

Received October 16, 2020, accepted November 18, 2020, date of publication November 24, 2020, date of current version December 7, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3040165

Crash Severity Prediction Using Two-Layer Ensemble Machine Learning Model for Proactive Emergency Management

UMER MANSOOR¹, NEDAL T. RATROUT¹, SEYD MASIUR RAHMAN², AND KHALED ASSI¹

¹Department of Civil & Environmental Engineering, King Fahd University of Petroleum & Minerals, Dhahran 31261, Saudi Arabia

²Center for Environment & Water, Research Institute, King Fahd University of Petroleum & Minerals, Dhahran 31261, Saudi Arabia

Corresponding author: Khaled Assi (khaledassi@kfupm.edu.sa)

This work was supported by the Deanship of Scientific Research at King Fahd University of Petroleum & Minerals, Dhahran, Saudi Arabia, under Grant SR171023.

ABSTRACT Many unfortunate victims in road traffic crashes do not receive ideal treatment because their injury severity is not understood at an early stage. Swift crash severity prediction enables trauma and emergency centers to estimate the potential damage resulting from a road traffic crash and accordingly dispatch the proper emergency units to provide appropriate emergency treatment. A two-layer ensemble machine learning model is proposed in this study to predict road traffic crash severity. The first layer integrates four base machine learning models: k-nearest neighbor, decision tree, adaptive boosting, and support vector machine; the second layer classifies the crash severity based on the feedforward neural network model. The models are developed using road traffic crash data of road intersections over 6 years (2011–2016) obtained from Great Britain's Department of Transport online database. Only the crash features that can be instantaneously and easily obtained are used as an input. To simplify the two-layer ensemble model, principal component analysis technique is used for dimensionality reduction in the second layer of the model. The performance of the two-layer ensemble model is compared with five base models: k-nearest neighbor, decision tree, adaptive boosting, support vector machine, and feedforward neural network. The prediction results reveal that the two-layer ensemble model outperforms the five base classification models based on two performance indicators: testing accuracy and F1 score. The transferability of the developed model is tested using the 3-year crash dataset for Canada obtained from the National Crash Database Online. The outcome indicates that the two-layer ensemble model shows the best performance for the Canadian dataset also. The proposed two-layer ensemble model would be beneficial in predicting crash severity with high accuracy based on limited initial crash information obtained from the crash location. Using this information, trauma centers would be able to prepare for appropriate and prompt medical treatment.

INDEX TERMS Emergency treatment, machine learning, principal component analysis, road intersections, and two-layer ensemble model.

I. INTRODUCTION

Road traffic crashes are considered a major threat around the world as they result in fatalities and injuries, which lead to economic and societal losses. Approximately 1.25 million people die annually in leading to an annual economic loss of 260 billion dollars, while non-fatal crashes affect no fewer than 20–50 million people per year, as reported by World Health Organization (WHO) [1]. Although road intersections

account for a very small proportion of road infrastructure, a high proportion of traffic crashes occur at road intersections in urban areas [2]. Many unfortunate victims in traffic crashes do not receive ideal treatment because their injury severity is not understood at an early stage. Significant attention is required to minimize the severity of collisions.

Since the severity of vehicular collisions is random, traditional parametric techniques such as logit and probit models have been widely used to predict crash severity. However, these parametric techniques have shortcomings. These techniques require a predefined mathematical form; the presence

The associate editor coordinating the review of this manuscript and approving it for publication was Jenny Mahoney.

of outliers and missing values in the dataset negatively affect the outcome of the prediction model. In contrast to parametric techniques, machine learning (ML) techniques can manage outliers and missing values in the dataset. Recently, several ML techniques have been employed to extract useful information from large traffic crash datasets for different road networks.

Crash severity models can predict the expected severity of a crash, which can help trauma centers to estimate the potential impacts and provide appropriate and prompt medical treatment. This is especially important when crashes occur in remote areas and when several crashes happen in proximity. Swift severity prediction enables trauma centers to dispatch the properly equipped emergency vehicles to the crashes and subsequently direct them to the hospitals or emergency centers capable of handling the patients efficiently and promptly. A previous study sought to predict on-scene crash severity for occupants using the conventional regression approach. However, this study considered multiple features that are not readily available from crash sites [3].

Our main objective in this research is to aid trauma and emergency centers in proactively managing emergencies based on limited initial crash information. This initial information could be conveyed to emergency centers through on-road CCTV or by an emergency phone call at the crash location. Using this information, trauma centers would be able to predict the severity of injuries, dispatch an appropriate emergency unit, and prepare accordingly for prompt medical treatment. None of the reviewed research has such characteristics and ability. Our contribution through this paper is the introduction of a two-layer ensemble ML model that can predict crash severity with high accuracy based on crash features that can be obtained quickly and easily.

II. LITERATURE REVIEW

The importance of traffic safety studies is highlighted by significant economic and societal losses, including unnecessary delays for road users, property damage, and health costs. Previous studies on traffic safety have mainly focused on two main aspects: prediction of road traffic collision severity and identification of significant factors affecting road traffic collision severity. Statistical techniques have traditionally been used for modeling crash severity. The most widely used techniques are the ordered probit (OP) model [4]–[7], the binary logit (BL) model [8]–[10], and the multinomial logit (MNL) model [11]. These statistical models have a clear mathematical relationship between independent and dependent variables, so the output is easy to interpret. These models have a few limitations; the first limitation is the assumption about data distribution, and the second limitation is an assumption about the linear relationship between predictor variables and the dependent variable. Incorrect factor estimates are produced if any of these assumptions are violated [12]–[14]. To overcome the shortcomings of statistical methods, many ML techniques have been employed, which do not assume any underlying relationship between dependent and predictor

variables [15]–[20]. In this section, studies relating to severity prediction are presented.

Many studies have compared parametric and non-parametric techniques for predicting crash severity. One study was conducted to predict and compare road traffic collision severity using machine learning and statistical techniques. The two statistical techniques employed for the severity prediction were the OP and MNL models, while the four machine learning techniques were the k-nearest neighbor (KNN), decision tree (DT), random forest (RF), and support vector machine (SVM). The study used road traffic collision data for Florida. The study concluded that the machine learning techniques, while they suffered from over-fitting issues, outperformed the classical statistical techniques in terms of prediction accuracy. Among these six techniques, RF showed the highest overall prediction accuracy while OP had the lowest accuracy [21]. Another study modeled and compared crash severity using the MNL, mixed multinomial logit (MMNL), and SVM models using rear-end crash data for California. The study found that SVM outperformed other models in terms of prediction accuracy [22]. Singh *et al.* [23] modeled the traffic crash severity using MNL and two non-parametric techniques – RF and DT – for a dataset in Haryana, India. To balance the crashes by severity level, synthetic minority oversampling and randomized class balancing techniques were used. The RF model performed better than the other two models for classifying crash severity levels.

Decision trees and RFs are widely used as classifiers. In one study, J48 DT, RF, instance-based learning with parameter k (IBK), and MNL models were employed for motorcycle crash severity prediction. Five-year motorcycle crash data with four severity levels was collected from the road traffic collision database in Ghana. The study revealed that ML techniques outperform statistical techniques in terms of accuracy and efficiency. Location type, collision type, day and week of the crash, road surface condition, and shoulder condition were some of the factors that determined the motorcycle crash severity [24]. Decision trees (J48, ID3, and CART) and naïve Bayes algorithms were employed using the WEKA software tool for predicting crash severity. The J48 decision tree algorithm outperformed the other data mining classification techniques in terms of accuracy [25]. Wang and Kim [26] compared discrete models and tree-based models for predicting crash severity using the data for Maryland State for 2017. The MNL model belonging to the discrete models' class and the RF model belonging to the tree-based models' class were employed. The RF model outperformed the MNL model in terms of accuracy. Similarly, in another study, multilayer perceptron (MLP), rule induction (PART), and simple cart models were used for predicting motorcycle crash severity in Ghana. The study revealed that the testing accuracy was highest for the simple cart model (73.85%) followed by PART (73.45%) and MLP (72.16%). The crash severity was affected by location type, crash type, crash time, settlement type [27].

