Contents lists available at SCCE

Journal of Soft Computing in Civil Engineering

Journal homepage: www.jsoftcivil.com

# A Real-Time Warning System for Rear-End Collision Based on Random Forest Classifier

**F. Teimouri[1], M. Ghatee[1*]**

1. Department of Computer Science, Faculty of Mathematics and Computer Science, Amirkabir University of Technology, Tehran, Iran

Corresponding author: *ghatee@aut.ac.ir*

https://doi.org/10.22115/SCCE.2020.217605.1172

## ARTICLE INFO

Article history:
Received: 28 January 2020
Revised: 07 March 2020
Accepted: 07 March 2020

Keywords:
Rear-end collision;
Driver assistant systems;
Data mining;
Classification algorithms;
TOPSIS.

## ABSTRACT

Rear-end collision warning system has a great role to enhance driving safety. In this system, some measures are used to evaluate the safety and in the case of dangerous, the system warns drivers. This system should be executed in real-time, to remain enough time to avoid collision with the front vehicle. To this end, in this paper, a new system is developed by using a random forest classifier to extract knowledge about warning and safe situations. This knowledge can be extracted from accidents and vehicle trajectory data. Since the data of these situations are imbalanced, a combination of cost-sensitive learning and classification methods was used to improve the sensitivity, specificity, and processing time of classification. To evaluate the performance of this system, vehicle-trajectory-data of 100 cars that have been provided by Virginia tech transportation institute, are used. The comparison results are given in terms of accuracy and processing time. By using TOPSIS multi-criteria selection method, it is shown that the implemented classifier is better than different classifiers including Bayesian network, Naive Bayes, MLP neural network, support vector machine, k-nearest neighbor, rule-based methods and decision tree. The implemented random forest gets 88.4% accuracy for detection of the dangerous situations and 94.7% for detection of the safe situations. Also, the proposed system is more robust compared with the perceptual-based and kinematic-based algorithms.

How to cite this article: Teimouri F, Ghatee, M. A real-time warning system for rear-end collision based on random forest classifier. J Soft Comput Civ Eng 2020;4(1):49–71. https://doi.org/10.22115/scce.2020.217605.1172.

# 1. Introduction

Road safety is an important subject all over the world. The general parameters influential on road accidents are driver, vehicle, road, and environment [1]. According to the recent research studies about the accidents' reasons, human fault affects almost 93% of accidents and it is the main reason in 75% of accidents [2]. Using the safety systems installed on vehicles, the rate of accidents can be decreased. These safety systems are divided into two groups, passive safety systems, and active safety systems [3]. Usually, the passive safety systems including seat-belt and airbag are limited and expensive. Currently, active systems like an advanced driver assistance system (ADAS) has been followed by different vehicle manufacturers [4]. The main purpose of this system is to reduce the driver's faults by warning dangerous situations. Collision warning system as a subsystem of ADAS helps drivers to evaluate non-obvious dangerous situations to reduce related human's faults.

According to the wide variety of car accidents, NHTSA reported 32.9 % of car accidents as rear-end collisions [5]. Thus, in the current paper an intelligent and online ADAS is proposed for rear-end collision warning, see e.g. [6–10] to see the effects of such systems. The processing time of a rear-end collision warning is very important in its effects. A poorly timed warning may actually undermine driver safety [11]. Too soon or too frequent alarms (false alarms) bother driver and too late or missed alarms decrease the effectiveness of the system [12]. As a result, finding a balance between opportune alarming (not too soon, not too late) and detecting dangerous situations are important for developing a useful system and they are considered in this paper.

Following [13–15] and [16], we have classified the existing algorithms for rear-end collision warnings into two groups including perceptual-based and kinematic-based. In the former, there are some criteria to evaluate driver performance. For each criterion, there is a threshold value and when the value of criterion is less than that threshold, warning will be issued. Time-to-collision (TTC) [17] and time-gap (TG) [18] are two criteria for analyzing the performance of a driver who is following a preceding vehicle [19]. A variety of existing studies have attempted to identify rear-end collision situations using time-to-collision, time-gap or both of them. In [20] and in Chapter 1 of [21], two warning distances between vehicles based on a critical threshold for TTC were defined. They have been referred to as Honda algorithm and Hirst and Graham algorithm. The authors of [22] and [23] proposed a vision based forward collision warning by using TTC and a possible collision course to trigger a warning. TTC was used in [24] to judge collision threat and to estimate the potential effectiveness of a forward collision avoidance system with a forward-collision warning system, a brake assist, and autonomous braking functionality. In [25], a methodology was proposed to estimate rear-end crash probability based on an exponential decay function using TTC. The authors of [26] used a non-dimensional warning index and TTC for determining driving situations and the main idea of this system is to maintain a specified TG between vehicles. In [27], a fuzzy rear-end collision warning system was developed using TTC and TG in order to warn a driver of a possible collision.

On the other hand, the systems based on the kinematic information of two vehicles, take pre-determined driver's reaction time and maximum deceleration rate to calculate safe distance. The

safe distance which is calculated differently is taken as a critical threshold and when the distance between two vehicles is less than this threshold, the system warns. In the researches of [28,29] and [12], three warning distances were calculated between vehicles based on different scenarios. The constructed algorithms were entitled as "stop distance algorithm", "Mazda algorithm" and "PATH algorithm". A vision-based ADAS with forward-collision warning function was also developed in [30] applying a headway distance estimation model to detect the potential forward collision. The concept of the forward-collision warning based on distance was also proposed in [31]. In [32], a rear-end collision risk index was extended using safety distance and then a fuzzy-clustering algorithm was applied to identify the rear-end collision risk levels.

To summarize the scientific gaps of the previous algorithms, one can note that the accuracy of the detection of the dangerous and the safe situations was not acceptable because of using some constants and pre-defined values for parameters and threshold values. Really, a lot of researches on rear-end collisions detection and alarming exist, but there is not enough attention to extracting the knowledge about the warning and safe situations. Such knowledge can be extracted from accidents and vehicle trajectory data. Data mining algorithms can be used to extract patterns from vehicle trajectory data to define the threshold values for parameters and to find a useful criterion adaptable to reality. This is the most important contribution of this paper to use data mining to enhance the rear-end collision warning system.

Secondly, the data of collision warning is imbalanced and the usual classification techniques cannot provide reasonable results [33].

Thirdly, the existing techniques consider the fixed values as the thresholds to split the warning and the safe classes, but these thresholds should be adapted with the problem.

To cover, these gaps, this paper proposes a new classification system by considering the random forest as the classifier together a cost-sensitive learning mechanism [34,35] to improve the results. The structure of the paper is as follows. The proposed methodology for the rear-end collision warning system is presented in the next section. Some details are dedicated to optimizing the classification of movement situations into dangerous and safe classes. Section 3 presents the evaluation of the proposed methodology. The final section ends the paper with a brief conclusion.

## 2. Rear-end collision warning system

The vehicles continuously interact with the other vehicles to realize car-following and lane-changing, etc. When these interactions are not stable, collision possibly happens. Therefore it is possible to evaluate the potential of collision with analyzing the vehicle’s motion and unsafe situations[24] . Data mining techniques are appropriate tools to detect warning situations based on the vehicle's trajectory data. These methods are used to analyze a large volume of data and to extract patterns and rules to extract accident knowledge [36]. Among data mining techniques, classification algorithms are the most famous techniques for knowledge extraction and they are considered in the current paper to detect warning situations for rear-end collisions and safe

situations. In our study, 100 car's database is considered for constructing and testing algorithms [37]. This dataset includes 100 sets of naturalistic cars' trajectories that were conducted in the Northern Virginia / Washington, D.C. area over a 2 year period. To collect the data, no special instructions were given to drivers in the absence of the experimenters. The vehicles were instrumented with different sensors containing the forward and rearward radar, lateral and longitudinal accelerometers, gyro, GPS, access to the vehicle CAN, and five channels of compressed digital video. Collection rates for the various sensors ranged from 1Hz to 10Hz. This dataset contains approximately 2,000,000 vehicle miles and 43,000 hours of driving data. Detailed video analysis has been compiled for 68 crashes and 760 near-crashes. See [37] for more details. To classify the situations, we follow the steps given in Figure 1. The details of the modules will be discussed in the next subsections.

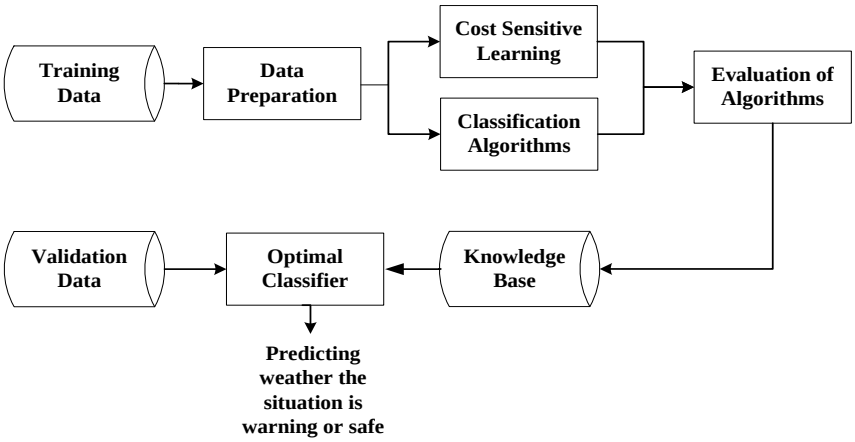



**Fig. 1.** The structure of the proposed rear-end collision warning system.

## 2.1. Cost-sensitive learning module

Usually, a cost-sensitive learning method is used on databases with imbalanced class distribution [34,35,38]. In cost-sensitive learning, the class with fewer elements is considered as a positive class and the other with more elements is taken as a negative class. Often misclassification of actual positive classes that are predicted as a negative class, is greater than the misclassification of actual negative classes that are predicted as positive class [39]. The cost of classification can be calculated with a confusion matrix given in Table 1.

