another distribution, which can be incorrect and may be difficult to validate. Data mining methods intense rely on use of computing power. One of its strength points, it is able to handle categorical variables with a large number of categories, incomplete and noisy datasets [4]. Considerably less researchers considered feature selection compared to the growth achieved in resampling techniques [5]. Under imbalanced scenarios, minority class samples can easily be

discarded as noise. However, such risk can be reduced if the irrelevant features in the feature space are removed.

The literature review indicates a great interest in adapting data mining algorithms in analysing road accidents data [8-18]. Our main task is to develop machine learning based intelligent model that could classify the severity of injuries in FARS dataset more accurately. The Fatal Accidents Dataset contains all serious accidents that happened on public roads in 2007 reported to the National Highway Transportation Safety Administration (NHTSA) [6]. The dataset can be downloaded from California Polytechnic State University and all data originally came from FARS. The dataset contains 37,248 records and 55 attributes. The data description can be found in the document FARS Analytic Reference Guide during the years 1975-2007 [7]. The objective of this study is twofold: firstly, to propose an algorithm for extracting factors that are significantly related to the car accidents according to their injury severity. Secondly, to investigate the impact of data-oriented/re-sampling techniques methods on enhancing the classification performance the imbalanced dataset.

The remaining parts of the paper are organized as follows. Section 2 summarizes the recent research related work that addresses the problem of roadway traffic safety, followed by sections 3 which include the methodology, the applied meta-classifiers and the applied feature selection techniques. The evaluation environment, and performance metrics along with the evaluation of the different models and experimental results are discussed in Section 4. Finally, conclusions are presented in Section 5.

2. RELATED WORKS

In 1987, individual states in the USA were allowed to raise speed limits on rural freeways from 55 to 65 mph. Ossiander and Cummings analysed the effect of the increased speed limits through designing an ecological study of crashes and vehicle speeds on Washington State freeways from 1974 through 1994 [8]. They concluded that the incidence of fatal crashes more than doubled after 1987 and the death rate go up by 27% compared to increase in 10% in the states that did not increase the speed limit. Solaiman et. al. developed an Internet-based prototype GIS and Road Accident View System for automated road accident analysis and visualization [9]. The suggested system had the capabilities to perform query over accident information, trend analysis, statistical analysis, color-coded mapping and other accident information displayed within the web-based environment. In addition, it could predict the possible dangerous accident sites from the data gathered using the map API the system.

Researchers in [10] used partition-based and density-based clustering to analyse the road accidents data. They first cluster the accident data using K-modes algorithm and then association-rule mining technique is applied to identify the correlation among various sets of attributes in which an accident may occur for each cluster. Chang et.al in [11] applied the classification and regression tree model (CART) to analyse the Taiwan traffic accidents data in 2001. It studied the relationship between fatal injuries and driver/vehicle, highway/environment and accident variables. The results indicated that the most important variable related to crash severity is the vehicle type. They identified that pedestrians, motorcycle and bicycle riders have higher risks of being injured than other types of vehicle drivers in traffic accidents. Krishnaveni and Hemalatha applied several classification models to predict the injury severities in traffic accidents that occurred in Hong Kong during 2008 [12]. Naive Bayes, AdaBoostM1, PART Rule, J48 and Random Forest classifiers were selected for classifying the type of injury severity. Meanwhile, Genetic Algorithm was applied to reduce the dimensionality of the dataset. The final results showed that Random Forest outperformed the other four algorithms.

Kwon et al. applied several classification algorithms for ranking main factors that cause accidents [13]. With a binary logistic regression model used as the basis for the comparisons, they applied the decision tree classifier and Naive Bayes algorithms. Bahiru et al. built a decision tree based on J48, ID3, CART and Naïve Bayes classifiers to model the severity of injury [14]. The experimental result showed that the accuracy of J48 classifier is higher than others models. The author believed that closely studying the road traffic accident data with the accident severity and time components of the accident can help to identify any hidden temporal patterns. Silva and Saraee proposed novel approach of combining Decision Tree algorithm and Time-Series Calendar Heatmaps technique to extract the knowledge with time factors from the accident datasets that happened on a region in North of England [15]. Based on the decision tree models and evaluation measures, they noticed that there is a correlation between hour and month of the accident and the severity of the accident.

Researchers in [16], noticed that traffic accidents datasets are usually imbalanced. Therefore, they investigated under-sampling, oversampling and a mix technique that combines both techniques. Different Bayes classifiers were used to analyse imbalanced and balanced traffic crashes datasets in Jordan for three years. The results indicated that the most influencing parameters were the number of vehicles involved, accident pattern, number of directions, accident type, lighting, surface condition, and speed limit. Moreover, they noticed that using the balanced data sets using oversampling technique with Bayesian networks improved classification results. Li et al. [17] applied Apriori algorithm, Naive Bayes classifier and k-means clustering algorithms on the FARS Fatal Accident dataset to study the relationship between injury severity and other attributes such as collision manner, light condition, drunk driver weather conditions. The analysis result suggested that the human factors like drunk or not and the collision type have a stronger effect on the fatality rate more than environmental factors like roadway surface, weather, and light conditions. Pakgohar et al. [18] explored the role of human factors on incidence and severity of road crashes in Iran by employing descriptive analysis; Logistic Regression, Classification and Regression Tree. The study indicated the human responsibility on the occurrence of fatal accidents. The result of the study recommends the important role of issuing ‘Driving License’ and using ‘Safety Belt’ safety policies which might lower the severity of injuries in traffic accidents in Iran.

3. METHODOLOGY

First of all, as a pre-processing step we calculated injury_severity variable according to number of casualties and the number of persons involved in the accident and binned to two categories, high and low severity. This variable is adapted from [17] and it represents the percentage of severity in accidents and calculated as

$$\text{Injury\_severity} = \text{FATALS} / \text{PERSONS} \quad (1)$$

Where FATALS is the number of fatalities and PERSONS is the number of persons involved in the accident. Injury_severity is referred as “class” in the analysis. And according to this, the number of samples in the FARS dataset are 25545 instances of minor car accidents and 11703 instances for serious accidents. So, the current imbalance ratio in FARS dataset is 2:1 for majority and minority classes, respectively. Therefore, we decided to tackle this issue with more investigation.