Similarly, artificial neural networks (ANN) have also been employed for predicting crash severity. One study used ANN models to predict the severity of alcohol-related vehicular crashes using the data of North Carolina. The validation accuracies for three-class neural networks and binary class neural networks were found to be 65.33% and 69.65% respectively. The study revealed that the main factor contributing to fatalities and injuries in crashes was vehicles overturning, while the use of seat belts and deployment of airbags influenced crashes that did not cause injuries [28]. Arhin and Gatiba [29] developed around 25 ANN models to predict the severity of road traffic collisions at unsignalized intersections in the USA. The study used 3301 crash records of angle crashes. The architecture of the best ANN model had five, ten, and five neurons in three hidden layers respectively, giving an accuracy of 85.62%. In another study, a novel rule-based genetic algorithm (GA) was proposed and compared to ANN, SVM, and DT using three years' (2011–2013) worth of data for Iran's Tehran province. The study revealed that GA outperformed the other three classification models with an accuracy of 87% [30].

Ensemble models, like RF and adaptive boosting (AdaBoost), are also used to model crash severity. In one study, a two-layer stacking framework was employed to predict crash severity. The stacking model is an ensemble technique consisting of two layers. The first layer integrates three basic classification models – RF, AdaBoost, and gradient boosting decision tree (GBDT); the second layer classifies the injury severity using the logistic regression (LR) model. When compared with SVM, the stacking model, neural network, and RF showed better classification performance in terms of accuracy and recall [31]. Jiang *et al.* [32] modeled highly unbalanced crash severity data using the ensemble modeling technique and global sensitivity analysis. This study aimed to efficiently model each severity level as most existing methods favor the crash categories with maximum observations. Three ensemble models – RF, AdaBoost, and GBDT – were used to model unbalanced CIS data. The study concluded that AdaBoost and GBDT produce better results and more balanced prediction accuracies. Furthermore, global sensitivity analysis revealed that grade percentage, driver restraint, crash type, heavy vehicle percentage, and road characteristics significantly influence the injury severity.

With the current applications of deep learning in almost every field, researchers in traffic safety have also started employing deep learning to model crash severity. In one study, a deep-learning-based convolutional neural network (CNN) was employed to predict road traffic collision severity. The CNN was trained and tested using eight years' worth of crash data for the city of Leeds, United Kingdom. The feature matrix to the grey image algorithm (FM2GI) was used to convert the single feature crash relationship to the grey image. The proposed method was compared with several statistical techniques like the LR model and ML techniques like SVM and ANN. The study revealed that the CNN

model performed better than all the other techniques [33]. Das *et al.* [34] used deep learning to model the crash severity of at-fault motorcycle riders. A deep scooter (deep learning framework) was used to predict severity using five years' worth of data (2010–2014) for Louisiana State. It was found that the model could predict severity with a testing accuracy of 94%, while SVM and multinomial logistic regression models for the same dataset showed an accuracy of no more than 78%. Gradient boosting trees, deep learning, and naïve Bayes machine learning techniques were used to predict the severity of road traffic collisions using six years' worth of raw data from the Spanish traffic agency. The study concluded that the deep learning model outperformed the other two models in terms of accuracy and precision [35].

Deep learning models, including CNN, could be used in this study as they have shown significant performance enhancement compared to shallow models. However, these models require large datasets and require more time to train. Recently, a few studies have discussed some specific limitations of deep learning models [36]. For example, the CNN cannot differentiate the spatial arrangement of features, which may cause erroneous results. To solve the problem, additional significantly large datasets are needed. Thus, one study has proposed a new modeling approach, such as capsule neural network [37]. However, some other recent approaches, such as deep forest [38] and others, are expected to perform better than CNN with small datasets.

The preceding section of the literature review presented several techniques employed for predicting crash severity. These include many statistical techniques and ML techniques. As a general conclusion, ML techniques like SVM, DT, and RF perform better than statistical techniques like logit and probit models. Furthermore, among the ML techniques, deep learning models generate the most accurate results for severity prediction, although deep learning requires a large amount of data for better performance.

III. CLASSIFICATION TECHNIQUES

Several techniques are used for modeling a classifier. In this study, we employed five base machine learning classification models – KNN, DT, AdaBoost, feedforward neural network (FNN), and SVM – to model road traffic crash severity. Subsequently, a two-layer ensemble model was developed using these base models to enhance the performance. This section presents a brief conceptual description of these techniques.

A. K-NEAREST NEIGHBOR

The most basic classification technique is KNN, which is used as the first choice if the information about data distribution is very little or none. It is a supervised learning technique, where the input contains k nearest training examples and the output is class membership. The KNN technique works by classifying an observation based on closest k neighbor observations. The new data point is classified based on the majority of k closest observations. A popular metric used for

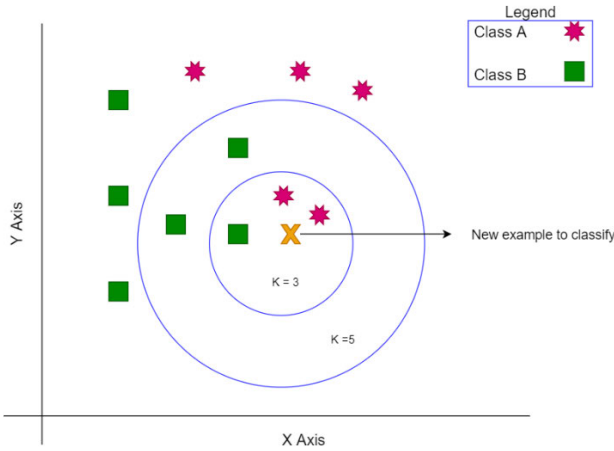


FIGURE 1. Illustration of k nearest neighbour.

the KNN algorithm is the Euclidean distance method. Two important features to be identified in KNN are the k value and the distance function. The k value is determined by testing many values of k and selecting the one with the best precision accuracy; the Euclidean distance (distance function) is determined as a distance between two-dimensional points [39]. Generally, KNN is classified based on the Euclidean distance between training samples and the test sample, as shown in the following equation:

$$d(x_i, x_l) = \sqrt{(x_{i1} - x_{l1})^2 + (x_{i2} - x_{l2})^2 + \dots + (x_{ip} - x_{lp})^2} \quad (1)$$

Here x_i is an input data, having p features $(x_{i1}, x_{i2}, \dots, x_{ip})$; n is the total number of input samples ($i = 1, 2, \dots, n$) and p is the total number of features ($j = 1, 2, \dots, p$).

KNN is clearly illustrated in Figure 1, where “red star” and “green square” represent two separate classes of training data. The value of k is a hyperparameter that can be selected through several heuristic techniques. A greater value of k generally minimizes the noise in classification data, but the distinction of boundaries between classes is not clear [40]. In the following Figure 1, when $k = 3$, the test point falls in category A, while when $k = 5$, the test point is classified the same as class B.

B. DECISION TREES

For both regression and classification problems, DT is a widely used ML technique. The goal of the DT model is to predict a target value based on many input features as illustrated in Figure 2. A DT is a flowchart-like structure used for classification of data. A DT classifies the instances by sorting them based on feature values. Each tree has nodes and branches: the node represents the feature to be classified; the branch represents the values that a node can assume. The root node is the starting point for the classification of instances in the decision tree, and instances are sorted based on their feature values. In the decision tree model, the relationship between features is clear, and feature importance is also obvious [41].

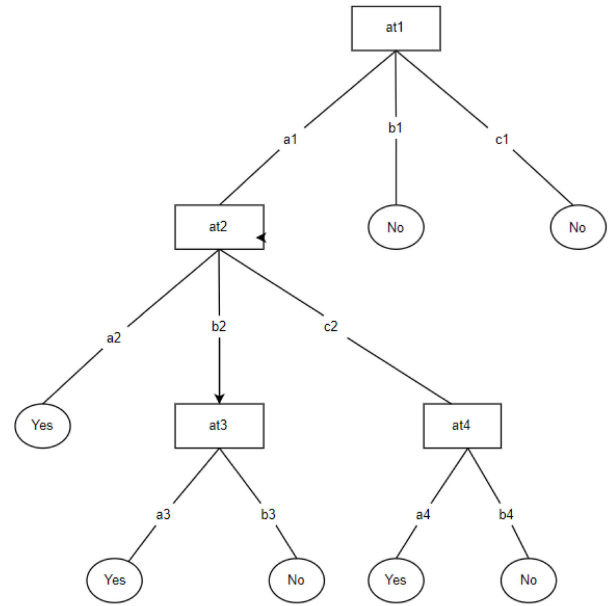


FIGURE 2. Illustration of decision tree.