**Table 1**
Confusion/Cost matrix for a two-class problem.

| | **Positive prediction** | **Negative prediction** |
|---|---|---|
| **Positive class** | True Positive (TP) | False Negative (FN) |
| **Negative class** | False Positive (FP) | True Negative (TN) |

In the database which is used for classification, a cost-sensitive learning model is used, because the amount of data for safe situations is 5 times more than the warning situations. The cost matrix is demonstrated in Table 2.

**Table 2**
Confusion/Cost matrix for warning and safe situations problem.

| | Warning prediction | Safe prediction |
|---|---|---|
| **Warning class** | 0 | 5 |
| **Safe class** | 1 | 0 |

## 2.2. Classification algorithms module

All classification methods can be used to find a relation between input and output properties. In the proposed rear-end collision warning system, we implement a random forest classifier which is an ensemble classifier including several unpruned trees. Each tree is independently built based on bootstrap samples of training dataset and the best split at each node is calculated within a randomly selected subset of descriptors. After trees are grown, predictions for test data are made by the majority voting of ensemble tees. For more information about this classifier, one can refer [40–42]. In the next section, it is shown that this algorithm is preferable on other classifiers for the proposed warning system. For this aim, classification algorithms in Weka are called. Details of some examined classification methods are given as follows:

- Bayesian network: It is a directed acyclic graph (DAG) that illustrates a factorization of a joint probability distribution (JPD). Given a sufficiently large dataset, the Bayesian network can learn by structural learning and parameter learning. More details can be found in[43].
- Naïve Bayes: It is a simple probabilistic method that assumes class conditional independence. This classifier predicts the class with the highest posterior probability which is calculated by Bayes theorem. For more study, see [44].
- Multi-layer perceptron: It is a feed-forward neural network with one or multiple hidden layers including different numbers of neurons. The best algorithm for tuning the weights in this network is the backpropagation algorithm [45,46].
- Support Vector Machine: It uses a sequential optimization algorithm to find an optimal hyperplane that correctly classifies data points by separating the points of two classes as much as possible [47].
- K-nearest neighbor: Among different methods of supervised learning, the nearest neighbor has the highest efficiency because there is no preliminary assumption about training data distribution. For details, see [48].
- JRip rule-based (RIPPER[1]): It is a rule-based method in which there is a set of rules and if a sample has the properties of one of the rules, that sample is a member of rule's class otherwise it would not be counted as a rule's class. The RIPPER algorithm is a two stages algorithm to reduce errors by iteratively pruning the design space and extracting the knowledge of rules. For more research, see [49].

[1] Repeated Incremental Pruning to Produce Error Reduction

- C4.5 decision tree: It constructs a decision tree in a top-down manner with a divide and conquer strategy. The construction begins by evaluating each attribute using a criterion known as the information gain ratio to determine the best attribute for classifying the training samples. This attribute is chosen for the root node of the tree. Then the decision tree splits the data into subsets according to the value of the chosen attribute, and the process repeats for each child. Details of C4.5 have been given in [50,51].

### 2.3. Evaluation measures

To evaluate the results of the different classifiers, the data set is divided into training data set and validation data. Accuracy criterion (Eq. (1)), sensitivity criterion (Eq. (2)), specificity criterion (Eq. (3)) and also processing time of building classifier model and classifying validation data are used to compare the results in this paper. They are usually used in machine learning and data mining procedures, see e.g., [36].

$$Accuracy = \frac{TruePositive + TrueNegative}{TruePositive + TrueNegative + FalsePositive + FalseNegative} \tag{1}$$

$$\text{Sensitivity} = \frac{TruePositive}{TruePositive + FalseNegative} \tag{2}$$

$$\text{Specificity} = \frac{TrueNegative}{TrueNegative + FalsePositive} \tag{3}$$

Moreover to evaluate the different methods in the proposed system, the properties of data samples are used for the data mining process. These properties include the follower vehicle's speed, the relative distance between two vehicles (follower and leader vehicles), the relative speed between two vehicles, time to collision and the time gap. The time-to-collision (TCC) is the time interval between two vehicles colliding together without changing the movement direction and speed. TTC is calculated by Eq. (4).

$$TTC(t) = \frac{\Delta x}{-\Delta v} = \frac{x_l - x_f}{v_f - v_l} \tag{4}$$

In addition, based on Eq. (5), the time gap (TG) is the time interval between the follower and the leader vehicles without any change in the situations.

$$TG(t) = \frac{\Delta x}{v_f} = \frac{x_l - x_f}{v_f} \tag{5}$$

## 3. Experimental results

### 3.1. The experimental data

In this section, we have used 100 car's database from Virginia tech transportation institute [37,52] to examine the data of vehicles trajectory. This database includes information about 68 crashes and 760 near-crashes. In these near-crashes, a fast movement such as braking or lane changing avoids an accident. To use crashes' data, it is possible to use reinforcement learning. In reinforcement learning, the behaviors aligned with final goals will be rewarded and the other

behavior farther from the final goal will be penalized. For the cases that the driver's reaction is lane changing, the other lane properties should be taken into account to analyze the risky behavior. However, such data are not given in 100 car's database. So in the experiment of the current paper, we use the near-crashes data as the rear-end collisions where a driver's reaction is braking. After such preprocess on 100 car's database, we have obtained 39938 data samples. The properties of these samples are used for the data mining process in the proposed system as mentioned in Subsection 2.3.

Moreover, to install supervised learning methods, the following steps are repeated:

1. The sample data are divided into two classes: warning (dangerous) class or safe class.
2. Each near-crashes' data includes 30 seconds trajectory before the event, the event and 10 seconds trajectory after the event. Event means the time interval between leader vehicle's starting to stop, therefore rear-end collision starts to happen, until when the follower's driver does some fast actions like braking to avoid colliding the leader vehicle.
3. Obviously, before and after the event, there is not necessary to warn, however, the warning is necessary during the event. So, the warning level is considered as zero or one. Zero is assigned to all time-interval with 30 seconds before the event and 10 seconds after the event. One is also assigned to the time interval of the event.

## 3.2. The selection methodology for classifiers

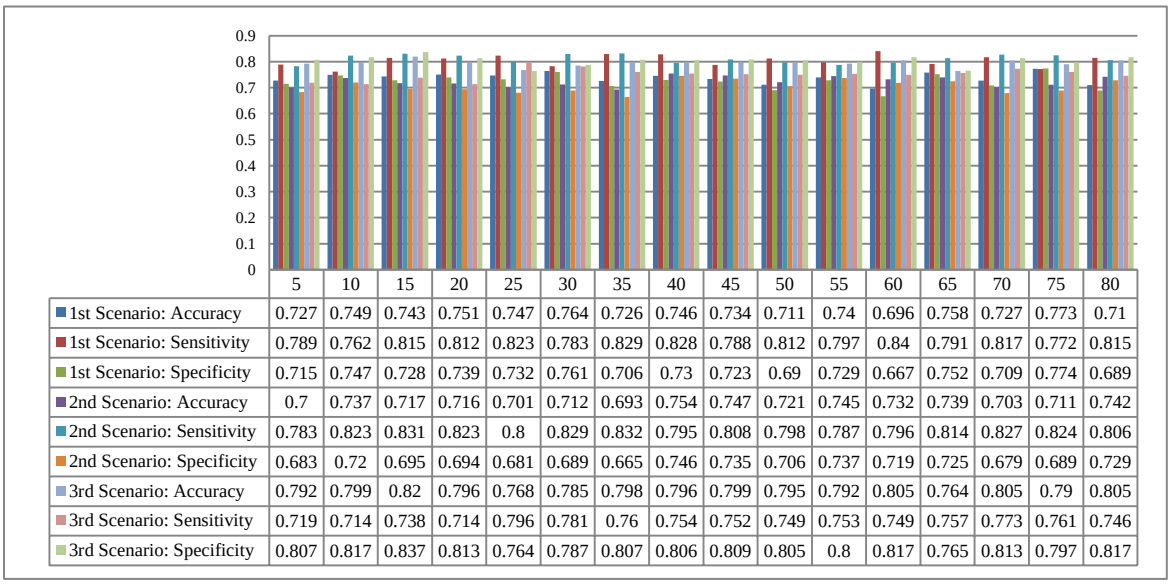
To tune up the classifiers and to compare the results of different classifiers, the dataset is divided into a training dataset and a validation dataset to create and to validate the models, respectively. In this study, to guarantee independence judge between the results and the data, three scenarios are considered with three random selection with 65%, 70% and 80% of the total data. The remaining data is used as validation data. Then to select the best classifier for each scenario, based on the assigned weights to all criteria, it is possible to use TOPSIS technique[53], which is a famous method for solving multi-criteria decision making problems. To assign weights to the different criteria, there are different cases. In this study, the following 4 assumptions on the weights are considered:

- Assumption 1: All three criteria have the same weight and importance,
- Assumption 2: The specificity and sensitivity are important and have the same weights,
- Assumption 3: All three criteria are important but specificity has greater weight than sensitivity and sensitivity hast greater weight than processing time,
- Assumption 4: All three criteria are important but sensitivity has greater weight than specificity and specificity has greater weight than processing time.