Several studies give equal importance to all classes, assuming that datasets are balanced and this often results in poor classification results [19-23]. The imbalance problem happens when the numbers of instances of one class outnumber the others. This case is commonly known in the field of real datasets which need to be usually handled [19,20]. Schierz et al. [21] compared four

Cost-sensitive classifiers, namely Naïve Bayes, SVM, Random Forest and C4.5 decision tree to classify pharmaceutical data. They noticed that SVM and C4.5 decision tree classifiers have performed relatively well. Other studies discussed adapting data-oriented/resampling techniques as a solution for the imbalance problem. Data level resampling is one of the many ways to handle class imbalance problem. In a recent study, Singh proposed a data level resampling method to improve learning from class imbalanced datasets in health applications [23]. The learning process on different datasets has improved by incorporating the distribution structure of minority class samples to generate new data samples using deep learning neural networks.

The primary objective of this research work is aimed at enhancing FARS dataset classification with exploring minimum number of features which are significantly related to the car accidents. The primary focus is on investigating the impact of data-oriented/re-sampling techniques methods on enhancing the classification performance over the imbalanced dataset. In this study we will evaluate the performance of four meta-classifiers and four data-oriented techniques to handle the imbalance ratio.

3.1. Meta Classifiers

Numerous classification solutions have been suggested in literature to handle imbalanced datasets. Several studies investigated the performance of classifiers which should help in choosing the appropriate classification method. Ensemble learning algorithms which utilizes ensembles of classifiers such as neural networks, Random Forest, bagging and boosting, have received an increasing interest for of their ability to deliver an accurate prediction and robust to noise and outliners than single classifiers [19,24]. The basic idea behind ensembled classifiers is based upon the premise that a group of classifiers can perform better than an individual classifier.

In this research, we compared the performance of four different meta-classifiers known for their ability to handle imbalanced datasets, explicitly, Bagging, Cost-sensitive, MetaCost and Threshold Selection classifiers with Random Forest as base-classifier. Random Forest consists of a combination of individual base classifiers where each tree is generated using a random vector sampled independently from the classification input vector to enable a much faster construction of trees. In 2001, Breiman proposed a promising tree-based ensemble classifier based on a combination tree of predictors such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees and named it Random Forest [25]. Random Forest is as one of the best learning methods as it is more robust and can achieve better performances than single decision trees.

3.1.1. Bagging Classifier

Leo Breiman proposed Bagging or Bootstrap aggregating as a meta-algorithm based on ensemble of decision trees to improve classification and regression models in terms of stability and classification accuracy in machine learning [25]. Additionally, it decreases the variance and reduces overfitting. This technique can be used for any type of model such as NN, although it is most applied to decision tree models [26].

In Bagging diversity is obtained by using bootstrapped replicas of the original training set, where different training datasets are randomly drawn with replacement. Consequently, with each training data replica a decision tree is built based on the standard approach. So, each tree can be defined by a different set of variables, nodes and leaves. Finally, their predictions are combined to obtain the final result. The final results can be obtained in regression cases by averaging votes and by combining the outputs of models during classification [27].

3.1.2. Cost-Sensitive Classifier

Cost-sensitive classification learning is a type of learning in data mining that considers the penalty of misclassification cost into consideration [28]. In Cost-sensitive learning process, the objective is to develop a hypothesis that seek to minimize the high cost errors and the total misclassification cost. Therefore, a Cost-sensitive classification technique takes the cost matrix into consideration during model building and generates a model that generate the minimum expected cost [29].

For two-class problem, let C(Min,Maj) denote the cost of misclassifying a majority class instance as minority instance and C(Maj,Min) as the cost of misclassifying a minority class instance as majority instance. When dealing with the class imbalance problem, the cost of misclassifying minority examples is higher than the cost of misclassifying majority, examples (C(Maj,Min) > C(Min,Maj)) and there is no penalty for correct classification (i.e., C(Maj,Maj)=C(Min, Min)= 0).

3.1.3. Threshold-Based Selector Classifier

Threshold Selector is a meta-classifier that sets a threshold on the probability output of a base-classifier. Threshold adjustment for the classifier's decision is one of the methods used for dealing with imbalanced datasets [30]. A meta-classifier selects a mid-point threshold on the probability output by a classifier. The midpoint threshold is set so that a given performance measure is optimized [31]. By default, the probability threshold is assigned to 0.5, i.e. if an instance is attributed with a probability of equal or less than 0.5, it is classified as negative for the respective class, while if it is greater than 0.5, the instance is classified as positive.

Performance is measured either on the training data, a hold-out set or using cross-validation. In addition, the probabilities returned by the base learner can have their range expanded so that the output probabilities will reside between 0 and 1, which is useful if the scheme normally produces probabilities in a very narrow range. For our experiments, the optimal threshold was selected automatically by the meta-classifier by applying internal five-fold cross validation to optimize the threshold according to f-measure (Eq. 13), as measure of a model's accuracy [32].

3.1.4. MetaCost Classifier

MetaCost procedure is based on relabelling the classes of the training examples, and then employs a modified training set to produce the final model. MetaCost depends on an internal Cost-sensitive classifier in order to relabel classes of training examples. Nevertheless the study by Domingos made no comparison between MetaCost's final model and the internal Cost-sensitive classifier on which MetaCost depends [33]. This comparison is worth making as it is credible that the internal Cost-sensitive classifier may outperform the final model without the additional computation required to derive the final model. Boosting [34,35] can be effective and can be better than bagging [36] in minimizing errors as it uses bagging internally. Using a boosting procedure in MetaCost may improve MetaCost's performance. This is the reason why we choose to use boosting procedures in MetaCost in this paper. This meta-classifier makes its base-classifier Cost-sensitive using the method specified in [33]. This implementation uses all bagging iterations when reclassifying training data.