In the root node or first split of a decision tree, all the features are considered, and the training dataset is divided into as many groups as the number of features. The feature that is selected first is decided based on the information gain. Information gain is the measurement amount of information a feature can give about output class. The split with maximum information gain is selected as the first split, and this process continues until the information gain approaches zero. Features with the low information gain do not separate the output classes clearly, while features with high information gain can separate two output classes and further the process of reaching the decision [42]. To measure information gain, the entropy must first be measured. The entropy is the level of impurity in the training dataset. The mathematical relationship for entropy is shown below:

$$Entropy : \sum_{i=1}^n p_i \log_2(p_i) \quad (2)$$

$$Entropy(S) = -p_+ \log_2 p_+ - p_- \log_2 p_- \quad (3)$$

where S is the sample of training examples, p_+ and p_- are the proportion of positive and negative training examples respectively.

After calculating the entropy, the information gain can be calculated using the following equations:

$$Gain(S, A) = Entropy(S) - \sum_{v \in Values(A)} \frac{rac |S_v|}{|S|} . Entropy(S_v) \quad (4)$$

Here, the set of all possible values for feature A is $Values(A)$; S_v is the subset of S , for which feature A has value v . The first term in the equation is original entropy, while the second term is the entropy of children node, which is calculated by partitioning S using feature A . Information

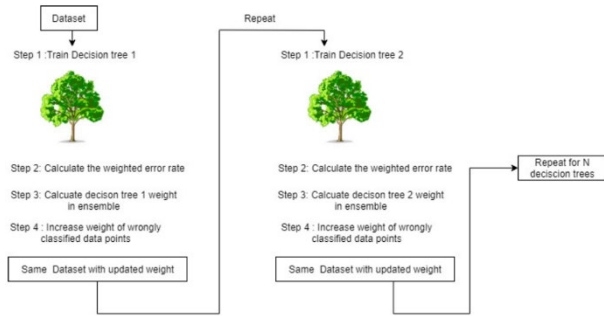


FIGURE 3. Step for the learning process of AdaBoost Model.

gain can be briefly shown by the following equation:

$$Information\ Gain = Entropy(\text{parent node}) - Average\ Entropy(\text{children node}) \quad (5)$$

The performance of the DT model can be enhanced by using a technique called pruning. Pruning is cutting off the branches that involve features with little information gain. This way, the decision tree becomes simpler and the problem of overfitting is resolved, thus enhancing the accuracy of the model [43].

C. ADABOOST

Adaptive boosting (AdaBoost) is a boosting ensemble model that was first introduced by Freund and Schapire [44]. AdaBoost performs exceptionally well with decision trees. AdaBoost learns from the misclassification error of previous data points through many iterations. The main purpose of AdaBoost is to train multiple weak learners using the same training dataset and then construct a strong learner by grouping the weak learners. AdaBoost has several features:

1. The sampling of the dataset remains the same for each iteration and only the distribution of the dataset is changed.
2. The change in the distribution of dataset depends on the accuracy of classification. In the training dataset D, every sample has a weight w associated with it. The samples that are often misclassified have higher weight, while the correctly classified samples have lower weight
3. Every weak learner in the AdaBoost algorithm has a weight represented by a vector. The input samples are represented as follows:

$$[x = (x_1, x_2, \dots, x_i)y_i], \quad i = 1, 2, 3, \dots m$$

The output is the probability of a sample belonging to different severity levels. The steps for the learning process and prediction of AdaBoost are shown in Figures 3 and 4.

D. ARTIFICIAL NEURAL NETWORKS

Artificial neural networks are among the most commonly used tools for machine learning, and they are inspired by human brains. As the name suggests, ANNs try to mimic the way a human brain reacts. Artificial neural networks

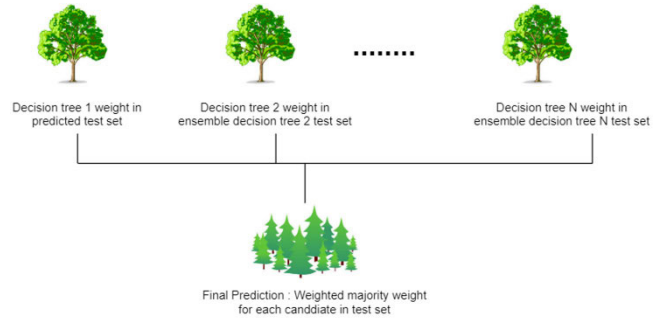


FIGURE 4. Illustration of final prediction by AdaBoost model.

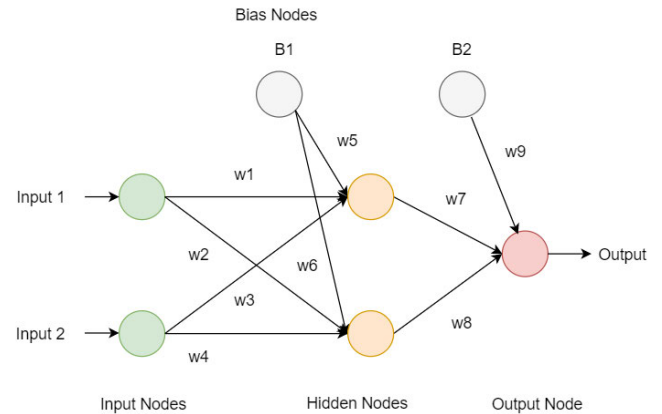


FIGURE 5. Typical feedforward neural network.

consist of input, output, and in most cases hidden layers, as shown in Figure 5. The hidden layers consist of neurons that connect input and output layers by transforming the input layer into something that can be used by the output layer. Artificial neural networks can find complex patterns in data that are far too complicated to be recognized by human programmers [45], [46].

The basic unit of an ANN is an artificial neuron that works by receiving numerical information through several input nodes. After receiving the information, the neuron processes it internally and generates an outcome. The processing phase consists of two steps: after a linear combination of input variables in the first step, the result is used as an argument for a non-linear activation function. Each connection in the ANN has a weight term (w_i) and a constant bias term Θ. The activation function is a differentiable function that can be either an identity function (y = x) or a sigmoid function. In the neural network algorithm, some hyperparameters must be fixed before the training process starts. Some of the hyperparameters that are determined before training are learning rate, number of hidden layers, and batch size. Network architecture is defined by the organization of neurons. For example, in the multilayer perceptron (MLP) type, neurons are organized in layers; the same inputs may contribute to neurons of each layer but have no connection with each other. In the feed-forward architecture of ANN, the preceding layer's output is taken as an input for the following layer [45].

TABLE 1. Descriptive statistics of key variables.

Features	Description	Percentage (%)
Crash Features		
Number of vehicles involved	One, two, three	40.3/53.4/6.3
Day of the week	Sunday = 1, Monday = 2, Tuesday = 3, Wednesday = 4, Thursday = 5, Friday = 6, Saturday = 7	11.7/14.5/14.4/14.2/15.5/16.2/13.6
Roadway Features		
Road surface condition	Dry = 1, Wet or damp = 2, Snow/ice = 3, Flood over 3cm. deep = 4,	70.6/28.0/1.4/0.1
Road Type	One-way street = 1, Dual Carriageway = 2, Single Carriageway = 3, Slip road = 4	10.5/10.6/77.9/1.0
1st Road Class	Motorway = 1, A(M) = 2, A = 3, B = 4, C = 5, Unclassified = 6	0.5/0.1/51.3/13.5/8.7/26.0
Speed Limit (mph)	20, 30, 40, 50, 60, 70	1.8/67.5/10.3/4.0/13.7/2.7
Environmental Features		
Light Condition	Daylight = 1, Darkness – lights lit = 2, Darkness – lights unlit = 3, Darkness – no street lighting = 4	69.0/29.3/0.5/5.3
Weather Condition	Fine no high winds = 1, Raining no high winds = 2, Snowing no high winds = 3, Fine with high winds = 4, Fog or Mist = 5, Others = 8	82.5/11.6/0.4/1.4/1.5/2.6
Vehicle Feature		
Vehicle Type*	Motorcycle = 1, Taxi or private car = 2, Bus or coach = 3, Goods vehicle = 4	11.4/76.1/4/8.5
Area Feature		
Area Type	Urban = 1, Rural = 2	68.8/31.2
intersection Type	Roundabout = 1, T Junction = 2, Four or more than four-arm intersection = 3	13.9/66.8/19.4
intersection Control	Authorized person = 1, Auto traffic signal = 2, Stop sign = 3, Give way or uncontrolled = 4	0.3/17.3/0.8/81.6

* Denotes vehicle type for each vehicle in the crash

E. SUPPORT VECTOR MACHINES

Support vector machines, just like ANN, are an extensively used machine learning technique for classification problems. Support vector machines transform the input data into a higher dimensional feature space. The main aim of the SVM technique is to find the best hyperplane, which maximizes the margin between support vectors. Support vectors are the points in the data that are nearest to the hyperplane and that would alter the position of the hyperplane if removed, as shown in Figure 6. The greater the margin between support vectors, the greater confidence one can have in the correct classification of the data by the hyperplane. Support vector machines solve multi-classification problems by finding a hyperplane in high dimensional space for separating points into different groups [47], [48].