## 3.3. Experiment results on multi-layer neural network

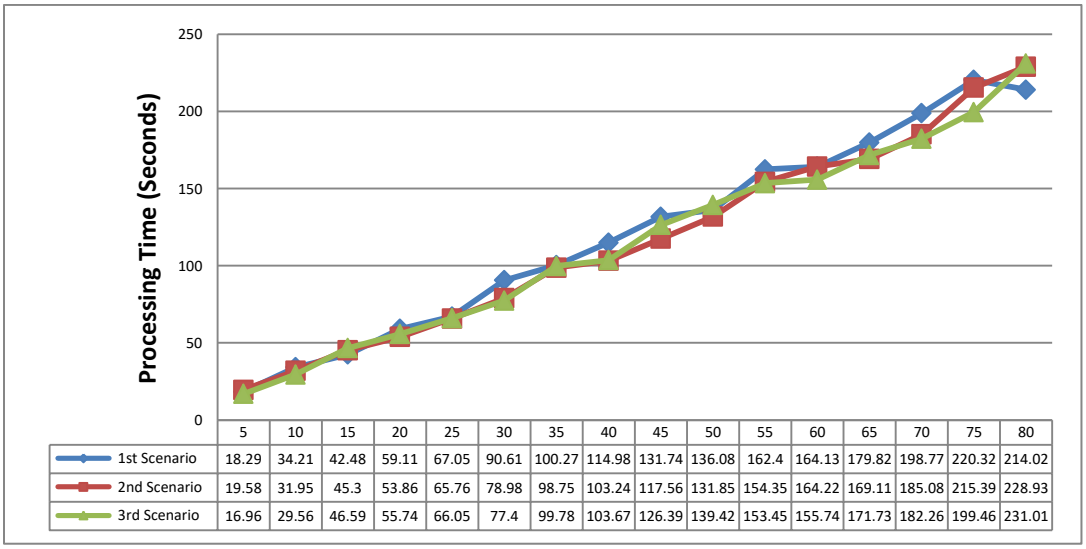
An artificial neural network with a single hidden layer can be used to approximate every non-linear function by a pre-determined precision degree [54], however, the number of neurons in the hidden layer is not easy to obtain. In this study, to obtain a sufficient number of neurons in the hidden layer, a trial-and-error method was used and the performances were compared. In what

follows, the different numbers (5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70, 75, 80) are compared. Figure 2 shows the simulation results for the three-layer perceptron. Obviously, the highest accuracy and specificity of the first scenario happen for a neural network with 75 neurons in the hidden layer and the highest sensitivity occurs for a network with 60 neurons in hidden layer. For the second scenario, the highest accuracy and specificity happen for a network with 40 neurons in the hidden layer and the highest sensitivity occurs for 35 neurons. In the third scenario, the highest accuracy and specificity happen with 15 neurons in the hidden layer and the highest sensitivity is for 25 neurons. In addition, Figure 3 compares the processing time for three scenarios in which the lowest processing time happens for 5 neurons.




| | 5 | 10 | 15 | 20 | 25 | 30 | 35 | 40 | 45 | 50 | 55 | 60 | 65 | 70 | 75 | 80 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1st Scenario: Accuracy | 0.727 | 0.749 | 0.743 | 0.751 | 0.747 | 0.764 | 0.726 | 0.746 | 0.734 | 0.711 | 0.74 | 0.696 | 0.758 | 0.727 | 0.773 | 0.71 |
| 1st Scenario: Sensitivity | 0.789 | 0.762 | 0.815 | 0.812 | 0.823 | 0.783 | 0.829 | 0.828 | 0.788 | 0.812 | 0.797 | 0.84 | 0.791 | 0.817 | 0.772 | 0.815 |
| 1st Scenario: Specificity | 0.715 | 0.747 | 0.728 | 0.739 | 0.732 | 0.761 | 0.706 | 0.73 | 0.723 | 0.69 | 0.729 | 0.667 | 0.752 | 0.709 | 0.774 | 0.689 |
| 2nd Scenario: Accuracy | 0.7 | 0.737 | 0.717 | 0.716 | 0.701 | 0.712 | 0.693 | 0.754 | 0.747 | 0.721 | 0.745 | 0.732 | 0.739 | 0.703 | 0.711 | 0.742 |
| 2nd Scenario: Sensitivity | 0.783 | 0.823 | 0.831 | 0.823 | 0.8 | 0.829 | 0.832 | 0.795 | 0.808 | 0.798 | 0.787 | 0.796 | 0.814 | 0.827 | 0.824 | 0.806 |
| 2nd Scenario: Specificity | 0.683 | 0.72 | 0.695 | 0.694 | 0.681 | 0.689 | 0.665 | 0.746 | 0.735 | 0.706 | 0.737 | 0.719 | 0.725 | 0.679 | 0.689 | 0.729 |
| 3rd Scenario: Accuracy | 0.792 | 0.799 | 0.82 | 0.796 | 0.768 | 0.785 | 0.798 | 0.796 | 0.799 | 0.795 | 0.792 | 0.805 | 0.764 | 0.805 | 0.79 | 0.805 |
| 3rd Scenario: Sensitivity | 0.719 | 0.714 | 0.738 | 0.714 | 0.796 | 0.781 | 0.76 | 0.754 | 0.752 | 0.749 | 0.753 | 0.749 | 0.757 | 0.773 | 0.761 | 0.746 |
| 3rd Scenario: Specificity | 0.807 | 0.817 | 0.837 | 0.813 | 0.764 | 0.787 | 0.807 | 0.806 | 0.809 | 0.805 | 0.8 | 0.817 | 0.765 | 0.813 | 0.797 | 0.817 |

**Fig. 2.** Accuracy, sensitivity and specificity of three layer perceptron with different hidden neurons.




| | 5 | 10 | 15 | 20 | 25 | 30 | 35 | 40 | 45 | 50 | 55 | 60 | 65 | 70 | 75 | 80 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1st Scenario | 18.29 | 34.21 | 42.48 | 59.11 | 67.05 | 90.61 | 100.27 | 114.98 | 131.74 | 136.08 | 162.4 | 164.13 | 179.82 | 198.77 | 220.32 | 214.02 |
| 2nd Scenario | 19.58 | 31.95 | 45.3 | 53.86 | 65.76 | 78.98 | 98.75 | 103.24 | 117.56 | 131.85 | 154.35 | 164.22 | 169.11 | 185.08 | 215.39 | 228.93 |
| 3rd Scenario | 16.96 | 29.56 | 46.59 | 55.74 | 66.05 | 77.4 | 99.78 | 103.67 | 126.39 | 139.42 | 153.45 | 155.74 | 171.73 | 182.26 | 199.46 | 231.01 |

**Fig. 3.** Processing Time of three layer perceptron with different hidden neurons.

In the presented cases, for each scenario, there is a decision matrix including 16 alternatives and 3 different criteria. Alternatives are the number of hidden neurons and the criteria are the sensitivity, the specificity and the processing time. Accuracy is not taken as a criterion into account because it is a combination of sensitivity and specificity. The decision matrix values can be extracted from Figure 2 and Figure 3. For all of the assumptions, the weights of criteria are given in Table 3.

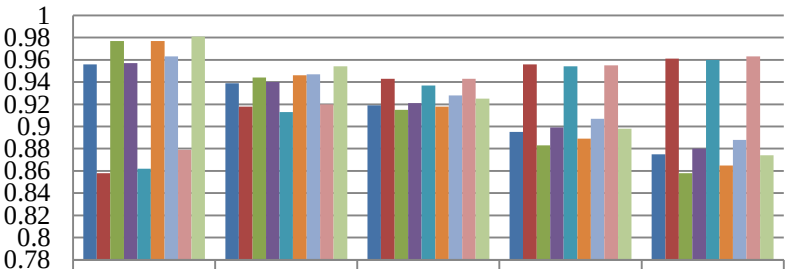
**Table 3**
Weight matrix for different criteria for the considered assumptions and the best number of neurons in the hidden layer.

| Assumptions Index | Sensitivity | Specificity | Processing Time | Third Scenario | Second Scenario | First Scenario |
|---|---|---|---|---|---|---|
| **1** | $\frac{1}{3}$ | $\frac{1}{3}$ | $\frac{1}{3}$ | 5 neurons | 10 neurons | 5 neurons |
| **2** | $\frac{1}{2}$ | $\frac{1}{2}$ | 0 | 70 neurons | 45 neurons | 20 neurons |
| **3** | $\frac{2}{6}$ | $\frac{3}{6}$ | $\frac{1}{6}$ | 5 neurons | 10 neurons | 10 neurons |
| **4** | $\frac{3}{6}$ | $\frac{2}{6}$ | $\frac{1}{6}$ | 5 neurons | 10 neurons | 5 neurons |

Applying TOPSIS method for all of the mentioned assumptions shows that the best number of neurons of the hidden layer of the corresponding neural networks is different. See0including the best number of hidden neurons based on the presented experiments.

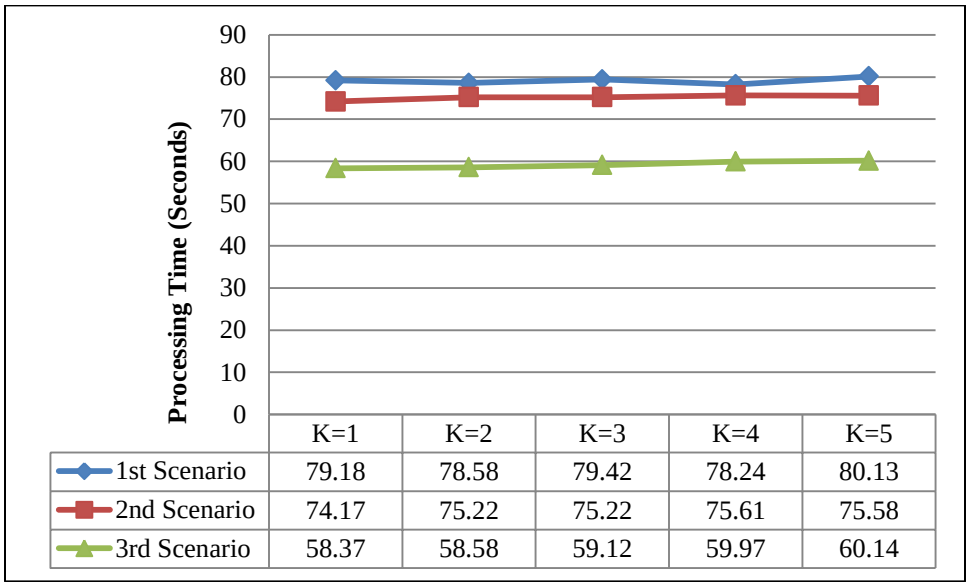
## 3.4. Experiment results on k-nearest neighbor method