3.2. Feature Selection Algorithms

Feature Selection is a challenging machine learning-related task that aims at reducing the number of features by removing irrelevant, redundant and noisy data while maintaining an acceptable

level of classification accuracy. Essentially, the feature space is explored to reduce the feature space and prepare the conditions for the classification step. The performance of the selected dimension reduction techniques is examined to find the most effective one.

3.2.1. Tabu Search

Tabu Search is a memory-based metaheuristic algorithm proposed by Glover in 1986 to solve combinatorial optimization problems [37,38]. Since then, Tabu Search has been successfully applied in other feature selection problems [39]. Tabu Search is a local neighbourhood search algorithm that simulates the optimal characteristics of human memory functions. Tabu Search involves a local search combined with a tabu mechanism.

It starts with an initial feasible solution X'$\in\Omega$ among the neighbourhood solutions, where Ω is the set of feasible solutions, and at each iteration, the algorithm searches the neighbourhood of the best solution N(X)$\subseteq\Omega$ to obtain a new one with an improved functional value. A solution X' $\in\Omega$ N(X) can be reached from X in two cases, X' is not included in the Tabu list; and X' is included in the Tabu list, but it satisfies the aspiration criterion [40]. Surely, if the new solution *X*' is superior to *X*best, the value of *X*best is overridden. To avoid cycling, solutions that were previously visited are declared forbidden or tabu for a certain number of iterations and this surely improve the performance of the local search. Then, the neighbourhood search is resumed based on the new feasible solution *X*'. This procedure is iteratively executed until the stopping criteria is met. After the iterative process has terminated, the current best solution so far *X*best is considered the final optimal solution provided by the Tabu Search method [41].

3.2.2. Particle Swarm Optimization Search

The particle swarm optimization (PSO) is a population-based stochastic optimization technique introduced in 1995 by Kennedy and Eberhart [42]. In PSO, a possible candidate solution is encoded as a finite-length string called a particle pi in the search space. All of the particles make use of its own memory and knowledge gained by the swarm as a whole to find the best solution. With the purpose of discovering the optimal solution, each particle adjusts its searching direction according to two features, its own best previous experience (p_{best}) and the best experience of its companions flying experience (g_{best}).

Each particle is moving around the n-dimensional search space S with objective function $f : S \subseteq \Re^n \to \Re$. Each particle has a position $x_{i,t}$ fitness function $f(x_{i,t})$ and ''flies'' through the problem space with a velocity $v_{i,t}$. A new position $z1 \in S$ is called better than $z2 \in S$ iff $f(z1) < f(z2)$. Particles evolve simultaneously based on knowledge shared with neighbouring particles; they make use of their own memory and knowledge gained by the swarm as a whole to find the best solution. The best search space position particle i has visited until iteration t is its previous experience p_{best}. To each particle, a subset of all particles is assigned as its neighbourhood. The best previous experience of all neighbours of particle i is called g_{best}. Each particle additionally keeps a fraction of its old velocity. The particle updates its velocity and position with the following equations in continuous PSO [43]:

$$v_{pd}^{new} = \omega * v_{pd}^{old} + C_1 * rand_1() * (pbest_{pd} - x_{pd}^{old}) + C_2 * rand_2() * (gbest_{d_d} - x_{pd}^{old}) \quad (2)$$

$$x_{pd}^{new} = x_{pd}^{old} + v_{pd}^{new} \quad (3)$$

3.2.3. Cuckoo Search

Cuckoo search is a recently developed population-based metaheuristic algorithm developed by Xin-She Yang and Suash Deb in 2009 [44]. Since then it has been used as a successful adaptive search strategy for solving optimization problems. Recent studies showed that Cuckoo search algorithm is computationally efficient and easy to implement with less parameters [45].

The basic idea behind the Cuckoo search algorithm is derived from the brood parasitism of some cuckoo species. These species use the nests of other host birds to lay their eggs in that look like the pattern and color of the native eggs to reduce the probability of discovering them and rely on these birds for accommodating their eggs. Sometimes, some of host birds discover and throw the alien eggs away or simply abandon their nests and build a new one in another place. For the cuckoo search algorithm, each egg in a nest represents a solution, and a cuckoo egg represents a new solution. The goal is to employ the new and potentially better solutions (cuckoos) to replace a not-so-good solution in the nests [45]. Hence, the Cuckoo search algorithm is more efficient in exploring the search space as it will make sure the algorithm will not fall into a local optimum.

3.2.4. Bat Search

The Bat Algorithm is a meta heuristic Swarm Intelligence algorithm proposed in 2010 by Yang [46], who was inspired by the abilities of bats in searching for their prey and discriminating different types of insects and obstacles even at complete darkness. Bats emit loud sound pulses that help them detect target and avoid obstacles. In order to transpose this behaviour into an intelligent algorithm, the author states three hypothesis. First all bats will use echolocation to identify its prey. Secondly, all bats fly randomly and their trajectory is characterized by their internal encoded frequency (freq), velocity (v) and position in space (x). At each iteration of the algorithm these three variables are updated as:

$$freq_i = freq_{min} + (freq_{max} - freq_{min}) \cdot \beta \quad (4)$$

$$vi^t = vi^{t-1} + (xi^{t-1} - x\_best_j) \cdot freq_i \quad (5)$$

$$xi^t = xi^{t-1} + vi^t \quad (6)$$

where $\beta \in [0,1]$ is a random vector drawn from a uniform distribution. Moreover, as the bat attains a position closer to its target then, it will decrease its loudness (A_i) and increase its rate of the pulse emission (r_i) as follows:

$$Ai^{t+1} = \alpha \cdot Ai^t \quad (7)$$

$$ri^{t+1} = ri^0 \cdot [1 - e^{-\gamma \cdot t}] \quad (8)$$

where α ($0 < \alpha < 1$) and γ ($\gamma > 0$) are constants. Finally, the author assumes that the loudness will vary from a large value to a minimum one.