In the SVM used in this study, all the crash-related input variables are represented by vectors $(x_i \in R^n)$ for $[i = 1, 2, 3 \dots N]$ and the crash severity (the training output) is represented as $(y_i \in R^n)$. The hyperplane that separates the outcome can be formulated as a set of point X that satisfies the equation

$$W \cdot X - b = 0 \tag{6}$$

where “.” represents dot product and vector W denotes normal vector that is perpendicular to the hyperplane.

In a binary classification problem, the SVM needs to be optimized for a given set of input and output variable pairs

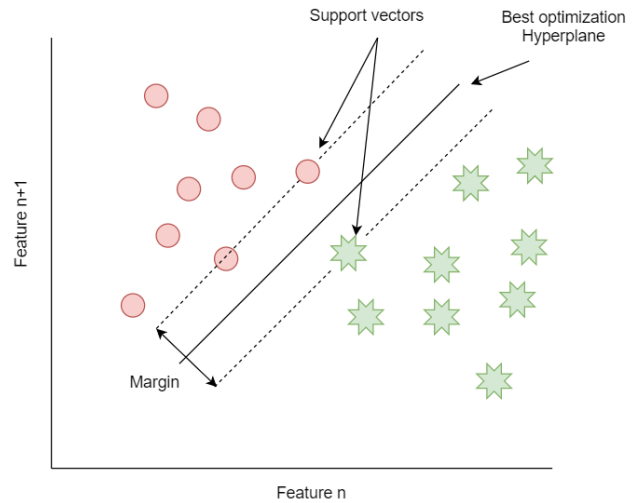


FIGURE 6. Illustration of maximum separation hyperplane.

(x_i, y_i) :

$$\min_{w,b,\xi} \frac{1}{2} w^T w + C \sum_{i=1}^N \xi_i \tag{7}$$

Subject to

$$y^i (w^T \phi(x_i) + b) \geq 1 - \xi_i, \quad \xi \geq 0 \tag{8}$$

where ξ are slack variables that measure misclassification errors and C is the penalty factor to errors that enhance the capacity control of the classifier. Lagrange multipliers

are used to minimize the objective function. Several kernels have been suggested by researchers, while radial basis function (RBF) is widely used [47], [49].

IV. DATA DESCRIPTION AND PREPARATION

Road traffic crash models are largely dependent on the availability of crash data, and thus the accuracy of these models relies on the quality of the available crash dataset. To ensure the reliability of crash models, six years' worth (2011–2016) of road traffic collision data from the Department of Transport, Great Britain,¹ were utilized in this research. The data were filtered to obtain crashes that occurred at road intersections only. Data cleaning was conducted by deleting any duplicate, irrelevant, or incomplete data fields. Extraction of the dataset after filtering and cleaning resulted in 251,000 crash records for the road intersections. Crashes at road intersections represent 60% of crashes in Great Britain between 2011 and 2016. The data contained a total of 64 input features related to road, environment, and vehicle. Out of the total features, fourteen child features were selected as input; every child feature belonged to one of five parent features as shown in Table 1: (1) crash features, (2) roadway features, (3) environmental features, (4) vehicle features, and (5) area features. The fourteen selected input features are those that can be quickly and easily obtained from the crash location within no time of a crash. A random sample of crashes containing 6000 crash points was extracted from the filtered dataset using a randomized class balancing procedure [50]. This technique eliminated any possibility of bias toward a specific severity level; the severity levels were categorized as slight, serious, and fatal in the original dataset. The severity level explains the level of injury sustained by individuals involved in a crash. Due to a significantly low percentage of fatal crashes in the dataset, fatal and serious crashes were merged and categorized as severe crashes. The severity levels used in the analysis are illustrated in Table 2.

Most of the crashes in the dataset involved two vehicles (53%), most of the vehicles were passenger cars (76%), and the highest number of crashes occurred on Fridays (16%). A substantially higher number of crashes are recorded when the road surface was dry (71%) compared to when the road surface was wet. Furthermore, 78% of the crashes occurred on single-carriageway roads. The environmental features of our data suggest that most crashes occurred during the daylight (69%) and in fine weather conditions (83%). Around 69% of the crashes were recorded in urban areas; 32% occurred in rural areas. For crashes occurring at intersections, most crashes were at T intersections (67%); 82% of the crashes were at uncontrolled intersections.

To achieve accurate results from machine learning models, all the variables utilized in model development were on the same scale. Data standardization assists most machine learning algorithms to converge by minimizing the loss function.

¹<https://data.gov.uk/dataset/cb7ae6f0-4be6-4935-9277-47e5ce24a11f/road-safety-data>

TABLE 2. Crash severity level.

Crash Severity	Level
Slight	Non-Severe
Serious	Severe
Fatal	

This data scaling is achieved by subtracting the mean value of each variable from the original score of each observation and then dividing it by the standard deviation of the variable. This scaling results in variables' transformation having a zero mean and unit variance. The standardized value of each variable denoted by Z is calculated using the following equation:

$$Z = \frac{(x - \mu)}{\sigma} \quad (9)$$

where μ is the mean of a variable, σ is the standard deviation and x is the original encoded value of each observation of a variable.

V. METHODOLOGY

The predictive models for all six ML techniques were developed using a high-level programming language, MATLAB. The fourteen input features used for model development are shown in Table 1. The five base models used in this study are KNN, DT, AdaBoost, SVM, and FNN. These five base models have been used by many researchers in the recent past to predict traffic crash severity [20], [21], [31]. To improve the performance, a sixth model – a two-layer ensemble model – was developed by stacking the base models. The input crash data were randomly divided into training and testing datasets with percentages of 70% and 30%, respectively.

The parameters were carefully set in all the machine learning models to achieve accurate predictive results. Two important features to be identified in KNN are the k value and distance function. The k value was determined through trial and error. Values of k between 1 and 100 were tried, and the predictive performance was checked against every k value. In this study, the k value was finally set at 65; Euclidean distance, which is most commonly used, was selected as a distance function. In the DT model, the parameters were chosen upon consideration of the type of data, sample size, and critical interest in the fatal crash. In DT, the initially selected feature was decided based on the information gain. The split with maximum information gain was selected as the first split, and this process continued until the information gain approached zero. To overcome the problem of overfitting, pruning was conducted by removing the splits with little information gain. In the AdaBoost model, the parameter that must be optimized is the number of weak learners or trees to train. There is a tradeoff between model accuracy and computational time during parameter tuning. After performing several experimental tests and consulting literature, the number of trees was selected to be 1500; this returned the best testing accuracy.

In the SVM model, two hyperparameters that must be optimized are penalty factor C and Gamma (γ). The penalty factor C controls the cost of misclassification on training data, while the γ parameter characterizes how far the impact of single training example extends. A systematic trial and error procedure was followed to determine the values of C and γ in SVM. The values of C and γ were finally set to be 130 and 5, respectively. Along with these two parameters, kernel function also affects the classification accuracy. In this study, several kernel functions (as suggested by many researchers [47]) were tried. Radial basis function (RBF), a widely used kernel, was finally chosen due to better results.