In the k-nearest neighbor method, it is possible to find the best k to find reasonable results. To this aim, we simulate this classifier for different k and for the previously mentioned three different scenarios. Then, for each scenario, the best value of k has been obtained. The weights of criteria are similarly defined based on the assumptions given in Table 3. Figure 4 shows the simulation results for the k-nearest neighbor. Obviously, in all three scenarios, the greatest accuracy and specificity are obtained for k=1 and the maximum sensitivity is fetched for k=5. Thus as much as k increases, the accuracy, and the specificity decreases and the sensitivity increases. In Figure 5 the processing times of all different k-nearest neighbor methods for all of the scenarios and different k are illustrated. As one can note that there is not a great difference between these experiments in terms of processing times.




| | K=1 | K=2 | K=3 | K=4 | K=5 |
|---|---|---|---|---|---|
| 1st Scenario: Accuracy | 0.956 | 0.939 | 0.919 | 0.895 | 0.875 |
| 1st Scenario: Sensitivity | 0.858 | 0.918 | 0.943 | 0.956 | 0.961 |
| 1st Scenario: Specificity | 0.977 | 0.944 | 0.915 | 0.883 | 0.858 |
| 2nd Scenario: Accuracy | 0.957 | 0.94 | 0.921 | 0.899 | 0.88 |
| 2nd Scenario: Sensitivity | 0.862 | 0.913 | 0.937 | 0.954 | 0.96 |
| 2nd Scenario: Specificity | 0.977 | 0.946 | 0.918 | 0.889 | 0.865 |
| 3rd Scenario: Accuracy | 0.963 | 0.947 | 0.928 | 0.907 | 0.888 |
| 3rd Scenario: Sensitivity | 0.879 | 0.92 | 0.943 | 0.955 | 0.963 |
| 3rd Scenario: Specificity | 0.981 | 0.954 | 0.925 | 0.898 | 0.874 |

**Fig. 4.** Accuracy, sensitivity and specificity of k-nearest neighbor classifier with different k.




| | K=1 | K=2 | K=3 | K=4 | K=5 |
|---|---|---|---|---|---|
| 1st Scenario | 79.18 | 78.58 | 79.42 | 78.24 | 80.13 |
| 2nd Scenario | 74.17 | 75.22 | 75.22 | 75.61 | 75.58 |
| 3rd Scenario | 58.37 | 58.58 | 59.12 | 59.97 | 60.14 |

**Fig. 5.** Processing time of k-nearest neighbor classifier with different k

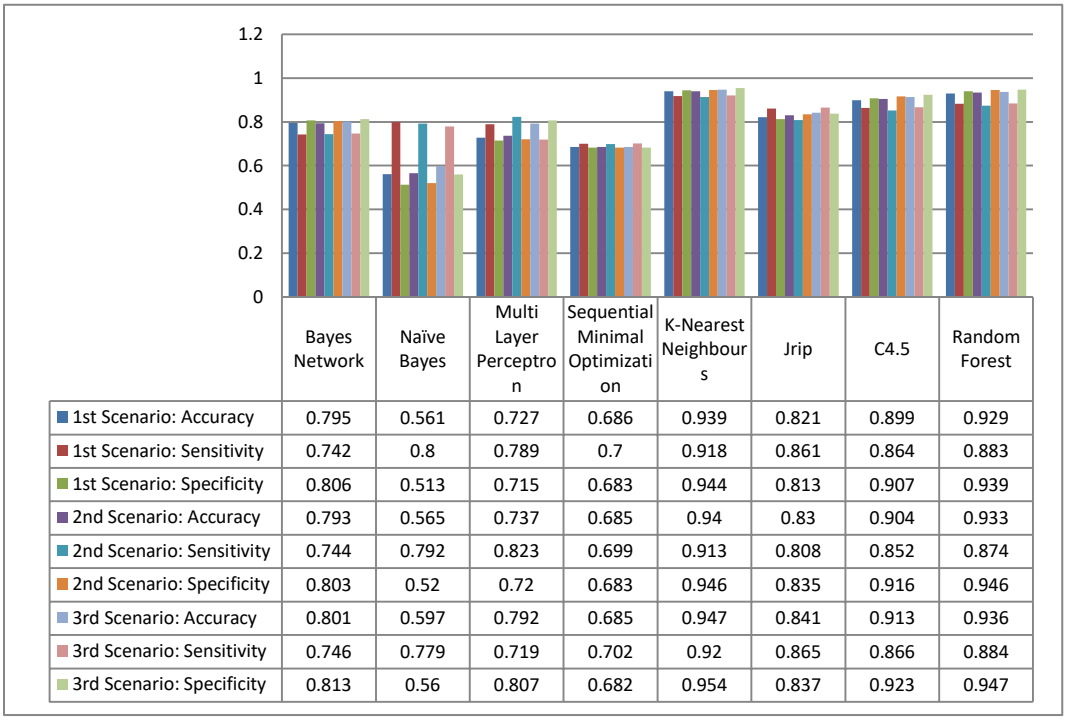
By using TOPSIS method, the best k for all scenarios is given in Table 4 for different assumptions.

**Table 4**
The best k for k-nearest neighbor method.

| Assumptions Index | First Scenario | Second Scenario | Third Scenario |
|---|---|---|---|
| 1 | K=2 | K=2 | **K=2** |
| 2 | K=2 | K=2 | **K=2** |
| 3 | K=2 | K=2 | **K=2** |
| 4 | K=3 | K=3 | **K=3** |

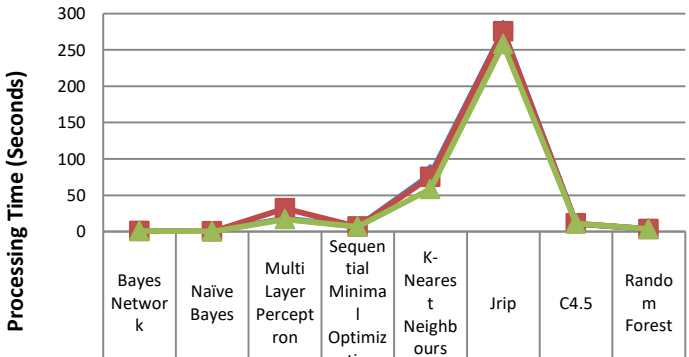
## 3.5. Experimental results on a great range of the classifiers

To compare different classification methods for all three scenarios based on the functional criteria and the assigned weights, in what follows the tuned up multi-layer perceptron and k-nearest neighbor are considered for all of the scenarios and weighting assumptions.
Figure 6 presents the accuracy, sensitivity, and specificity of classification methods. Figure 7 displays the processing times of classification methods. As it is shown in the figures, the highest accuracy, sensitivity, and specificity are obtained first for k-nearest neighbor and then for the random forest. High processing time is one of the drawbacks of k-nearest neighbor method, while processing time of random forest is strongly less than the k-nearest neighbor method.




| | Bayes Network | Naïve Bayes | Multi Layer Perceptron | Sequential Minimal Optimization | K-Nearest Neighbours | Jrip | C4.5 | Random Forest |
|---|---|---|---|---|---|---|---|---|
| 1st Scenario: Accuracy | 0.795 | 0.561 | 0.727 | 0.686 | 0.939 | 0.821 | 0.899 | 0.929 |
| 1st Scenario: Sensitivity | 0.742 | 0.8 | 0.789 | 0.7 | 0.918 | 0.861 | 0.864 | 0.883 |
| 1st Scenario: Specificity | 0.806 | 0.513 | 0.715 | 0.683 | 0.944 | 0.813 | 0.907 | 0.939 |
| 2nd Scenario: Accuracy | 0.793 | 0.565 | 0.737 | 0.685 | 0.94 | 0.83 | 0.904 | 0.933 |
| 2nd Scenario: Sensitivity | 0.744 | 0.792 | 0.823 | 0.699 | 0.913 | 0.808 | 0.852 | 0.874 |
| 2nd Scenario: Specificity | 0.803 | 0.52 | 0.72 | 0.683 | 0.946 | 0.835 | 0.916 | 0.946 |
| 3rd Scenario: Accuracy | 0.801 | 0.597 | 0.792 | 0.685 | 0.947 | 0.841 | 0.913 | 0.936 |
| 3rd Scenario: Sensitivity | 0.746 | 0.779 | 0.719 | 0.702 | 0.92 | 0.865 | 0.866 | 0.884 |
| 3rd Scenario: Specificity | 0.813 | 0.56 | 0.807 | 0.682 | 0.954 | 0.837 | 0.923 | 0.947 |

**Fig. 6.** Comparison between accuracy, sensitivity and specificity of classifiers.




| | Bayes Network | Naïve Bayes | Multi Layer Perceptron | Sequential Minimal Optimization | K-Nearest Neighbours | Jrip | C4.5 | Random Forest |
|---|---|---|---|---|---|---|---|---|
| 1st Scenario | 0.78 | 0.34 | 18.29 | 6.85 | 78.58 | 276.7 | 10.83 | 3.68 |
| 2nd Scenario | 0.8 | 0.32 | 31.95 | 6.89 | 75.22 | 275.27 | 11.23 | 3.53 |
| 3rd Scenario | 0.84 | 0.3 | 16.96 | 6.88 | 58.58 | 257.67 | 11.32 | 3.45 |

**Fig. 7.** Comparison between processing time of classifiers.

By using TOPSIS, it is possible to select the most effective classification method for each scenario. The values of the decision matrix are given in Figure 6 and Figure 7.

As it is displayed in Table 5, using the first assumption on weights, in all three scenarios, the random forest appears to be the best method for classification. The random forest in the third scenario has the highest sensitivity, specificity, and accuracy, and a reasonable processing time. Really, the random forest with 80% training data (third scenario) can be selected as the most effective classifier method to detect the warning and safe situations for rear-end collisions. Its accuracy, the detection rates for the warning situations and the safe situations are 93.6%, 88.4 % and 94.7 %, respectively.