4. EXPERIMENTS AND RESULTS

To estimate the generalized error of our method, we have applied 10-fold cross-validation to avoid overfitting the learned models. For a highly imbalanced datasets, accuracy may be confusing. Therefore, we considered more appropriate performance measures to compare different classifiers for their ability to handle imbalanced datasets such as balanced accuracy

which considers both sensitivity and specificity. Hence, if two models delivered the same sensitivity value, the model that demonstrated higher balanced accuracy will be prioritized for selection. Specificity, Sensitivity, Accuracy, Balanced Accuracy, Precision, f-measure and MCC are the quality metrics for assessing the performance of the learning performance of each classifier before and after data resampling [47]. Their formulae are shown in Equations 9-15.

$$Sensitivity = \frac{TP}{TP+FN} \quad (9)$$

$$Specificity = \frac{TN}{TN+FP} \quad (10)$$

$$Accuracy = \frac{(TP+TN)}{(TP+FP+TN+FN)} \quad (11)$$

$$Balanced\ Accuracy = \frac{1}{2}\left(\frac{TP}{TP+NP} + \frac{TN}{TN+FP}\right) \quad (12)$$

$$Precision = \frac{TP}{TP+FP} \quad (13)$$

$$f\text{-measure} = \frac{2\times TP}{2\times TP+FP+FN} \quad (14)$$

$$MCC = \frac{\{(TP\times TN)-(FP\times FN)\}}{\sqrt{\{(TP+FP)\times(TP+FN)\times(TN+FP)\times(TN+FN)\}}} \quad (15)$$

Where TP: true positives; TN: true negatives; FP: false positives; FN: false negatives.

One of the interesting aspects in data mining is to build computational models with abilities to extract hidden knowledge using data mining schemes. Regarding model settings, not all classifiers require parameter optimization. For example, ClassBalancer automatically reassigns weights to the instances in the dataset such that each class has the same total weight [48,49], therefore, it does not require adjustment. For Bagging, the only parameter to optimize would be the number of bags. In our case, the number of bags was adjusted to 100. For Cost-sensitive Classifier and MetaCost, the cost for misclassification was initially applied in accordance with the imbalance ratio, which, in case it did not provide a sensitivity of at least 0.5, was further increased to arrive at the final model. All the meta-learning methods evaluated in this study were implemented in the WEKA software suite, which is a Java based open source data-mining tool [48]. The number of trees for Random Forest was arbitrarily set to 100, since it has been shown that the optimal number of trees is usually between 64 and 128, and increasing the number of trees does not necessarily improve the model's performance [27,49].

To determine which data-oriented technique is the most suitable for our FARS dataset and before executing the feature selection, we start with performing experiments over the class-imbalanced dataset. The performance values of the meta-classifiers and the suggested sampled dataset are also illustrated in Table 1 to facilitate the comparison with the investigated methods. Observing the values of Specificity, Balanced Accuracy and Precision it is clear that the Threshold selector algorithm has the highest values where it accomplished 94%, 97% and 97%. Whereas, the values of Sensitivity, Accuracy, f-measure and MCC showed that Bagging, Cost-sensitive and MetaCost outperformed the Threshold selector algorithm. We have noticed that Bagging, Cost-sensitive and MetaCost classifiers recorded a comparable performance in most cases. In general, all average Balanced Accuracy were compared and provided improvement in the results with average Accuracies of 90%.

In our experiments, three over-sampling techniques, explicitly Synthetic Minority Oversampling (SMOTE), oversampling minority and Class Balancer together with the Under-sampling method, were applied. SMOTE technique resamples the dataset by applying the over-sampled the minority class. While, Over-sampling minority technique achieves oversampling of the minority class, rather than under-sampling of the majority class, so that both classes have the same number of instances. Under-sampling technique produces a random subsample of a dataset to implement under-sampling of the majority class. These over-sampling techniques work through increasing the number of examples in the minority class to balance the distribution of the data sets and improve the detection rate of the minority class. Whereas Class Balancer reweights the instances in the data so that each class has the same total weight. Regarding the data-oriented techniques results, obviously oversampling the minority class is the best algorithm in all measures. It accomplished 93%, 99%, 96% 100%, 99% 96% and 89% in terms of Sensitivity, Specificity, Accuracy, Balanced Accuracy, Precision, f-measure and MCC. Here we can notice that data- oriented techniques have helped in improvement of prediction performance.

Table 1. Classification results of FARS Dataset

Model	Sensitivity	Specificity	Accuracy	Balanced Accuracy	Precision	f-measure	MCC
Bagging	0.891	0.829	0.872	0.914	0.919	0.905	0.698
Cost-sensitive	0.886	0.828	0.868	0.914	0.918	0.902	0.689
MetaCost	0.882	0.848	0.871	0.924	0.926	0.904	0.694
Threshold selector	0.809	0.943	0.851	0.971	0.968	0.882	0.645
SMOTE	0.868	0.954	0.911	0.977	0.949	0.907	0.790
Oversampling minority	0.928	0.988	0.958	0.994	0.987	0.957	0.890
Class Balancer	0.852	0.920	0.886	0.960	0.914	0.882	0.747
Under sampling	0.894	0.842	0.878	0.921	0.925	0.910	0.711

Feature selection techniques are meant to identify a set of crucial features that have maximum relevancy for target classes and minimum redundancy with other features in the dataset at the same time. The next step is to investigate the potential of using feature selection techniques. At the end of this step a subset of features is chosen for the next round. The optimal features by the PSO, Cuckoo, Bat and Tabu techniques are listed in Table 2. It is noteworthy that the number of features has remarkably reduced, compared with original dataset. In this phase we reduced the size of FARS features from 50 to only 7-15 features.