A feedforward neural network (FNN) was eventually developed to predict crash severity. Classification by FNN is an iterative process for adjusting weights and bias based on the provided information. FNN architecture consists of three types of layers: the input layer, hidden layers, and output layer. These layers consist of neurons that are interconnected with the subsequent layer. In this study, the input layer consisted of 14 neurons, each representing one input feature as shown in Table 1; the output layer consisted of one neuron that represented the target variable. An iterative searching procedure was followed to set the number of hidden layers and the number of neurons in each hidden layer until an optimized model was obtained. Several iterations resulted in two hidden layers consisting of five and two neurons, respectively. After testing several training algorithms, the Levenberg-Marquardt (LM) training algorithm was selected for the FNN, as this provided the best predictive accuracy. After setting the parameters for the above ML techniques, the models were tested on the testing dataset, and the performance of each model was evaluated based on the confusion matrix (CM).

After developing the base models, a stacking framework was applied to integrate five base machine learning models – KNN, DT, AdaBoost, SVM, and FNN – for crash severity prediction. Stacking is an ensemble modeling technique for combining multiple models using a meta-classifier [51]. This framework consisted of two layers, as illustrated in Figure 7. The four base models – KNN, DT, AdaBoost, and SVM – were trained and validated in the first layer. The four base models were selected based on their diversity. In the second layer, a meta-classifier, FNN, was used for classification based on the outputs of the four base models from the first layer. However, by using only the outputs of the first layer as input for the second layer, the model suffered from underfitting. To overcome this, some features from Table 1 were used as additional input, along with the outputs of layer one. The purpose of this was to simplify the model by using a reduced number of input features in layer two compared to those used for the base models. To minimize the number of features, principal component analysis (PCA) was performed using the SPSS statistical package.

Principal component analysis is a mathematical technique employed to transform a high-dimensional dataset into low-dimensional orthogonal feature space. To do this,

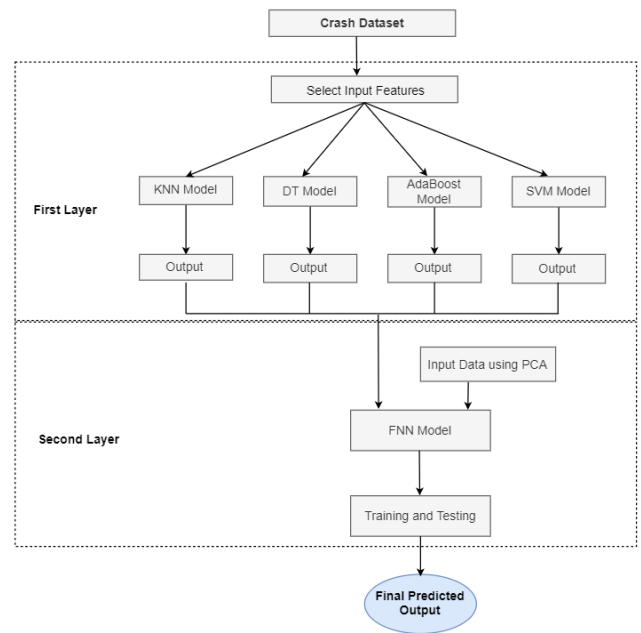


FIGURE 7. Structure for the two-layer ensemble model.

PCA transforms several highly correlated features into a smaller number of uncorrelated features; the maximum variance of the dataset is retained. The small number of uncorrelated features are called principal components. The maximum variance in the data is explained by the first principal component, followed by each following component, which explains the next maximum possible variance. Principal component analysis works on the principle of a mathematical technique called Eigen analysis, where one solves for the eigenvalue and eigenvector of a square matrix, called the covariance matrix. In our analysis, the average eigenvalue criterion, also known as an eigenvalue-one rule, was followed to select the principal components. According to this rule, only principal components with an eigenvalue greater than 1.0 were selected [52]. The six principal components had eigenvalue greater than 1.0, as shown by a scree plot in Figure 8. These six principal components explain 64.7% of the variance in the data. Extracted principal components can be interpreted according to component loadings, which represent the correlation between original features and the principal components. For ease of interpreting the six principal components, varimax rotation was conducted. The component loading obtained from varimax rotation is shown in Table 3. Only the loadings with an absolute value of more than 0.3 were tabulated; this facilitates the interpretation of the principal components [53]. For more information about PCA, see Rencher [54]. The principal component analysis of the entire dataset of 14 features resulted in six principal components (Table 3) based on the eigenvalue-one rule. These six principal components, along with the output of layer one, were used as input for layer two of the ensemble model.

The input features in the second layer were trained using FNN. The input layer of FNN consisted of 10 neurons, each representing one input features, while the output layer

TABLE 3. Rotated component loadings for PCA of crash data.

Variables	Components					
	1	2	3	4	5	6
Number of Vehicles	.919					
Vehicle2 Type	.739					
Vehicle3 Type	.686					
Speed limit		.882				
Urban or Rural Area		.879				
Road Type			.831			
intersection Detail			.739		-.466	
1st Road Class		-.388	.408		.523	
Road Surface Conditions				.794		
Weather Conditions				.709		
Light Conditions				.531		
intersection Control					.861	
Day of Week						.732
Vehicle1 Type						-.643

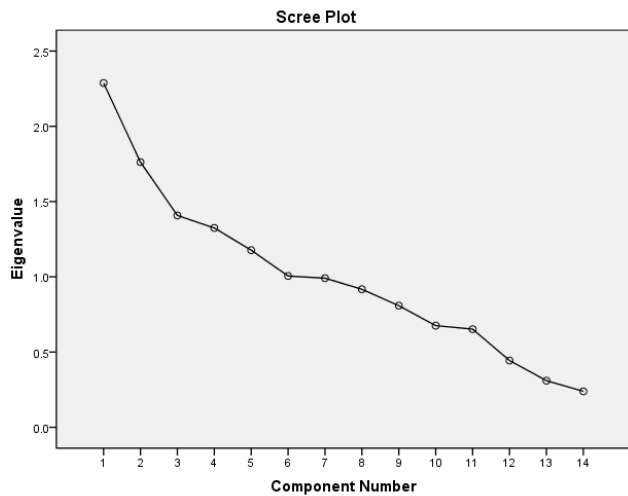


FIGURE 8. Scree plot for principal components.

consisted of one neuron that represented the target variable. An iterative searching procedure resulted in two hidden layers consisting of ten and two neurons respectively. In FNN, the LM training algorithm was selected after testing several training algorithms – it provided the best predictive accuracy. After setting these parameters, the two-layer ensemble model was tested on the test set.

VI. RESULTS AND DISCUSSION

The results of the proposed two-layer ensemble model were compared with five base machine learning models. The models were trained and tested on a dataset that was randomly divided into a training set and a testing set with a ratio of 7:3. The performance of each model was evaluated based on the data generated by the CM, which contains the results of the original and predicted classifications provided by a classification model. A general representation of a CM for binary output classes is shown in Table 4 – observations of

TABLE 4. Confusion matrix.

Total Instances		Predicted	
		Negative	Positive
Actual	Negative	True Negative (TN)	False Positive (FP)
	Positive	False Negative (FN)	True Positive (TP)

an actual class are shown in the rows; observations of the predicted class are represented in the columns.

The entities in CM are defined as follows:

- TN represents the entities that are originally negative and classified correctly as negative.
- FN represents entities that are positive but incorrectly classified as negative.
- TP represents the entities that are originally positive and classified correctly as positive.
- FP represents entities that are originally negative but incorrectly classified as positive.

The observations of the confusion matrix for every model were used to calculate the following performance metrics and evaluate model performance based on these metrics:

- Accuracy: the proportion of the total number of instances that were classified correctly, shown by the following equation:

$$Accuracy(AC) = \frac{(TP + TN)}{(TP + TN + FP + FN)} \quad (10)$$

- Error rate: the rate of misclassification of predictions.

$$Error_rate(ER) = 1 - Accuracy \quad (11)$$

- Recall: the proportion of positive instances that were classified correctly, shown by the following equation:

$$Recall(R) = \frac{TP}{TP + FN} \quad (12)$$

TABLE 5. Classification of different models.