**Table 5**
Effective classifiers for different scenarios under various assumptions.

| | **First Scenario** | **Second Scenario** | **Third Scenario** |
|---|---|---|---|
| Assumption 1 | Random Forest | Random Forest | Random Forest |
| Assumption 2 | 2-Nearest Neighbor | 2-Nearest Neighbor | 2-Nearest Neighbor |
| Assumption 3 | Random Forest | Random Forest | Random Forest |
| Assumption 4 | Random Forest | Random Forest | Random Forest |

## 3.6. Knowledge extraction from the proposed classifier

The proposed random forest includes a set of decision trees. Traversing any decision tree from the root to leaves provides some controlling rules. In the proposed random forest, since the rules are extracted from decision tree, the parameters' thresholds and indicator's thresholds are different and they will be initialized according to the other parameters. Some of the obtained rules are as the following:

- **If** "Speed<=73 km/h" & "DeltaX∈[4.7,8.6] m" & "DeltaV∈[-13.5,-3.2] m/s" & "TimeGap<=1.5 s " & "TimeToCollision<=2.8 s" , **Then** "Warning with frequency=(320.26, 24.5)"
- **If** "Speed∈[69,88] km/h " & "DeltaX∈[48,69] m" & "TimeGap∈[2.3,3.1] s" & "TimeToCollision>5.25 s" **Then** "Safe with frequency 66.92"

Based on the used data, the first rule demonstrates that when the speed of follower vehicle is less than 73 km/h, the distance of leader vehicle from follower one is between 4.7 m and 8.6 m, the relative speed of leader vehicle from follower one is between -13.5 m/s and -3.2 m/s, the time gap is less than 1.5 s and the time to collision is less than 2.8 s, the status is warning. This rule is supported by 320.26 samples (pattern's weight) in the corresponding dataset where for 24.5 of samples (pattern's weight) the conclusion is not valid (not correctly predicted). In addition, the second rule shows when the speed of follower vehicle is between 69 km/h and 88 km/h, the distance of two vehicles is between 48 m and 69 m, the time gap belongs to [2.3,3.1] seconds and the time to collision is larger than 5.25, the status is safe.

## 3.7. Experimental results on the generalization capability

As it is previously mentioned, in this experiment, the near-crashes' data of rear-end collisions were considered where the driver's reaction is braking. Now, the generalization capabilities of the proposed system over data of crashes and data of the lane-changing are presented. Figure 8 shows the results of the generalization capability of the proposed system.

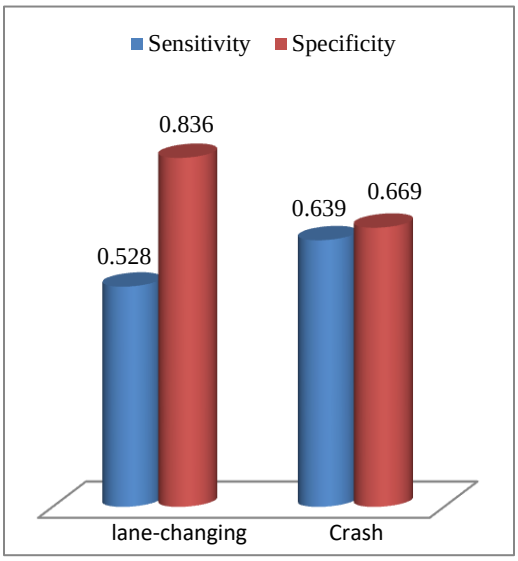



**Fig. 8.** Generalization capability of proposed system on two datasets about lane-changing and crash.

## 3.8. Comparison between the proposed system and perceptual-based systems

The most popular criteria which are used for evaluation of the safe and the warning situations are the time-to-collision and time-gap. Moreover, the time-to-collision has been used in "Honda" and "Hirst and Graham" algorithms in the category of perceptual-based systems. If the values of these criteria are equal or less than their threshold values, the situation is called a warning, otherwise, it is called safe. When the relative distance of two vehicles is less than or equal to warning distance which is calculated by "Honda" and "Hirst and Graham" algorithms, the situation is a warning, otherwise, it is detected as safe.

The success of perceptual-based warning systems depends on the appropriate and correct selection of threshold values. Due to the critical threshold of time-to-collision, there are some research studies to suggest these values, which are gathered in Table 6.

**Table 6**
Previous results about recommended time-to-collision.

| Recommended time-to-collision | Source |
|---|---|
| 2 | [27] |
| 2.2 | [20] |
| 3 | Chapter 1 of [21] |
| 3.5 | [55] |
| 4 | [56] |

In addition to the previously suggested threshold values, we will calculate the threshold value for the available vehicle trajectory data. To do this, the pruned decision tree C4.5 was used as a cost-sensitive learning model. Now, to find a threshold value for vehicle trajectory data used in this study, the time-to-collision index for data samples are determined by a pruned C4.5 decision tree. The results of this tree using cost-sensitive learning is shown in Figure 9.

Note that 0 presents the safe situation and 1 refers to the warning situation. Based on Figure 9, using cost-sensitive learning, the threshold value is 6.5. It is also worth mentioning that the threshold value of the time-to-collision criterion means a warning situation for the values between 0 to the threshold value, otherwise, it means a safe situation.

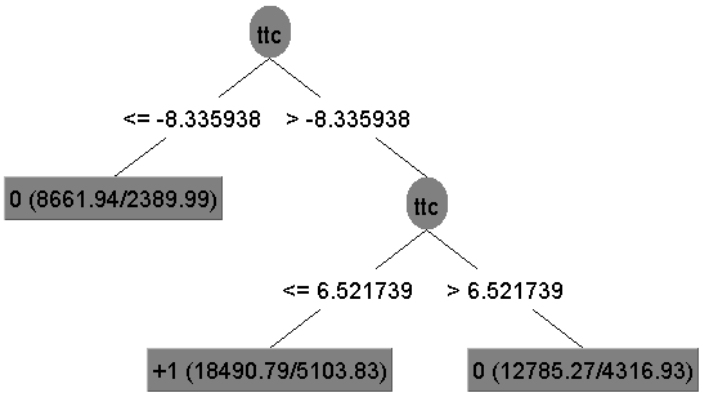
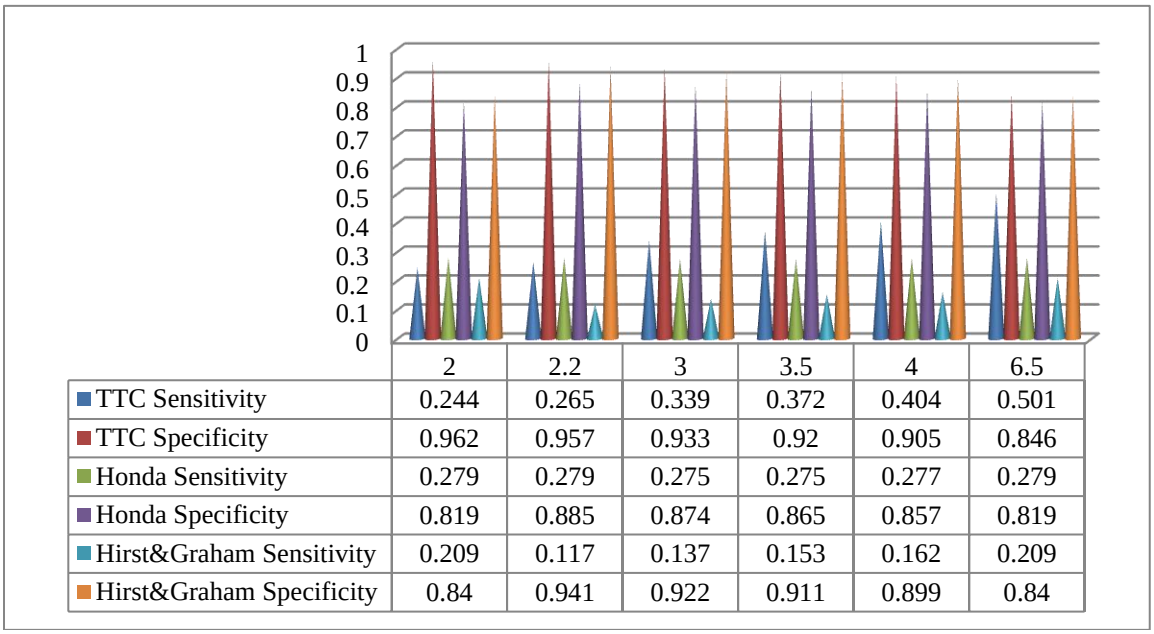



**Fig. 9.** Critical threshold for the time-to-collision using cost-sensitive learning.

Figure 10 shows the sensitivity and the specificity for the time-to- collision index with different threshold values. In addition, it shows the sensitivity and the specificity for "Honda" and "Hirst and Graham" algorithms by taking different threshold values for the time-to-collision index.




| | 2 | 2.2 | 3 | 3.5 | 4 | 6.5 |
|---|---|---|---|---|---|---|
| TTC Sensitivity | 0.244 | 0.265 | 0.339 | 0.372 | 0.404 | 0.501 |
| TTC Specificity | 0.962 | 0.957 | 0.933 | 0.92 | 0.905 | 0.846 |
| Honda Sensitivity | 0.279 | 0.279 | 0.275 | 0.275 | 0.277 | 0.279 |
| Honda Specificity | 0.819 | 0.885 | 0.874 | 0.865 | 0.857 | 0.819 |
| Hirst&Graham Sensitivity | 0.209 | 0.117 | 0.137 | 0.153 | 0.162 | 0.209 |
| Hirst&Graham Specificity | 0.84 | 0.941 | 0.922 | 0.911 | 0.899 | 0.84 |

**Fig. 10.** Sensitivity and specificity for time to collision, Honda and Hirst&Graham algorithms with different critical thresholds.

The experimental results show that the systems which work based on time-to-collision criterion are more precise for detection of the safe situations but they are weak in detection of the warning situations. So the ratio of warning situations detected by these systems is smaller than the other systems. This observation was also reported by [57].

About the critical threshold of time-gap, there are some researches that suggest the values given in Table 7.