Table 2. Feature selection results

Feature Selection Algorithm	Features No.	Features Details
PSO	15	MINUTE VE_TOTAL PEDS HARM_EV MAN_COLL REL_ROAD SP_LIMIT ALIGNMNT C_M_ZONE NOT_MIN
		HOSP_HR SCH_BUS CF1 DRUNK_DR VE_FORMS

Cuckoo	13	STATE MINUTE VE_TOTAL PEDS HARM_EV MAN_COLL REL_ROAD ALIGNMNT C_M_ZONE HOSP_HR SCH_BUS DRUNK_DR VE_FORMS
Bat	14	VE_TOTAL PEDS HARM_EV MAN_COLL REL_ROAD ALIGNMNT TRA_CONT HOSP_HR HOSP_MN SCH_BUS CF1 DRUNK_DR VE_FORMS WEATHER
Tabu	7	MINUTE HARM_EV REL_ROAD HOSP_MN SCH_BUS DRUNK_DR VE_FORMS

Table 3 shows the classification results of the features selected by PSO. It can be observed that the classification Balanced Accuracy using PSO technique varies between 91% and 96% with the selected 15 feature set. In this step Threshold selector has achieved the results of 88%, 93%, 96.5% and 96%, in terms of Sensitivity, Specificity, Balanced Accuracy and Precision. While the score of other classification models were similar with an average Accuracy of 86%, 89% for f-measure and 67% for MCC. Observing the data-oriented techniques, obviously under sampling the majority class is the best algorithm in most measures. It achieved the results of 92%, 90%, 94%, 93% and 77%, in terms of Sensitivity, Accuracy, Precision f-measure and MCC. Oversampling minority technique scored the highest Specificity and Balanced Accuracy scores. We noticed that the overall performance has not been affected by reducing the number of features from 50 to 15.

Table 3. Performance comparison of the different learning paradigms after applying PSO algorithm

Model	Sensitivity	Specificity	Accuracy	Balanced Accuracy	Precision	f-measure	MCC
Bagging	0.880	0.835	0.866	0.917	0.921	0.900	0.683
Cost-sensitive	0.874	0.849	0.866	0.924	0.926	0.899	0.682
MetaCost	0.873	0.852	0.867	0.926	0.928	0.900	0.683
Threshold selector	0.833	0.931	0.864	0.965	0.963	0.893	0.671
SMOTE	0.859	0.936	0.897	0.967	0.930	0.893	0.766
Oversampling minority	0.842	0.940	0.891	0.970	0.935	0.886	0.746
Class Balancer	0.851	0.885	0.868	0.942	0.881	0.866	0.724
Under sampling	0.920	0.869	0.904	0.935	0.939	0.929	0.772

Table 4 presents the classification results of the features selected by Cuckoo search technique. It can be observed that the classification Balanced Accuracy using Cuckoo technique varies between 91% and 98% with the selected 13 feature set. In this step Threshold selector model has achieved the best results of 96%, 98% and 98% in terms of Specificity, Balanced Accuracy and

Precision. While the score of other classification models were similar with an average of 88% for Sensitivity, an Accuracy of 87%, f-measure of 90% and MCC of 68%. Observing the data-oriented techniques readings, obviously over sampling the minority class is the best algorithm in most measures. It achieved the results of 93%, 96%, 94%, 98%, 96%, 94% and 88%, in terms of Sensitivity, Specificity, Accuracy, Balanced Accuracy, Precision, f-measure and MCC. Oversampling minority technique scored the highest Specificity and Balanced Accuracy scores. We noticed that the overall performance has not been affected by reducing the number of features too.

Table 4. Performance comparison of the different learning paradigms after applying Cuckoo algorithm

Model	Sensitivity	Specificity	Accuracy	Balanced Accuracy	Precision	f-measure	MCC
Bagging	0.886	0.829	0.868	0.914	0.918	0.902	0.689
Cost-sensitive	0.887	0.830	0.869	0.915	0.919	0.903	0.691
MetaCost	0.882	0.843	0.870	0.921	0.925	0.903	0.691
Threshold selector	0.790	0.964	0.844	0.982	0.979	0.874	0.631
SMOTE	0.874	0.885	0.879	0.942	0.883	0.878	0.755
Oversampling minority	0.931	0.961	0.946	0.981	0.960	0.945	0.879
Class Balancer	0.857	0.874	0.866	0.937	0.872	0.865	0.726
Under sampling	0.922	0.872	0.906	0.936	0.940	0.931	0.777

Next, Table 5 shows the classification results of the 14 features selected by Bat search technique. We observed that the classification Balanced Accuracy using Bat algorithm varies between 90% and 93%. Threshold selector, in this step, has gained the results of 86%, 93% and 93% in terms of Specificity, Balanced Accuracy and Precision. While the score of other classification models were similar with an average of 87% for Sensitivity, an Accuracy of 85%, f-measure of 89% and MCC of 65%. Observing the data-oriented techniques, obviously over sampling the minority class is the best algorithm in most measures.

As it achieved the results of 91%, 90%, 94%, 77% and 93%, in terms of Sensitivity, Accuracy, Precision, MCC and f-measure. Oversampling minority scored the highest Specificity and Balanced Accuracy scores. Worth mentioning that SMOTE algorithm scored the best scores in Specificity and Balanced Accuracy. The results obtained suggest that over sampling the minority class technique has the ability to enhance the predictive accuracy of the Random Forest.

