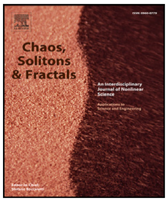
Contents lists available at ScienceDirect

Chaos, Solitons and Fractals

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



Deep learning model of convolutional neural networks powered by a genetic algorithm for prevention of traffic accidents severity

Luis Pérez-Sala [b], Manuel Curado [a], Leandro Tortosa [b], Jose F. Vicent [b,*]

[a] Polytechnic School, Catholic University of Murcia, Campus Los Jerónimos, s/n, E-30107 Murcia, Spain

[b] Department of Computer Science and Artificial Intelligence, University of Alicante, Campus de San Vicente del Raspeig, Ap. Correos 99, E-03080, Alicante, Spain

ARTICLE INFO

Keywords:
Convolutional neural networks
Genetic algorithm
Data analysis
Traffic accidents

ABSTRACT

The World Health Organization highlights that the number of annual road traffic deaths has reached 1.35 million (Global Status Report on Road Safety 2018). In addition, million of people suffer more or less important injuries as a consequence of this type of accidents. In this scenario, the prediction of the severity of traffic accidents is an essential point when it comes to improving the prevention and reaction of the entities responsible. On the other hand, the development of reliable methodologies to predict and classify the level of severity of traffic accidents, based on various variables, is a key component in the field of research in road safety. This work aims to propose a new approach, based on convolutional neural networks, for the detection of the severity of traffic accidents. Behind this objective is the preprocessing, analysis and visualization of data as well as the design, implementation and comparison of machine learning models considering accuracy as a performance indicator. For this purpose, a scalable and easily reusable methodology has been implemented. This methodology has been compared with other deep learning models verifying that the results of the designed neural network offer better performance in terms of quality measures.

1. Introduction

Artificial intelligence has long since left the specter of science fiction to sneak into our lives, it is called to lead a revolution. Its applications, in multiple sectors such as health, transportation, urban mobility or sustainability, among others, have caused an explosion of research work, from different points of view, that are interacting with many parts of our lives. The theory of Artificial Intelligence (AI) has been developing for a decade, but its use has had to wait for advances in the area of information technology, since it requires the development of computer components, mainly fast processors, high-capacity memory or wireless networks. Thanks to this technical progress, artificial intelligence contains sufficient techniques and means to be used in different areas. Thus, neural networks, AI planning, evolutionary algorithms, expert and knowledge systems, fuzzy logic, multi-agent systems, vector regression, data mining or optimization techniques, allow their use in the most various fields.

Focusing on the object of study of this work, traffic accidents, we can say that over the last few years different models have been proposed to analyze their causes and severity. In recent decades, two aspects related to the analysis of the severity of accidents have coexisted: the perspective based in statistical models and that based in machine learning. Statistical models have the characteristic of making certain assumptions about the data, assuming that they are found according to a given probability distribution [1–3]. However, if this premise about the input data is not fulfilled, erroneous results can be produced. On the other hand, machine learning models do not make any assumptions about the data, so they achieve performance similar to or better than statistical ones. In the literature there are multiple models that are applied to the problem of evaluating traffic accidents, such as the implementation of decision rules based on decision trees evaluating the importance of the characteristics [4], or the use of logistic regressions to classify their severity [5].

Other methodologies have recently been proposed applying genetic algorithms. One of the most influential proposals in this field involves the knowledge of road users, with the aim of classifying traffic accidents [6]. In addition, work has been done to train, through evolutionary programming, fuzzy classifiers for the discovery of influential features and important relationships between them [7]. It is common to find combinations of various methodologies seeking to exploit the advantages of each one, carrying out discussions between them, such as the comparison of performance between genetic algorithms combined with pattern search with respect to multilayer perceptrons and artificial neural networks (ANN) applied to prediction of traffic accidents [8]. It is also frequent to use these evolutionary algorithms to optimize the

[*] Corresponding author.
E-mail addresses: lpg95@alu.ua.es (L. Pérez-Sala), mcurado@ucam.edu (M. Curado), tortosa@ua.es (L. Tortosa), jvicent@ua.es (J.F. Vicent).

https://doi.org/10.1016/j.chaos.2023.113245
Received 10 October 2022; Received in revised form 6 January 2023; Accepted 6 February 2023
Available online 15 February 2023

input hyperparameters of other models applied to this problem, such as the hyperparameter optimization of the Support Vector Classifier (SVC) model using the Particle Swarm algorithm to infer the fatality of accidents [9]. Some investigations on the severity of accidents, trying to find key characteristics to predict the severity of injuries focuses both on the performance of the prediction model as well as on the interpretation of the causes of accidents. Thus, in [10] the authors propose the use of data visualization to help solve this problem since data visualization can not only compare multiple characteristics to detect the one that affects the severity of accidents, but it can also visualize variations on these characteristics to distinguish trends. Another research topic that has boomed in recent years is the use of the game theory to solve traffic-related problems. Thus, in [11] the authors propose a dynamical model based on evolutionary game theory, where the traffic authority, drivers and pedestrians compete with each other, forming an evolutionary system in which they interact and influence each other.