Model	Training Accuracy (%)	Testing Accuracy (%)
K Nearest Neighborhood (KNN)	67.9	67.1
Decision Trees (DT)	79.4	69.2
AdaBoost	75.6	71.4
Feedforward Neural Network (FNN)	71.7	69.7
Support Vector Machine (SVM)	71.4	68.8
Two-layer Ensemble Model	81.6	76.7

- **Precision:** the proportion of the anticipated positive cases that were correct, shown by the following equation:

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

- **F1-Measure:** the performance of the model is measured using the F1 measure that represents the harmonic mean of Recall and Precision. Its value ranges between 0 and 1, where 1 represents the best model while 0 represents the worst model. The equation for the F1 score is shown below.

$$F1 = \frac{2 * (R * P)}{R + P} \quad (14)$$

The classification accuracy of each model is shown in Table 5 and Figure 9. The overall training accuracy ranges from 67.9% to 81.6%, while the overall testing accuracy ranges from 67.1% to 76.7%. The FNN and SVM models show almost similar performance; DT suffers from overfitting. Among the base models, AdaBoost has the highest testing accuracy while KNN performs the worst among all models. The proposed two-layer ensemble model shows excellent performance in training and testing, outperforming all the other models with a training accuracy of 81.6% and testing accuracy of 76.7%.

Although accuracy is a metrics that represents the performance of an individual model, relying only on accuracy as a performance measure can be misleading – the model might be biased toward one severity class. To overcome these limitations, other performance measures like precision, recall, and F1 score were determined. These performance measures determine the performance of individual severity levels, providing better insight into the models. The results of these performance measures for both severity levels are illustrated in Tables 6 and 7 respectively. Figures 10 and 11 also depict the performance of models for both severity levels.

According to the definitions of precision and recall, any model that maximizes both performance measures is the best. The F1 score acts as a good performance indicator since it uses both precision and recall to interpret the model’s performance. In this study, all the models performed almost equally well for both levels of severity, as evident from the results of performance measures in Tables 6 and 7. Among

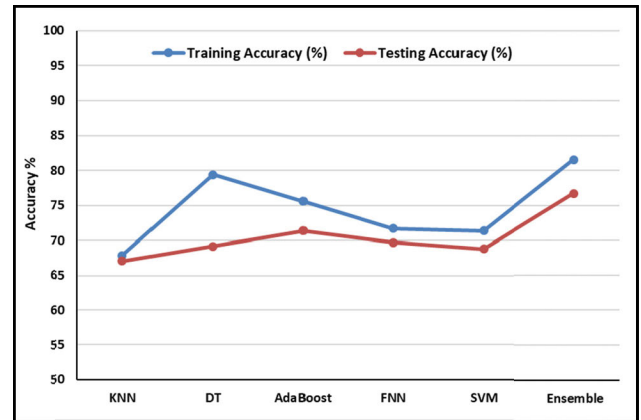


FIGURE 9. Training and testing accuracy for all models.

TABLE 6. Performance measures of models for severe crashes.

Model	Precision	Recall	F1 Score
K Nearest Neighborhood	0.6290	0.7031	0.6640
Decision Trees	0.6817	0.7076	0.6944
AdaBoost	0.7280	0.7210	0.7245
Feedforward Neural Network	0.7032	0.7063	0.7047
Support Vector Machine	0.7280	0.6979	0.7126
Two-Layer Ensemble	0.7331	0.7712	0.7517

TABLE 7. Performance measures of models for non-severe crashes.

Model	Precision	Recall	F1 Score
K Nearest Neighborhood	0.7161	0.6436	0.6779
Decision Trees	0.6989	0.6726	0.6855
AdaBoost	0.6989	0.7062	0.7025
Feedforward Neural Network	0.6874	0.6842	0.6858
Support Vector Machine	0.6632	0.6952	0.6788
Two-Layer Ensemble	0.7979	0.7628	0.7799

the base models, the KNN model has the lowest F1 score, while AdaBoost has the highest score for both the severity levels. Furthermore, DT, SVM, and FNN performed similarly for both levels of severity. Therefore, among individual models, AdaBoost outperformed the other models in terms of accuracy and F1 score, while KNN performed the worst of all the models. However, there is a significant enhancement of predictive performance with the introduction of the two-layer ensemble model. The test accuracy of the two-layer ensemble model increased to 76.7%, while the second-best model, AdaBoost, had a test accuracy of 71.4%. Similarly, the F1 score improved significantly for both levels of severity. The enhanced performance of the two-layer ensemble model indicates that it is a viable option for predicting traffic crash severity. Notably, the use of all 14 variables in the second layer of the ensemble model gives similar results to using six

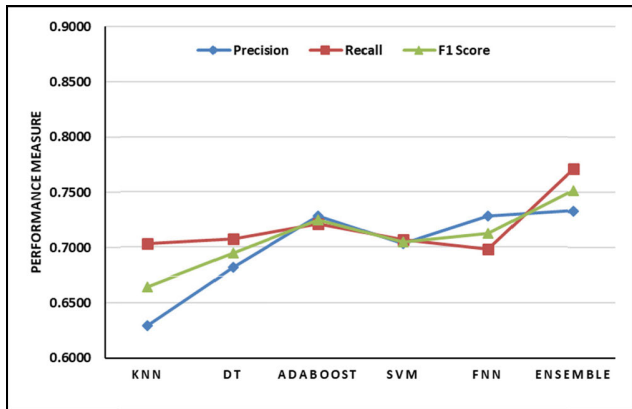


FIGURE 10. Performance measures for test dataset of severe crashes.

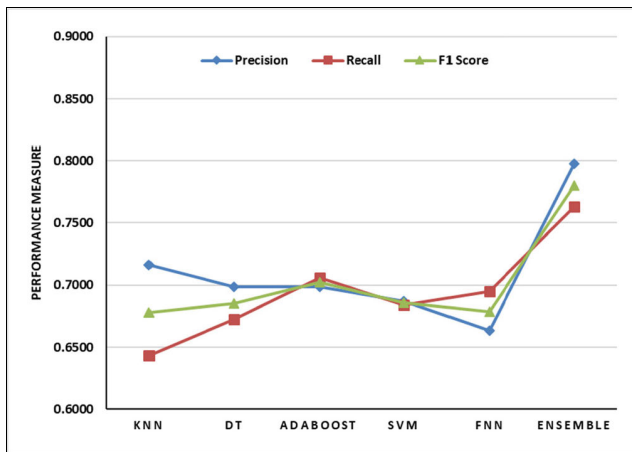


FIGURE 11. Performance measures for test dataset of non-severe crashes.

principal components obtained through PCA. Thus, to simplify the model, we suggest using only the six principal components. The two-layer ensemble model also showed reasonable performance with an accuracy reduction of only 1% if the KNN model was eliminated from the first layer.

Although the prediction accuracy of the developed ensemble model is not as high as other models in the literature [30], [34], the model is far more useful practically. The ensemble model's objective is not to predict the severity of a crash precisely, but rather to predict it using a minimum number of attributes that can be obtained quickly and easily from the crash site just when the crash happens, with reasonable accuracy. We aimed to develop a model beneficial for saving lives by predicting crash severity with acceptable accuracy based on the crash features that can be easily and rapidly obtained from the crash location. Based on this prediction, trauma centers would be able to predict the severity of injuries at any crash, dispatch an appropriate emergency unit to the crash site, and prepare prompt medical treatment for patients upon arrival at the nearest hospital.

VII. MODEL TRANSFERABILITY

This section summarizes the detailed analysis for model transferability. To check the transferability of the developed models, the models were applied on a crash dataset for three

years' worth of data (2014–2016) obtained from the online National Collision Database (NCDB) Canada.² The same procedure as that for Great Britain's dataset was followed for the model development, using similar input features, and the output of each crash was either fatal or non-fatal as indicated in the original dataset. A similar process of principal component analysis was followed, resulting in six principal components. The models were trained and tested on the crash dataset for Canada again.

The models performed similarly to Great Britain's dataset in terms of accuracy and F1 score. Among the base models, AdaBoost performed better than other base models with an accuracy of 73.6%, while KNN showed the lowest accuracy of 67.3%. The other three base models – ANN, SVM, and DT – showed similar performance, and DT again suffered from overfitting. The two-layer ensemble model also outperformed all other models with an accuracy of 79.3% and F1 scores of 0.78 and 0.80 for fatal and non-fatal crashes, respectively. The comparison of results for the two datasets indicates that these models are expected to show similar performance on any crash dataset with similar input features. These results indicate that a high accuracy of crash severity prediction is expected if these models are employed at a global level for any other dataset.

VIII. CONCLUSION AND RECOMMENDATIONS

This paper concentrates on accurately predicting crash severity using readily available features that can be easily and rapidly collected from the crash location, such as type of intersection control, weather condition, type of vehicles involved in the crash, and the speed limit in the area.