Table 5. Performance comparison of the different learning paradigms after applying Bat algorithm

Model	*Sensitivity*	*Specificity*	*Accuracy*	*Balanced Accuracy*	*Precision*	*f-measure*	*MCC*
Bagging	0.879	0.819	0.860	0.909	0.914	0.896	0.671
Cost-sensitive	0.878	0.818	0.859	0.909	0.913	0.895	0.668
MetaCost	0.874	0.807	0.853	0.903	0.908	0.891	0.654
Threshold selector	0.848	0.858	0.851	0.929	0.929	0.887	0.647
SMOTE	0.855	0.934	0.895	0.967	0.928	0.890	0.760
Oversampling minority	0.917	0.883	0.906	0.942	0.945	0.931	0.776
Class Balancer	0.837	0.900	0.869	0.950	0.893	0.864	0.715
Under sampling	0.903	0.847	0.886	0.924	0.928	0.916	0.730

Table 6 shows the classification results of the 7 features selected by Tabu search technique. We observed that the classification Balanced Accuracy using Tabu technique varies between 91% and 96%. In this step Threshold selector has obtained the results of 92%, 96% and 96% in terms of Specificity, Balanced Accuracy and Precision. While the scores of other classification models were similar with an average of 85% for Accuracy, 91% for Precision, 89% for f-measure and 66% for MCC. As for the data-oriented techniques, obviously Oversampling the minority class is the best algorithm in most measures. It achieved the results of 93%, 91%, 94%, 94% and 80%, in terms of Sensitivity, Accuracy, Precision, MCC and f-measure. Noteworthy that SMOTE algorithm scored the best performance at Specificity and Balanced Accuracy with a scores of 91% and 95% respectively.

Table 6. Performance comparison of the different learning paradigms after applying Tabu

Model	*Sensitivity*	*Specificity*	*Accuracy*	*Balanced Accuracy*	*Precision*	*f- measure*	*MCC*
Bagging	0.872	0.826	0.858	0.913	0.916	0.894	0.665
Cost-sensitive	0.870	0.844	0.861	0.922	0.924	0.896	0.671
MetaCost	0.869	0.835	0.858	0.918	0.920	0.894	0.664
Threshold selector	0.817	0.922	0.850	0.961	0.958	0.882	0.643
SMOTE	0.865	0.913	0.890	0.957	0.908	0.886	0.761
Oversampling minority	0.937	0.871	0.916	0.936	0.941	0.939	0.804
Class Balancer	0.846	0.861	0.853	0.930	0.858	0.852	0.701
Under sampling	0.916	0.829	0.889	0.915	0.921	0.919	0.740

Now, to test the prediction performance of the agreement between the feature selection techniques a group of experiments were conducted. Figure 1 visualizes the Venn diagram of the top five relevant features between PSO, Bat, Cuckoo and Tabu which are: MINUTE, HARM_EV, REL_ROAD, SCH_BUS and DRUNK_DR. Table 7 shows the classification results of the above mentioned five relevant features. we observed that the classification Balanced Accuracy using Tabu technique varies between 87% and 92%. In this step Threshold selector has obtained the results of 85%, 92% and 92% in terms of Specificity, Balanced Accuracy and Precision. While the scores of other classification models were similar with an average of 81% for Accuracy, 83% for Sensitivity, 85% for f-measure and 55% for MCC.

As for the data-oriented techniques, obviously under sampling the majority class is the best algorithm in most measures. It achieved the results of 87%, 90% and 88%, in terms of Sensitivity, Precision and f-measure. Noteworthy that SMOTE algorithm scored the best performance at Specificity, Accuracy, Balanced Accuracy and MCC with a scores of 91%, 86%, 95% and 70% respectively. It was fascinating to see that in general all different sets of classifiers have resulted in an Accuracy above 90%.

Table 7. Performance comparison of the different learning paradigms after choosing most relevant features

Model	*Sensitivity*	*Specificity*	*Accuracy*	*Balanced Accuracy*	*Precision*	*f-measure*	*MCC*
Bagging	0.837	0.756	0.811	0.878	0.882	0.859	0.562
Cost-sensitive	0.827	0.778	0.812	0.889	0.891	0.857	0.563
MetaCost	0.839	0.739	0.807	0.869	0.875	0.856	0.553
Threshold selector	0.795	0.849	0.812	0.924	0.920	0.853	0.566
SMOTE	0.818	0.910	0.864	0.955	0.900	0.857	0.697

Oversampling minority	0.821	0.757	0.802	0.879	0.883	0.851	0.539
Class Balancer	0.800	0.864	0.832	0.932	0.855	0.827	0.644
Under sampling	0.870	0.794	0.846	0.897	0.902	0.886	0.639

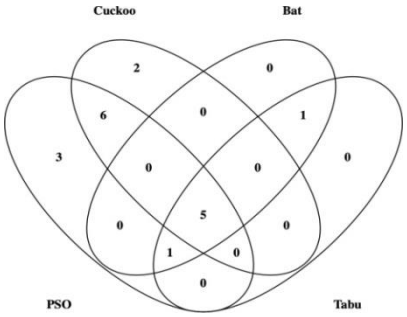

Figure 1. Relationship between PSO, Cuckoo, Bat and Tabu search algorithms

The last set of experiments confirmed that the suggested technique can efficiently compete with the best prediction meta-classifiers with least number of features. The suggested classification model is considered adequate enough for selection as the 10-fold cross-validation provided a sensitivity value of at least 0.5 and a specificity value not less than 0.5 in all the experiments. It also indicates that with the assistance of meta-classifiers performance has improved particularly when handling imbalanced datasets. Moreover, employing Random Forest classifier as a base-classifier surely improves their prediction accuracy. Among the meta-classifier-based methods, Threshold selector provided the best performance in many cases. Additionally, Over-sampling technique results are very satisfactory than other resampling techniques. In this regard, the proposed technique has gained a mean Overall Accuracy of 91% and a Balanced Accuracy that varies between 96% to 99% using 7-15 features instead of 50 features (Figure2). Based on the result analysis, it is suggested that the following factors related to the crash and might affect the fatality rate are: the minute which the crash occurred, the event that resulted in the most severe injury, the location of the crash as it relates to its position within or outside the traffic way based on the “First Harmful the Event”, if a school bus, or motor vehicle functioning as a school bus is involved, the number of drinking drivers involved in the crash. We agree with previous studies [17,18] which indicated that human factors and the collision type strongly affect the fatal rate more than the environmental factors. The results obtained, might help in considering and evaluating the factors related to fatal accidents.