**Table 7**
Previous results about recommended time-gap.

| Recommended time-gap | Source |
|---|---|
| 1.6 s or more (no secondary task distraction)<br>2.08 s or more (being distracted by secondary tasks) | **[58]** |
| 1.5-2.49 s (motorway)<br>1.66-3.21 s (rural wary) | **[59]** |
| 2 s or more | **[60]** |
| 1.1 s (young)<br>1.5 s (middle aged)<br>2.1 s (older) | **[61]** |
| 1.1-1.8 s | **[62]** |

To find the critical threshold value for the vehicle's trajectory data, again the time-gap of samples is determined by the pruned C4.5 decision tree. The decision tree obtained from cost-sensitive learning is shown in Figure 11. In this figure, 0 means a safe situation and 1 represents a warming situation.

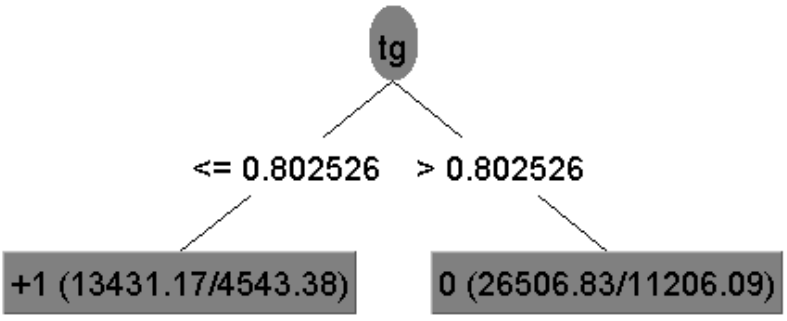
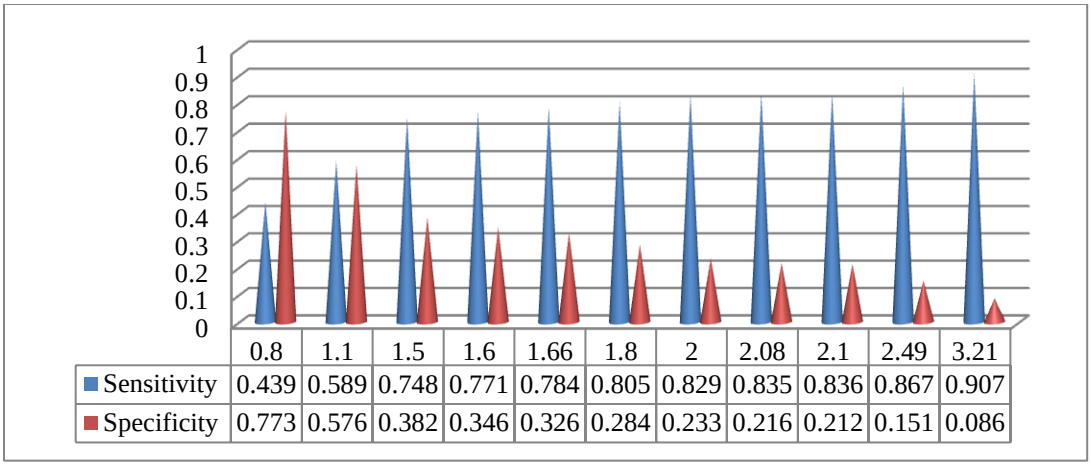



**Fig. 11.** Critical threshold for the time-gap using cost-sensitive learning

As Figure 11 shows, the critical threshold of the time-gap is 0.8. Figure 12 also shows the sensitivity and the specificity for the time-gap index with different threshold values.




| | 0.8 | 1.1 | 1.5 | 1.6 | 1.66 | 1.8 | 2 | 2.08 | 2.1 | 2.49 | 3.21 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Sensitivity | 0.439 | 0.589 | 0.748 | 0.771 | 0.784 | 0.805 | 0.829 | 0.835 | 0.836 | 0.867 | 0.907 |
| Specificity | 0.773 | 0.576 | 0.382 | 0.346 | 0.326 | 0.284 | 0.233 | 0.216 | 0.212 | 0.151 | 0.086 |

**Fig. 12.** The sensitivity and the specificity for the time-gap with different critical thresholds.

As Figure 12 shows, the systems which work based on the time-gap criterion with a threshold value greater than 1.5, are more precise for the detection of warning situations but they are weak in the detection of safe situations. As a result, the percentage of situations detected as warning by these systems is fairly more than other systems. The results are adaptable with the same results in [57].

Figure 13 shows the comparison results of the proposed classification system with the perceptual based systems, regarding the threshold values found from the vehicle's trajectory data.

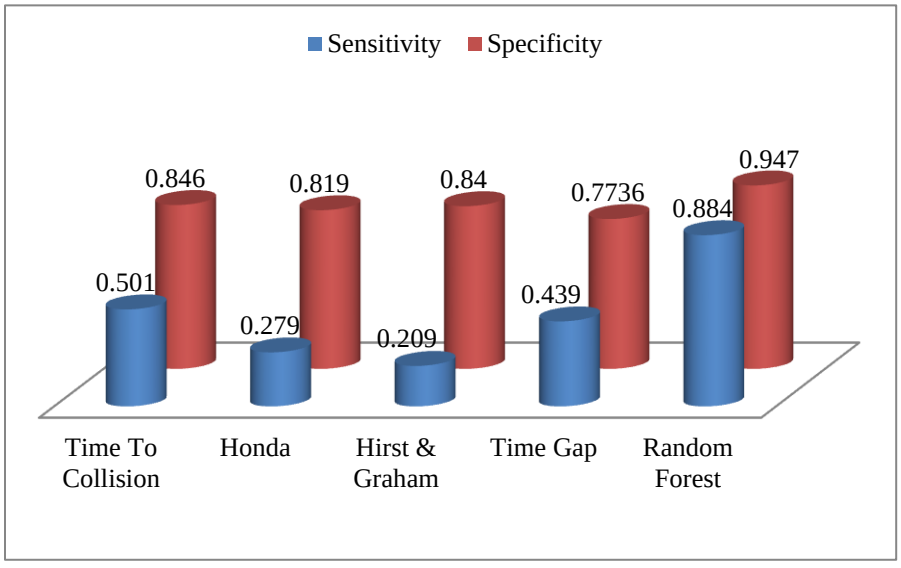



**Fig. 13.** Comparison between the proposed system and perceptual-based systems.

As Figure 13 shows, the specificity and sensitivity of the proposed random forest is different from the other perceptual-based algorithms. Really, the perceptual-based algorithms are using only one criterion and one constant threshold value for the criterion, therefore they cannot have good precision.

### 3.9. Comparison between proposed system and kinematic-based systems

In kinematic-based systems, the safe distance for the follower vehicle is used to give a warning. The most important algorithms for calculating the safe distance of the follower vehicle, are MAZDA, stop-distance, and PATH. In vehicles trajectory data that were used in this study,

1- The distances between vehicles are calculated.

2- If this distance is less than or equal to the obtained warning distance calculated by Mazda, stop-distance and PATH algorithms, the situation is considered as a warning, otherwise it is a safe situation.

By comparing the warning and the safe situations that have been obtained by these steps, with the real situations, one can obtain the sensitivity and the specificity for each algorithm based on vehicle trajectory data. Figure 14 compares the results of the proposed classification system with kinematic-based systems.

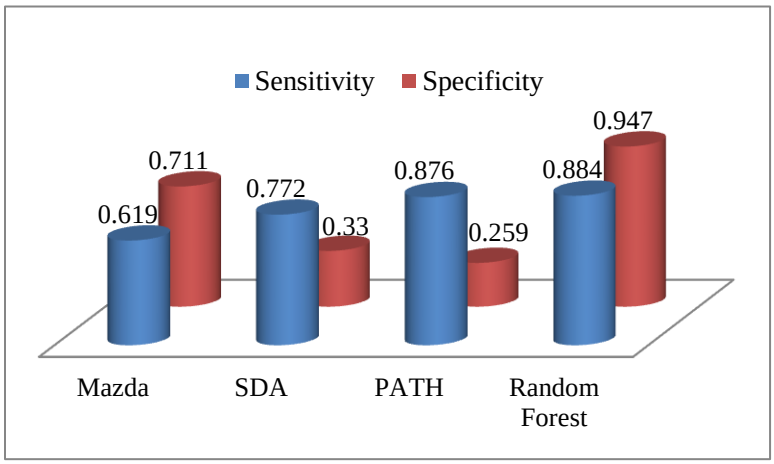



**Fig. 14.** Comparison between the proposed system and kinematic-based systems.

As Figure 14 shows, there is a huge difference between the proposed system and all of the other kinematic-based algorithms regarding the specificity and the sensitivity. Actually, kinematic-based algorithms cannot result in desirable precision because they used constant and predefined values for two parameters 1) reaction time of follower vehicle and 2) maximum decreasing rates of vehicle's speeds. They also assume a fixed decreasing rate of follower vehicle's speed. A bunch of researches has focused on finding the appropriate driver's reaction time and the maximum decreasing rate of vehicle's speed, where different values are considered for these two parameters. Therefore, the low precision of these algorithms is a result of these pre-defined and constant values. To investigate the effect of the parameters on the performance of the kinematic-based algorithms, the different parameters of Table 8 are considered for stop-distance algorithm.

The first, the second and the third series of the parameters have been considered with stop-distance, MAZDA and PATH algorithms previously. The fourth series are determined by the proposed system of the current paper. The sensitivity and the specificity results of this algorithm with the parameters of Table 8 are shown in Figure 15.