The machine learning techniques are proving to be effective in solving classification problems. Consequently, they have been applied to contexts related to traffic incidents, such as the prediction of highway accidents based on neural networks [12]. However, among the large number of deep learning models, the application of convolutional neural networks (CNN) offers very promising results regarding a large number of problems, such as matrix segmentation [13], their classification [14], language recognition [15] or accident severity classification [16] among others.

The nature of convolutional networks requires data input in the form of a matrix and this implies the need to study techniques for the transformation of categorical data to this format. One of these proposals is described in [17] and was originally proposed to capture small variations between DNA sequences. CNN architectures are able to find patterns in input data [18,19] and often provide great performance in many fields using one-dimensional convolutions on the input matrix (see [20–22]). Another type of convolutional neural network is the two-dimensional one, which by applying two-dimensional filters to the input, is capable of learning complex patterns, being widely used in tasks of a very different nature, such as the identification of people through facial recognition [23], classification of scanned documents [24] or even for the detection of extreme weather phenomena [25] among others.

One of the main drawbacks in these studies is usually the low quality of the datasets [26]. In addition, the imbalance of data associated with the nature of the problem, generates an added difficulty to these studies, since, in the case of traffic accidents, most of them are usually minor, with the number of serious and fatal accidents much smaller. There are numerous articles that analyze this problem and different solutions are proposed, such as the use of re-sampling techniques [27] or the definition of new classification metrics.

The main objective of the proposed model is to develop a predictive system for the severity of traffic accidents using characteristics that can be identified at the accident sites, such as the gender of the driver, the type of vehicle or characteristics of the environment, among others. This objective is accompanied by some tasks such as the use of techniques for the transformation of qualitative characteristics of traffic accidents into numerical matrices, the use of machine learning algorithms based on decision trees to infer weights, the use of evolutionary techniques for hyper-parameter optimization or the develop of a deep learning-based approach to predict the severity of traffic accidents. In this work, two architectures based on convolutional neural networks (CNN) have been applied, one-dimensional 1D-CNN and two-dimensional 2D-CNN. Both differ in the size of the kernel and the way it moves due to its dimensionality. Thus, in order to study the behavior of the proposed model, one of the specific objectives is to study its behavior by comparing it with other predictive models. Our proposal has the advantage of being able to be applied in real time since, once the network has been trained, the severity predictions of an accident are generated in fractions of a second. In this way, the process can be monitored and controlled in real time regardless of technological resources.

To achieve this objective, the paper is organized as follows: in Section 2 a detail description of the methodology of the model is presented. The results of the proposed model are discussed in Section 3. Section 4 compares the model with other important models based on deep learning techniques. Finally, some conclusions are presented in Section 5.

2. Methodology

In this section, we describe, in detail, each of the steps that make up the predictive system presented in the paper. Thus, Fig. 1 shows a summary of the different stages that has been designed and developed in order to predict the severity of traffic accidents. To do this, we use a traffic accidents dataset of the city of Madrid in a specific period of time (see [28]).

As can be seen in the flowchart, there are 6 well-differentiated phases in the predictive model: Firstly, a deep analysis of the data is mandatory; then, due to the imbalance of the data a resampling is necessary; thirdly, we apply a Genetic Algorithm with the aim to optimize the hyperparameters that serve as input to the phase four; in this phase, having the balanced data and the optimized hyperparameters as inputs, the weights of the features are calculated by means of Boosting Algorithm; Matrix phase shows the process to construct the inputs of the Convolutional neural networks; finally, we present the Model stage with a twofold objective: to implement the CNN networks (1D and 2D) and to compare with three deep learning models as Gaussian Naive Bayes models, Support Vector Classifier and K-Nearest neighbors.

In general, the convolutional neural networks come with a restriction regarding to the size of the input. Thus, in our proposal, the convolutional neural networks are designed in a way so that they can only accept matrices of a fixed size (5 × 5). To match this expected shape, the input data must be reshaped. This is required because the network makes assumptions about the data it will receive as input, which are built into the network's architecture. Thus, if the input data does not conform to the expected shape, the neural network will be unable to process it properly and may produce incorrect results. Summarizing, the Reshaping process of the Matrix phase is a technical step in the preparation of the input data so that the proposed models based on convolutional neural networks, both one-dimensional and two-dimensional, can learn from them effectively.