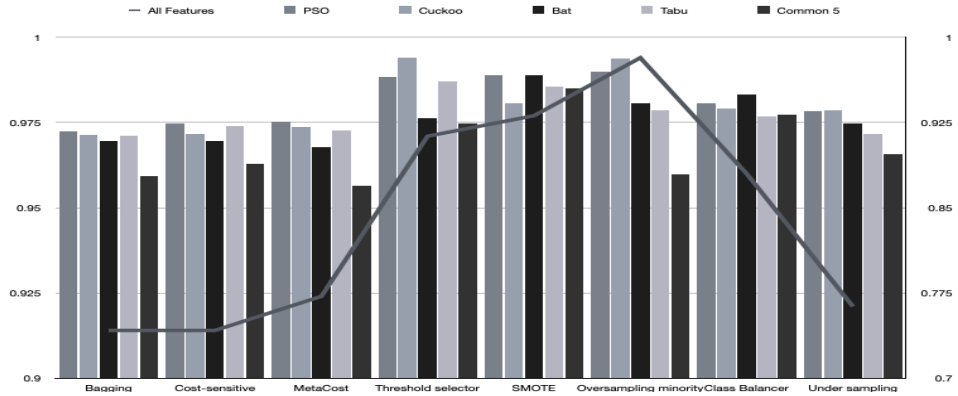

Figure 2. Performance comparison of the different learning paradigms

5. CONCLUSION

In this paper, we presented a computational method that can accurately determine the key features that influence in classification in FARS dataset. For that reason, we applied and compared the prediction performances of several meta-classifiers. Moreover, several data-oriented approaches were applied to handle the uneven class ratios problem. Results from the study showed that the Threshold Selection Classifier and Over-sampling technique had the better predictive ability among the other techniques with using Random Forest as a base-classifier. Experiments on the FARS dataset empirically proved that our proposed method can reduce the number of features with almost 90% and obtain satisfactory results.

REFERENCES

[1] World Health Organization- WHO, Global status report on road safety, Available from World Wide Web: https://www.who.int/violence_injury_prevention/road_safety_status/2018/en/ (accessed December 2019).

[2] Qiu, C., Wang, C., Fang, B.& Zuo, X., (2014). A multiobjective particle swarm optimization-based partial classification for accident severity analysis. Appl. Artif.Intell.28,555–576.

[3] Kwon, O.H.; Rhee, W.; & Yoon, Y. (2015). Application of classification algorithms for analysis of road safety risk factor dependencies. Accident Analysis and Prevention, 75, 1-15.

[4] Kolyshkina, I., & Brookes, R. (2002). Data mining approaches to modeling insurance risk. Retrieved December 14, 2019, from http:// www.salford-systems.com/doc/insurance.pdf

[5] Li, Y., Guo, H., Xiao, L., Yanan, L., & Jinling, L, (2016) Adapted ensemble classification algorithm based on multiple classifier system and feature selection for classifying multi-class imbalanced data, Knowledge-Based Systems, 94, 88–104.

[6] Trac Integrated SCM & Project Management. Fatal Accidents Dataset, https://wiki.csc.calpoly.edu/datasets/wiki/ HighwayAccidents.

[7] U.S. Department of Transportation, FARS analytic reference guide, 1975 to 2009. https://crashstats.nhtsa.dot.gov/Api/Public/ ViewPublication/811352, 2010.

[8] Eric M Ossiander & Peter Cummings, (2002), Freeway speed limits and traffic fatalities in washington state. Accident Analysis & Prevention, 34(1):13–18.

[9] KMA Solaiman, Md Mustafizur Rahman & Nashid Shahriar, (2013), Avra Bangladesh collection, analysis & visualization of road accident data in Bangladesh, In Proceedings of International Conference on Informatics, Electronics & Vision, 1–6.

[10] Sachin Kumar & Durga Toshniwal, (2015), Analysing road accident data using association rule mining. In Proceedings of International Conference on Computing, Communication and Security, 1–6.

[11] Chang L. & H. Wang, (2006), "Analysis of traffic injury severity: An application of non-parametric classification tree techniques Accident analysis and prevention", Accident analysis and prevention, Vol. 38(5), pp 1019- 1027.

[12] S. Krishnaveni & M. Hemalatha, (2011), A perspective analysis of traffic accident using data mining techniques. International Journal of Computer Applications, 23(7):40–48.

[13] Kwon, O.H.; Rhee, W. & Yoon, Y. (2015). Application of classification algorithms for analysis of road safety risk factor dependencies. Accident Analysis and Prevention, 75, 1-15.

[14] T. K. Bahiru, D. Kumar Singh & E. A. Tessfaw, (2018), "Comparative Study on Data Mining Classification Algorithms for Predicting Road Traffic Accident Severity," Second International Conference on Inventive Communication and Computational Technologies (ICICCT), Coimbatore, 2018, 1655-1660.

[15] Silva, HCE & Saraee, MH, (2019), Predicting road traffic accident severity using decision trees and time-series calendar heatmaps , in: The 6th IEEE Conference on Sustainable Utilization and Development in Engineering and Technology (2019 IEEE CSUDET), 7 - 9 November 2019, Penang, Malaysia.

[16] Mujalli, R.O.; López, G. & Garach, L. (2016). Bayes classifiers for imbalanced traffic accidents datasets. Accident Analysis and Prevention, 88, 37-51.