The study compared the performance of various machine learning models for predicting road traffic collision severity. Based on six years' worth (2011–2016) of road traffic collision data from the Department of Transport, Great Britain, five base models (KNN, DT, AdaBoost, SVM, FNN) and a two-layer ensemble model was developed. The two-layer ensemble model was developed by integrating KNN, DT, AdaBoost, SVM, and FNN in two layers. The models were compared with each other in terms of the testing accuracy. The dataset was randomly classified into training and testing datasets with a ratio of 7:3. Since accuracy is not always the recommended performance measure for model interpretation, three other performance indicators (precision, recall, and F1 score) were calculated to provide better insight into the performance of models.

Among the base models, AdaBoost outperformed all other individual models in terms of accuracy and F1 score, without facing the problem of overfitting. On the other hand, KNN was the least accurate model, with the lowest F1 score for both the severity levels. However, the proposed two-layer ensemble model resulted in significant improvement in the prediction of crash severity levels. The accuracy of the two-layer ensemble model for both training and testing was

²<https://open.canada.ca/data/en/dataset/1eb9eba7-71d1-4b30-9fb1-30cbdab7e63a>

substantially enhanced. The model performed better than all the base models, with a testing accuracy of 76.7% and F1 scores of 0.75 and 0.77 for severe and non-severe crashes respectively. If the KNN model was eliminated from the first layer, the two-layer ensemble model still showed reasonable performance, with just a 1% reduction in accuracy. The transferability of these models was checked on a crash dataset obtained from the online National Collision Database (NCDB) Canada. The models performed similarly to how they performed with Great Britain's dataset. The two-layer ensemble model outperformed all other models, with an accuracy of 79.3% and F1 scores of 0.78 and 0.80 for fatal and non-fatal crashes, respectively. This research indicates that a high accuracy of crash severity prediction is expected if these models are employed at a global level for any other crash dataset.

The introduction of the two-layer ensemble model to significantly improve the predictive performance for crash severity is a contribution of this research that may save human lives. This research will enable trauma and emergency centers to estimate the potential damage resulting from a traffic crash and accordingly dispatch the proper emergency units to provide appropriate emergency treatment. Although the proposed model has the highest accuracy, the limitation of the proposed method can be the higher running time of the two-layer ensemble model compared to individual models. Moreover, a randomized class balancing procedure was followed in this study to solve the problem of an imbalanced dataset. Other advanced approaches could have been used address the issue of the imbalanced dataset.

This study provides a few recommendations for future research. Firstly, sensitivity analysis can be conducted to provide complete inferences of feature importance. Although the selected features for this study are based on an extensive literature review, sensitivity analysis would provide a complete understanding of factors contributing to crash severity. Secondly, deep learning and other ensemble modeling techniques could be employed to compare the predictive performance for crash severity. Finally, the developed models should be applied on datasets for different countries of the world. Based on the results of this study, we strongly recommend that a standard crash data collection format should be used by traffic wardens across the globe. If a standard data collection format is followed, the transferability and validation of these models would be reasonable and easy.

ACKNOWLEDGMENT

The authors would like to acknowledge the support provided by the Deanship of Scientific Research at King Fahd University of Petroleum and Minerals (KFUPM), Saudi Arabia, for funding this work under the project number SR171023.

REFERENCES

[1] *Global Status Report on Road Safety 2015*, World Health Org., Geneva, Switzerland, 2015.

- [2] T. Neuman, "National cooperative highway research program report 500: Guidance for implementation of the AASHTO strategic highway safety plan, volume 5: A guide for addressing unsignalized intersection collisions," *Transp. Res. Board*, Washington, DC, USA, Tech. Rep. 500, 2003. [Online]. Available: <http://safety.transportation.org/guides.asp>
- [3] R. Buendia, S. Candefjord, H. Fagerlind, A. Bálint, and B. A. Sjöqvist, "On scene injury severity prediction (OSISP) algorithm for car occupants," *Accident Anal. Prevention*, vol. 81, pp. 211–217, Aug. 2015.
- [4] C. Lee and M. Abdel-Aty, "Comprehensive analysis of vehicle-pedestrian crashes at intersections in Florida," *Accident Anal. Prevention*, vol. 37, no. 4, pp. 775–786, 2005.
- [5] N. A. Siddiqui, X. Chu, and M. Guttenplan, "Crossing locations, light conditions, and pedestrian injury severity," *Transp. Res. Rec. J. Transp. Res. Board*, vol. 1982, no. 1, pp. 141–149, Jan. 2006.
- [6] K. K. W. Yau, H. P. Lo, and S. H. H. Fung, "Multiple-vehicle traffic accidents in Hong Kong," *Accident Anal. Prevention*, vol. 38, no. 6, pp. 1157–1161, Nov. 2006.
- [7] Y. Xie, Y. Zhang, and F. Liang, "Crash injury severity analysis using Bayesian ordered probit models," *J. Transp. Eng.*, vol. 135, no. 1, pp. 18–25, Jan. 2009.
- [8] V. Shankar and F. Mannering, "An exploratory multinomial logit analysis of single-vehicle motorcycle accident severity," *J. Saf. Res.*, vol. 27, no. 3, pp. 183–194, Sep. 1996.
- [9] A. Khorshadi, D. Niemeier, V. Shankar, and F. Mannering, "Differences in rural and urban driver-injury severities in accidents involving large-trucks: An exploratory analysis," *Accident Anal. Prevention*, vol. 37, no. 5, pp. 910–921, Sep. 2005.
- [10] P. Savolainen and F. Mannering, "Probabilistic models of motorcyclists' injury severities in single-and multi-vehicle crashes," *Accident Anal. Prevention*, vol. 39, no. 5, pp. 955–963, 2007.
- [11] A. S. Al-Ghamdi, "Using logistic regression to estimate the influence of accident factors on accident severity," *Accident Anal. Prevention*, vol. 34, no. 6, pp. 729–741, Nov. 2002.
- [12] L. Mussone, A. Ferrari, and M. Oneta, "An analysis of urban collisions using an artificial intelligence model," *Accident Anal. Prevention*, vol. 31, no. 6, pp. 705–718, Nov. 1999.
- [13] D. Delen, R. Sharda, and M. Bessonov, "Identifying significant predictors of injury severity in traffic accidents using a series of artificial neural networks," *Accident Anal. Prevention*, vol. 38, no. 3, pp. 434–444, May 2006.
- [14] Z. Li, P. Liu, W. Wang, and C. Xu, "Using support vector machine models for crash injury severity analysis," *Accident Anal. Prevention*, vol. 45, pp. 478–486, Mar. 2012.
- [15] M. A. Abdel-Aty and H. T. Abdelwahab, "Predicting injury severity levels in traffic crashes: A modeling comparison," *J. Transp. Eng.*, vol. 130, no. 2, pp. 204–210, Mar. 2004.
- [16] L.-Y. Chang and H.-W. Wang, "Analysis of traffic injury severity: An application of non-parametric classification tree techniques," *Accident Anal. Prevention*, vol. 38, no. 5, pp. 1019–1027, Sep. 2006.
- [17] Y. Xie, D. Lord, and Y. Zhang, "Predicting motor vehicle collisions using Bayesian neural network models: An empirical analysis," *Accident Anal. Prevention*, vol. 39, no. 5, pp. 922–933, Sep. 2007.
- [18] J. de Oña, R. O. Mujalli, and F. J. Calvo, "Analysis of traffic accident injury severity on spanish rural highways using Bayesian networks," *Accident Anal. Prevention*, vol. 43, no. 1, pp. 402–411, Jan. 2011.
- [19] J. Abellán, G. López, and J. de Oña, "Analysis of traffic accident severity using decision rules via decision trees," *Expert Syst. Appl.*, vol. 40, no. 15, pp. 6047–6054, Nov. 2013.
- [20] A. Iranitalab and A. Khattak, "Comparison of four statistical and machine learning methods for crash severity prediction," *Accident Anal. Prevention*, vol. 108, pp. 27–36, Nov. 2017.
- [21] J. Zhang, Z. Li, Z. Pu, and C. Xu, "Comparing prediction performance for crash injury severity among various machine learning and statistical methods," *IEEE Access*, vol. 6, pp. 60079–60087, 2018.
- [22] A. Ahmadi, A. Jahangiri, V. Berardi, and S. G. Machiani, "Crash severity analysis of rear-end crashes in California using statistical and machine learning classification methods," *J. Transp. Saf. Secur.*, vol. 12, no. 4, pp. 522–546, 2020.
- [23] G. Singh, S. Sachdeva, and M. Pal, "Comparison of three parametric and machine learning approaches for modeling accident severity on non-urban sections of Indian highways," *Adv. Transp. Stud.*, vol. 45, 2018.
- [24] L. Wahab and H. Jiang, "A comparative study on machine learning based algorithms for prediction of motorcycle crash severity," *PLoS ONE*, vol. 14, no. 4, Apr. 2019, Art. no. e0214966.
- [25] T. K. Bahiru, D. Kumar Singh, and E. A. Tessfaw, "Comparative study on data mining classification algorithms for predicting road traffic accident severity," in *Proc. 2nd Int. Conf. Inventive Commun. Comput. Technol. (ICICCT)*, Apr. 2018, pp. 1655–1660.