**Table 8**
Initializing the parameters of the stop-distance algorithm.

| | Maximum decreasing rate of follower vehicle $a_f(m/s^2)$ | Maximum decreasing rate of leader vehicle $a_l(m/s^2)$ | Driver's reaction time of follower vehicle $\tau_{driver}(s)$ |
|---|---|---|---|
| First Series of Parameters | 5 | 5 | 1.5 |
| Second Series of Parameters | 6 | 8 | 0.1 |
| Third Series of Parameters | 6 | 6 | 0.5 |
| Fourth Series of Parameters | 7 | 7 | 0.8 |

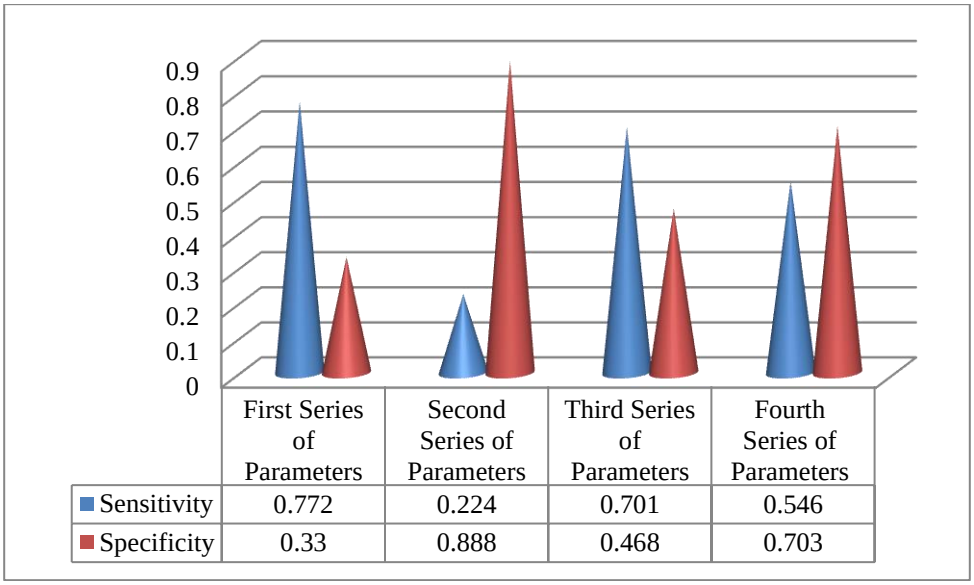



**Fig. 15.** Stop-distance algorithm with different series of parameters initialization.

As Figure 15 shows, any change in the constant parameters in the algorithm results in in a remarkable change in sensitivity and specificity. Therefore, kinematic-based systems are not robust for the rear-end collision warning.

## 3.10. Summarization of the results of the proposed warning system

Figure 16 shows a comparison between the proposed classification system with kinematic-based systems and perceptual-based systems. The threshold values for perceptual-based systems are equal to the threshold values obtained from the vehicles trajectory data used in this study. Actually, the time-to-collision index is 6.5 and the time-gap index is 0.8.

Since the kinematic-based algorithms use the constant and predefined values for the parameters and the perceptual-based algorithms use constant threshold values, they cannot result in good precision. Using only one criterion in rear-end collision warning systems improves and the others are left. Also, the time-to-collision index is weak to detect warning situations and the time-gap index bothers the driver with frequent warning alarms. However, the proposed random forest outperforms the other systems for detecting warning and safe situations. It provides high precision for the following reasons:

1) Using both of the time-to-collision and the time-gap indices associated with the speed, relative speed, relative distance,

2) Dynamic threshold values instead of the constant threshold values,

3) Recognizing the warning and safe situations by learning based on the naturalistic data of collisions, before event, through the event, and after event.

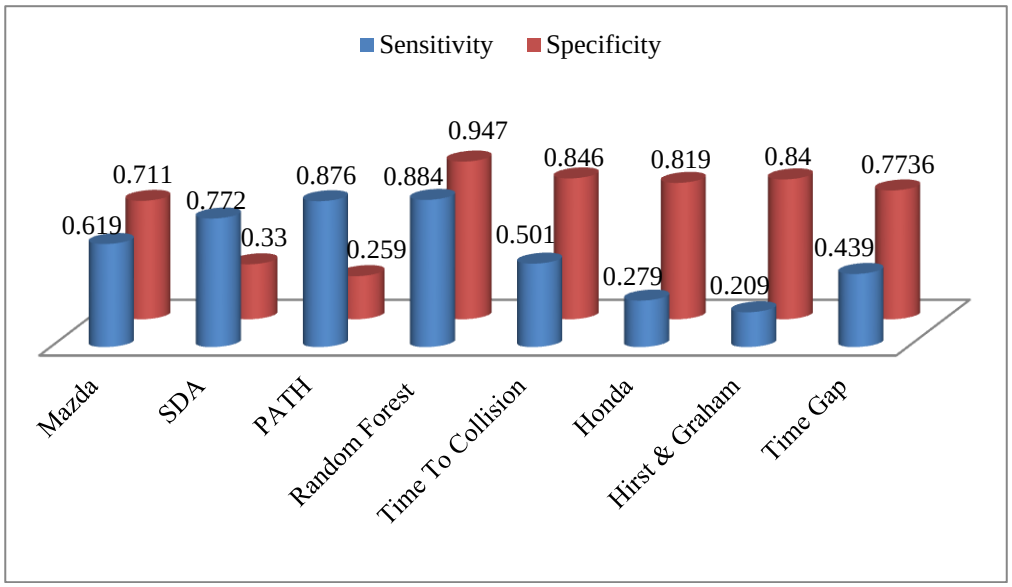



**Fig. 16.** Comparison between the proposed classification system with kinematic-based systems and perceptual-based systems.

## 4. Conclusions

In this paper, a rear-end collision warning system based on data mining was proposed. Using the classification algorithms including Bayesian network, Naïve Bayes, MLP neural network, support vector machine, k-nearest neighbor, rule-based methods, decision tree, and random forest, it was proved that the random forest got the best classification results. This classifier was powerful to recognize the warning and safe classes. Since the data of two classes were imbalanced, a combination of cost-sensitive learning and classification methods was used. Using sensitivity, specificity, and processing time as selection criteria, TOPSIS method was used to prove the preference of the random forest for the rear-end collision warning system. This classifier detected warning situations and safe situations with 88.4% and 94.7% accuracies. The

proposed rear-end collision warning classifier was compared with the perceptual-based and kinematic-based algorithms and it is shown that this system outperforms the previous algorithms.

The limitation of this study can be summarized as the following:

1. The presented random forest in some cases suffers from overfitting.
2. The effect of uncertainty is neglected in the corresponding classification system.
3. The results should be validated in real driver assistant systems.

In future works, one can consider these limitations to improve warning systems. Also, one can predict the degree of danger instead of the classification of the driving situations for implementing rear-end collision warning system. Multiple linear regression and hybrid algorithms in the recent literature[63] seem to be applicable to this problem, but more research is needed.

## Acknowledgment

The authors express their particular thanks to four anonymous reviewers whose comments led us to essentially improve in the paper.

## References

[1] Yan X, Radwan E, Abdel-Aty M. Characteristics of rear-end accidents at signalized intersections using multiple logistic regression model. Accid Anal Prev 2005;37:983–95. doi:10.1016/j.aap.2005.05.001.

[2] Liang J, Chen L, Cheng X, Chen X. Multi-agent and driving behavior based rear-end collision alarm modeling and simulating. Simul Model Pract Theory 2010;18:1092–103. doi:10.1016/j.simpat.2010.02.006.

[3] Jarašūniene A, Jakubauskas G. Improvement of road safety using passive and active intelligent vehicle safety systems. Transport 2007;22:284–9.

[4] Gietelink O, Ploeg J, De Schutter B, Verhaegen M. Development of advanced driver assistance systems with vehicle hardware-in-the-loop simulations. Veh Syst Dyn 2006;44:569–90. doi:10.1080/00423110600563338.

[5] NHTSA N. Traffic Safety Facts, 2012 Data: Pedestrians'. Ann Emerg Med 2015;65:452.

[6] Lee JD, McGehee D V., Brown TL, Reyes ML. Collision Warning Timing, Driver Distraction, and Driver Response to Imminent Rear-End Collisions in a High-Fidelity Driving Simulator. Hum Factors J Hum Factors Ergon Soc 2002;44:314–34. doi:10.1518/0018720024497844.

[7] Zhao X, Jing S, Hui F, Liu R, Khattak AJ. DSRC-based rear-end collision warning system – An error-component safety distance model and field test. Transp Res Part C Emerg Technol 2019;107:92–104. doi:10.1016/j.trc.2019.08.002.

[8] Fu Y, Li C, Luan TH, Zhang Y, Yu FR. Graded Warning for Rear-End Collision: An Artificial Intelligence-Aided Algorithm. IEEE Trans Intell Transp Syst 2020;21:565–79. doi:10.1109/TITS.2019.2897687.

[9] Chen T, Liu K, Wang Z, Deng G, Chen B. Vehicle forward collision warning algorithm based on road friction. Transp Res Part D Transp Environ 2019;66:49–57. doi:10.1016/j.trd.2018.04.017.

[10] Xiang Y, Huang S, Li M, Li J, Wang W. Rear-End Collision Avoidance-Based on Multi-Channel Detection. IEEE Trans Intell Transp Syst 2019:1–11. doi:10.1109/TITS.2019.2930731.

[11] McGehee D V, Brown TL, Wilson TB, Burns M. Examination of drivers' collision avoidance behavior in a lead vehicle stopped scenario using a front-to-rear-end collision warning system. Contract DTNH22-93-C-07326) Washington, DC Natl Highw Traffic Saf Adm 1998.

[12] Seiler P, Song B, Hedrick JK. Development of a collision avoidance system. SAE Trans 1998:1334–40.

[13] Brown TL, Lee JD, McGehee D V. Human Performance Models and Rear-End Collision Avoidance Algorithms. Hum Factors J Hum Factors Ergon Soc 2001;43:462–82. doi:10.1518/001872001775898250.

[14] Tang-Hsien Chang, Chih-Sheng Hsu, Chieh Wang, Li-Kai Yang. Onboard Measurement and Warning Module for Irregular Vehicle Behavior. IEEE Trans Intell Transp Syst 2008;9:501–13. doi:10.1109/TITS.2008.928243.

[15] Bella F, Russo R. A Collision Warning System for rear-end collision: a driving simulator study. Procedia - Soc Behav Sci 2011;20:676–86. doi:10.1016/j.sbspro.2011.08.075.