[17] Liling Li, Sharad Shrestha & Gongzhu Hu, (2017), "Analysis of Road Traffic Fatal Accidents Using Data Mining Techniques", Software Engineering Research, Management and Applications (SERA), 2017 IEEE 15th International Conference, DOI: 10.1109/SERA.2017.7965753
[18] Pakgohar A., Tabrizi R. S., Khalili M. & Esmaeili A., (2011), The role of human factor in incidence and severity of road crashes based on the cart and LR regression: a data mining approach. Procedia Computer Science, World Conference on Information Technology, 3, pp. 764– 769.
[19] Kotsiantis S, Kanellopoulos D & Pintelas P, (2006) Handling imbalanced datasets: a review. GESTS Int Trans Comput Sci Eng 30(1):25–36
[20] López V, Fernández A, Moreno-Torres JG & Herrera F, (2012), Analysis of preprocessing vs. cost-sensitive learning for imbalanced classification: open problems on intrinsic data character- istics. Expert Syst Appl 39:6585–6608. https://doi.org/10.1016/j. eswa.2011.12.043
[21] Schierz AC, (2009), Virtual screening of bioassay data. J Chemin- form 1:21. https://doi.org/10.1186/1758-2946-1-21
[22] Lin W-J & Chen JJ, (2013), Class-imbalanced classifiers for high-dimensional data. Brief Bioinform 14:13–26. https://doi. org/10.1093/bib/bbs006.
[23] N. D. Singh, (2018), "Clustering and learning from imbalanced data", arXiv preprint arXiv:1811.00972.
[24] Dietterich, T.G., (2000), An Experimental Comparison of Three Methods for Constructing Ensembles of Decision Trees: Bagging, Boosting, and Randomization, Machine Learning, 40, 139-157. http://dx.doi.org/10.1023/A:1007607513941
[25] Breiman L, (2001), Random Forests, Mach Learn 45:5–32
[26] L. Breiman, Bagging predictors, Machine Learning, 24:123–140, 1996. L. Breiman. Some Infinity Theory for Predictor Ensembles, Technical Report 577, UC Berkeley, 2000. URL http://www.stat.berkeley.edu/˜breiman.
[27] Oshiro TM, Perez PS & Baranauskas JA, (2012), How many trees in a Random Forest?, Machine learning and data mining in pattern recognition, Springer, Berlin, pp 154–168.
[28] Sun Y, Kamel MS, Wong AKC & Wang Y., (2007), Cost-sensitive boosting for classification of imbalanced data. Pattern Recognition, 40: 3358–3378.
[29] Guo Haixiang, Li Yijing, Jennifer Shang, Gu Mingyun, Huang Yuanyue & Gong Bing, (2016), Learning from class-imbalanced data: Review of methods and applications, Expert Systems with Applications.
[30] Kotsiantis S, Kanellopoulos D & Pintelas P, (2006), Handling imbalanced datasets: a review, GESTS Int Trans Comput Sci Eng 30(1):25–36.
[31] Chawla NV, Japkowicz N & Kotcz A, (2004), Editorial: special issue on learning from imbalanced data sets. SIGKDD Explor Newsl 6:1–6. https://doi.org/10.1145/1007730.1007733.
[32] Powers D, (2011), Evaluation: from precision, recall and f-measure to roc., informedness, markedness & correlation. J Mach Learn Technol 2:37–63.
[33] Domingos, P., (1999), MetaCost: A general method for making classifiers cost-sensitive, In Proceedings of the 15th international conference on knowledge discovery and data mining, 155–164.
[34] Quinlan, J.R. (1996), Bagging, boosting, and C4.5, in Proceedings of the 13th National Conference on Artificial Intelligence, pp. 725-730.
[35] Schapire, R.E., Y. Freund, P. Bartlett, & W.S. Lee (1997), Boosting the margin: A new explanation for the effectiveness of voting methods, in Proceedings of the Fourteenth International Conference on Machine Learning, 322-330.
[36] Bauer, E. & Kohavi, R. (1999), An Empirical Comparison of Voting Classification Algorithms: Bagging, Boosting, and Variants. Machine Learning, Kluwer Academic Publishers, 36, 105-139.
[37] F. Glover, (1989), "Tabu search—part I," ORSA Journal on Computing, vol. 1, no. 3, 190–206.
[38] F. Glover, (1990), "Tabu search—part II," ORSA Journal on Computing, vol. 2, no. 1, 4–32,.
[39] Tahir, M.A., Bouridane, A., Kurugollu, F. & Amira, A., (2004), Feature Selection using Tabu Search for Improving the Classification Rate of Prostate Needle Biopsies, In: Proc. 17th International Conference on Pattern Recognition (ICPR 2004), Cambridge, UK.
[40] Tahir, M.A., Bouridane, A., Kurugollu, F., 2004b. Simulataneous Feature Selection and Weigthing for Nearest Neighbor Using Tabu Search. In: Lecture Notes in Computer Science (LNCS 3177), 5th International Conference on Intelligent Data Engineering and Automated Learning (IDEAL 2004), Exeter, UK.

[41] Korycinski, D., Crawford, M., Barnes, J.W & Ghosh, J., (2003), Adaptive feature selection for hyperspectral data analysis using a binary hierarchical classifier and Tabu Search, In: Proceedings of the IEEE International Geoscience and Remote Sensing Symposium, IGARSS.
[42] J. Kennedy &R.C. Eberhart, (2001), "Swarm intelligence", Morgan Kaufmann.
[43] J. Kennedy & R. Eberhart, (1997), "A discrete binary version of the particle swarm algorithm", Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, Vol.5, 4104–4108.
[44] Xin-She Yang & Suash Deb, (2009), Cuckoo search via Lévy flights. In Nature & Biologically Inspired Computing, 2009. NaBIC 2009. World Congress on, 210–214.
[45] Xin-She Yang & Suash Deb, (2014), "Cuckoo search: recent advances and applications," Neural Computing and Applications, vol. 24, no. 1, 169–174.
[46] X.-S. Yang, (2010), A new metaheuristic bat-inspired algorithm. In Nature Inspired Cooperative Strategies for Optimization (NICSO 2010), volume 284 of Studies in Computational Intelligence, Springer Berlin Heidelberg, 65–74.
[47] Powers, D.M.W., (2011), Evaluation: From Precision, Recall and F-Measure to ROC. Inf. Mark. Correl. J. Mach. Learn. Technol. 2, 37–63.
[48] Hall M, Frank E, Holmes G et al (2009) The WEKA data mining software: an update. SIGKDD Explor Newsl 11:10–18. https:// doi.org/10.1145/1656274.1656278
[49] Oshiro TM, Perez PS, Baranauskas JA (2012) How many trees in a Random Forest? In: Machine learning and data mining in pattern recognition. Springer, Berlin, pp 154–168.