- [26] X. Wang and S. H. Kim, "Prediction and factor identification for crash severity: Comparison of discrete choice and tree-based models," *Transp. Res. Rec.*, vol. 2673, no. 9, 2019, Art. no. 0361198119844456.
- [27] L. Wahab and H. Jiang, "Severity prediction of motorcycle crashes with machine learning methods," *Int. J. Crashworthiness*, vol. 25, no. 5, pp. 485–492, 2020.
- [28] S. Banerjee and N. K. Khadem, "Factors influencing injury severity in alcohol-related crashes: A neural network approach using HSIS crash data," *ITE J.*, vol. 89, no. 3, pp. 42–49, Mar. 2019.
- [29] S. A. Arhin and A. Gatiba, "Predicting injury severity of angle crashes involving two vehicles at unsignalized intersections using artificial neural networks," *Eng., Technol. Appl. Sci. Res.*, vol. 9, no. 2, pp. 3871–3880, Apr. 2019.
- [30] S. H.-A. Hasheminejad, M. Zahedi, and S. M. H. Hasheminejad, "A hybrid clustering and classification approach for predicting crash injury severity on rural roads," *Int. J. Injury Control Saf. Promotion*, vol. 25, no. 1, pp. 85–101, Jan. 2018.
- [31] J. Tang, J. Liang, C. Han, Z. Li, and H. Huang, "Crash injury severity analysis using a two-layer stacking framework," *Accident Anal. Prevention*, vol. 122, pp. 226–238, Jan. 2019.
- [32] L. Jiang, Y. Xie, and T. Ren, "Modelling highly unbalanced crash injury severity data by ensemble methods and global sensitivity analysis," in *Proc. Transp. Res. Board 98th Annu. Meeting*, Washington, DC, USA, 2019, pp. 13–17.
- [33] M. Zheng, T. Li, R. Zhu, J. Chen, Z. Ma, M. Tang, Z. Cui, and Z. Wang, "Traffic accident's severity prediction: A deep-learning approach-based CNN network," *IEEE Access*, vol. 7, pp. 39897–39910, 2019.
- [34] S. Das, A. Dutta, K. Dixon, L. Minjares-Kyle, and G. Gillette, "Using deep learning in severity analysis of at-fault motorcycle rider crashes," *Transp. Res. Rec. J. Transp. Res. Board*, vol. 2672, no. 34, pp. 122–134, Dec. 2018.
- [35] L. G. Cuenca, E. Puertas, N. Aliane, and J. F. Andres, "Traffic accidents classification and injury severity prediction," in *Proc. 3rd IEEE Int. Conf. Intell. Transp. Eng. (ICITE)*, Sep. 2018, pp. 52–57.
- [36] A. Kamilaris and F. X. Prenafeta-Boldú, "Deep learning in agriculture: A survey," *Comput. Electron. Agricult.*, vol. 147, pp. 70–90, 2018.
- [37] Z. Wang, L. Zheng, W. Du, W. Cai, J. Zhou, J. Wang, X. Han, and G. He, "A novel method for intelligent fault diagnosis of bearing based on capsule neural network," *Complexity*, vol. 2019, Jun. 2019, Art. no. 6943234.
- [38] L. Sun, Z. Mo, F. Yan, L. Xia, F. Shan, Z. Ding, B. Song, W. Gao, W. Shao, F. Shi, H. Yuan, H. Jiang, D. Wu, Y. Wei, Y. Gao, H. Sui, D. Zhang, and D. Shen, "Adaptive feature selection guided deep forest for COVID-19 classification with chest CT," *IEEE J. Biomed. Health Informat.*, vol. 24, no. 10, pp. 2798–2805, Oct. 2020.
- [39] C. Elkan, "Nearest neighbor classification," Dept. Comput. Sci. Eng., Univ. California, San Diego, San Diego, CA, USA, Tech. Rep., 2011. [Online]. Available: <http://cseweb.ucsd.edu/~elkan/250B/nearestn.pdf-elkan/250B/nearestn.pdf>
- [40] B. S. Everitt, S. Landau, M. Leese, and D. Stahl, *Miscellaneous Clustering Methods Cluster Analysis*, 5th ed. Chichester, U.K.: Wiley, 2011, pp. 215–255.
- [41] S. Shalev-Shwartz and S. Ben-David, *Understanding Machine Learning: From Theory to Algorithms*. Cambridge, U.K.: Cambridge Univ. Press, 2014.
- [42] J. R. Quinlan, "Induction of decision trees," *Mach. Learn.*, vol. 1, no. 1, pp. 81–106, 1986.
- [43] T. Hastie, "Robert Tibshirani, and Jerome Friedman," in *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, vol. 2. New York, NY, USA: Springer, 2001.
- [44] Y. Freund and R. E. Schapire, "Experiments with a new boosting algorithm," in *Proc. 13th Int. Conf. Mach. Learn.*, Bari, Italy, vol. 96, 1996, pp. 148–156.
- [45] C. M. Bishop, *Neural Networks for Pattern Recognition*. Oxford, U.K.: Oxford Univ. Press, 1995.
- [46] S. Samarasinghe, *Neural Networks for Applied Sciences and Engineering: From Fundamentals to Complex Pattern Recognition*. Boca Raton, FL, USA: Auerbach publications, 2016.
- [47] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20. Boston, MA, USA: Kluwer, 1995, pp. 237–297.
- [48] G. Mountrakis, J. Im, and C. Ogole, "Support vector machines in remote sensing: A review," *ISPRS J. Photogramm. Remote Sens.*, vol. 66, no. 3, pp. 247–259, May 2011.
- [49] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*. Cambridge, U.K.: Cambridge Univ. Press, 2000.
- [50] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, *Data Mining: Practical Machine Learning Tools and Techniques*. San Mateo, CA, USA: Morgan Kaufmann, 2016.
- [51] D. H. Wolpert, "Stacked generalization," *Neural Netw.*, vol. 5, no. 2, pp. 241–259, Jan. 1992.
- [52] S. Valle, W. Li, and S. J. Qin, "Selection of the number of principal components: The variance of the reconstruction error criterion with a comparison to other Methods," *Ind. Eng. Chem. Res.*, vol. 38, no. 11, pp. 4389–4401, Nov. 1999.
- [53] C. Xu, Y. Wang, P. Liu, W. Wang, and J. Bao, "Quantitative risk assessment of freeway crash casualty using high-resolution traffic data," *Rel. Eng. Syst. Saf.*, vol. 169, pp. 299–311, Jan. 2018.
- [54] A. C. Rencher, *Methods of Multivariate Analysis*. New York, NY, USA: Wiley, 2002.



research interests include traffic safety, traffic operation, transportation planning, and artificial intelligence.



University of Petroleum & Minerals. He specializes in optimization of urban traffic flow, traffic safety, traffic impact studies, and intelligent transportation systems (ITS).



Since 2017, he has been a Research Engineer and an Associate Professor with the Center for Environment & Water, King Fahd University of Petroleum & Minerals. He is the author of three books, more than 60 articles, and one invention. His research interests include sustainable transportation, machine learning based modeling, environmental impact assessment, and climate change studies. He is also a member of the Editorial Board of the *Journal of Transportation and Logistics*.



include traffic safety, intelligent transportation systems ITS, traffic operations, and machine learning.

...