[16] Benedetto F, Calvi A, D'Amico F, Giunta G. Applying telecommunications methodology to road safety for rear-end collision avoidance. Transp Res Part C Emerg Technol 2015;50:150–9. doi:10.1016/j.trc.2014.07.008.

[17] Vogel K. A comparison of headway and time to collision as safety indicators. Accid Anal Prev 2003;35:427–33. doi:10.1016/S0001-4575(02)00022-2.

[18] Naranjo JE, Gonzalez C, Garcia R, de Pedro T. Cooperative Throttle and Brake Fuzzy Control for ACC$+$ Stop&Go Maneuvers. IEEE Trans Veh Technol 2007;56:1623–30. doi:10.1109/TVT.2007.897632.

[19] Fancher P, Bareket Z, Ervin R. Human-Centered Design of an Acc-With-Braking and Forward-Crash-Warning System. Veh Syst Dyn 2001;36:203–23. doi:10.1076/vesd.36.2.203.3557.

[20] Fujita Y, Akuzawa K, Sato M. Radar brake system. Jsae Rev 1995;1:113.

[21] Noy YI. Ergonomics and safety of intelligent driver interfaces. CRC Press; 1997.

[22] Kim S-Y, Kang J-K, Oh S-Y, Ryu Y-W, Kim K, Park S-C, et al. An Intelligent and Integrated Driver Assistance System for Increased Safety and Convenience Based on All-around Sensing. J Intell Robot Syst 2008;51:261–87. doi:10.1007/s10846-007-9187-0.

[23] Dagan E, Mano O, Stein GP, Shashua A. Forward collision warning with a single camera. IEEE Intell. Veh. Symp. 2004, IEEE; n.d., p. 37–42. doi:10.1109/IVS.2004.1336352.

[24] Kusano KD, Gabler HC. Safety Benefits of Forward Collision Warning, Brake Assist, and Autonomous Braking Systems in Rear-End Collisions. IEEE Trans Intell Transp Syst 2012;13:1546–55. doi:10.1109/TITS.2012.2191542.

[25] Oh C, Kim T. Estimation of rear-end crash potential using vehicle trajectory data. Accid Anal Prev 2010;42:1888–93. doi:10.1016/j.aap.2010.05.009.

[26] Moon S, Moon I, Yi K. Design, tuning, and evaluation of a full-range adaptive cruise control system with collision avoidance. Control Eng Pract 2009;17:442–55. doi:10.1016/j.conengprac.2008.09.006.

[27] Milanés V, Pérez J, Godoy J, Onieva E. A fuzzy aid rear-end collision warning/avoidance system. Expert Syst Appl 2012;39:9097–107. doi:10.1016/j.eswa.2012.02.054.

[28] Jamson AH, Lai FCH, Carsten OMJ. Potential benefits of an adaptive forward collision warning system. Transp Res Part C Emerg Technol 2008;16:471–84. doi:10.1016/j.trc.2007.09.003.

[29] Ararat O, Kural E, Guvenc BA. Development of a Collision Warning System for Adaptive Cruise Control Vehicles Using a Comparison Analysis of Recent Algorithms. 2006 IEEE Intell. Veh. Symp., IEEE; n.d., p. 194–9. doi:10.1109/IVS.2006.1689627.

[30] Liu J-F, Su Y-F, Ko M-K, Yu P-N. Development of a Vision-Based Driver Assistance System with Lane Departure Warning and Forward Collision Warning Functions. 2008 Digit. Image Comput. Tech. Appl., IEEE; 2008, p. 480–5. doi:10.1109/DICTA.2008.78.

[31] Akhlaq M, Sheltami TR, Helgeson B, Shakshuki EM. Designing an integrated driver assistance system using image sensors. J Intell Manuf 2012;23:2109–32. doi:10.1007/s10845-011-0618-1.

[32] Oh C, Park S, Ritchie SG. A method for identifying rear-end collision risks using inductive loop detectors. Accid Anal Prev 2006;38:295–301. doi:10.1016/j.aap.2005.09.009.

[33] González S, García S, Li S-T, Herrera F. Chain based sampling for monotonic imbalanced classification. Inf Sci (Ny) 2019;474:187–204. doi:10.1016/j.ins.2018.09.062.

[34] Jia X, Li W, Shang L. A multiphase cost-sensitive learning method based on the multiclass three-way decision-theoretic rough set model. Inf Sci (Ny) 2019;485:248–62. doi:10.1016/j.ins.2019.01.067.

[35] Min F, Liu F-L, Wen L-Y, Zhang Z-H. Tri-partition cost-sensitive active learning through kNN. Soft Comput 2019;23:1557–72. doi:10.1007/s00500-017-2879-x.

[36] Han J, Pei J, Kamber M. Data mining: concepts and techniques. Elsevier; 2011.

[37] Custer K. 100-Car Data. VTTI, 15-Oct-2019 n.d.

[38] Japkowicz N, Stephen S. The class imbalance problem: A systematic study1. Intell Data Anal 2002;6:429–49. doi:10.3233/IDA-2002-6504.

[39] Ling CX, Sheng VS. Class Imbalance Problem. 2010.

[40] Rodriguez-Galiano VF, Ghimire B, Rogan J, Chica-Olmo M, Rigol-Sanchez JP. An assessment of the effectiveness of a random forest classifier for land-cover classification. ISPRS J Photogramm Remote Sens 2012;67:93–104. doi:10.1016/j.isprsjprs.2011.11.002.

[41] Dutta RK, Rao TG, Sharma A. Application of Random Forest Regression in the Prediction of Ultimate Bearing Capacity of Strip Footing Resting on Dense Sand Overlying Loose Sand Deposit. J Soft Comput Civ Eng 2019;3:28–40.

[42] Dogru N, Subasi A. Traffic accident detection using random forest classifier. 2018 15th Learn. Technol. Conf., IEEE; 2018, p. 40–5. doi:10.1109/LT.2018.8368509.

[43] Hossain M, Muromachi Y. A Bayesian network based framework for real-time crash prediction on the basic freeway segments of urban expressways. Accid Anal Prev 2012;45:373–81. doi:10.1016/j.aap.2011.08.004.

[44] Farid DM, Zhang L, Rahman CM, Hossain MA, Strachan R. Hybrid decision tree and naïve Bayes classifiers for multi-class classification tasks. Expert Syst Appl 2014;41:1937–46. doi:10.1016/j.eswa.2013.08.089.

[45] Wefky A, Espinosa F, Prieto A, Garcia JJ, Barrios C. Comparison of neural classifiers for vehicles gear estimation. Appl Soft Comput 2011;11:3580–99. doi:10.1016/j.asoc.2011.01.030.

[46] Chandanshive V, Kambekar AR. Estimation of building construction cost using artificial neural networks. J Soft Comput Civ Eng 2019;3:91–107.

[47] Cao LJ, Keerthi SS, Ong CJ, Zhang JQ, Periyathamby U, Fu XJ, et al. Parallel sequential minimal optimization for the training of support vector machines. IEEE Trans Neural Networks 2006;17:1039–49.

[48] Aci M, İnan C, Avci M. A hybrid classification method of k nearest neighbor, Bayesian methods and genetic algorithm. Expert Syst Appl 2010;37:5061–7. doi:10.1016/j.eswa.2009.12.004.

[49] Cohen WW. Fast Effective Rule Induction. Mach. Learn. Proc. 1995, Elsevier; 1995, p. 115–23. doi:10.1016/B978-1-55860-377-6.50023-2.

[50] Quinlan JR. C4. 5: programs for machine learning. Elsevier; 2014.

[51] Ruggieri S. Efficient C4.5 [classification algorithm]. IEEE Trans Knowl Data Eng 2002;14:438–44. doi:10.1109/69.991727.

[52] Dingus TA, Klauer SG, Neale VL, Petersen A, Lee SE, Sudweeks J, et al. The 100-car naturalistic driving study, Phase II-results of the 100-car field experiment. United States. Department of Transportation. National Highway Traffic Safety …; 2006.

[53] Tzeng G-H, Huang J-J. Multiple attribute decision making: methods and applications. CRC press; 2011.

[54] Hornik K. Approximation capabilities of multilayer feedforward networks. Neural Networks 1991;4:251–7. doi:10.1016/0893-6080(91)90009-T.

[55] Hogema JH, Janssen WH. Effects of intelligent cruise control on driving behaviour: a simulator study. TNO; 1996.

[56] Horst RVD. Time-to-collision as a cue for decision-making in braking. Vis Veh 1991.

[57] Saffarzadeh M, Nadimi N, Naseralavi S, Mamdoohi AR. A general formulation for time-to-collision safety indicator. Proc Inst Civ Eng - Transp 2013;166:294–304. doi:10.1680/tran.11.00031.

[58] Lin T-W, Hwang S-L, Green PA. Effects of time-gap settings of adaptive cruise control (ACC) on driving performance and subjective acceptance in a bus driving simulator. Saf Sci 2009;47:620–5. doi:10.1016/j.ssci.2008.08.004.

[59] Trnros J, Nilsson L, Ostlund J, Kircher A. Effects of ACC on driver behaviour, workload and acceptance in relation to minimum time headway. 9th World Congr. Intell. Transp. Syst. Am. ITS Japan, ERTICO (Intelligent Transp. Syst. Serv., 2002.

[60] Zheng P, McDonald M. Manual vs. adaptive cruise control – Can driver's expectation be matched? Transp Res Part C Emerg Technol 2005;13:421–31. doi:10.1016/j.trc.2005.05.001.

[61] Fancher P. Intelligent cruise control field operational test. Final report. Volume II: appendices A-F. 1998.

[62] Reichart G, Haller R, Naab K. Driver assistance: BMW solutions for the future of individual mobility. Intell. Transp. Realiz. Futur. Abstr. Third World Congr. Intell. Transp. Syst. Am., 1996.

[63] Noori AM, Mikaeil R, Mokhtarian M, Haghshenas SS, Foroughi M. Feasibility of Intelligent Models for Prediction of Utilization Factor of TBM. Geotech Geol Eng 2020. doi:10.1007/s10706-020-01213-9.