2.1. Phase 1: Data

Data is a critical element in deep learning models and for this reason, it is essential that there is variety in them. This diversity of data means that there is a need for this data not to be too simple. The fact of the existence of not very simple data is primordial for the correct implementation of deep learning applications, but this does not mean that the volume of data is not important. Having a lot of data is important for the applications of predictive models, but the variety of data used is considered a more important parameter so that the algorithms can be efficient, and can establish more precise and accurate predictions about future behavior, in this case of traffic accidents.

The dataset used in this project describes traffic accidents of the city of Madrid in a specific period, from 2019 to 2022, and it has been obtained from an open data website [28]. The total number of records in this period is 60,966 and each of which has 18 features described in Table 1 .

The importance of the Severity attribute must be highlighted, since it is the predictive variable in this work. There are different types of values assigned to this category, so each data belongs to at least one of the following cases:

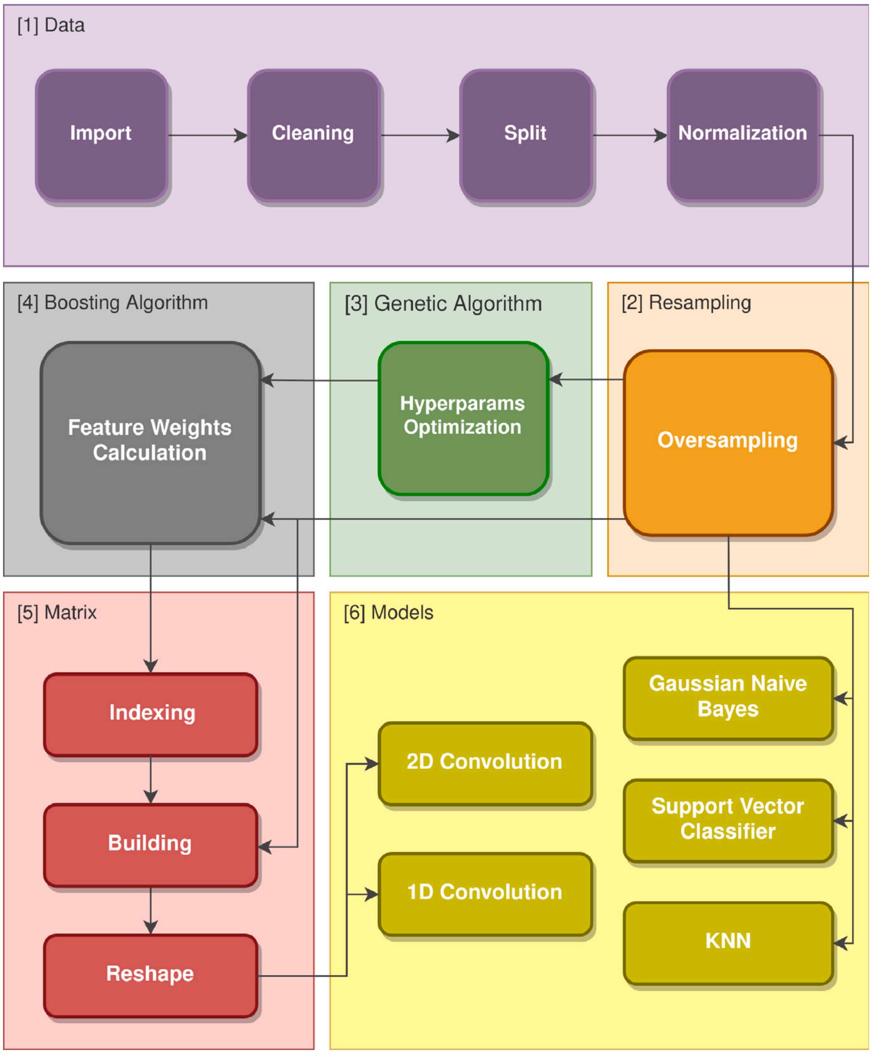


Fig. 1. Flow chart of the proposed model with its different phases.

1. Slight: this ranges from those who have not been injured to those who have needed to be admitted to a hospital for no more than 24 h. The numerical quantification is:
   - Emergency care without subsequent hospital admission: 1.
   - Hospital admission less than or equal to 24 h: 2.
   - Ambulatory health care after the accident: 5.
   - Medical care only at the accident site: 6.
   - Without healthcare: 7.
2. Severe: those involved who have required hospital admission for more than 24 h. In this case the numerical quantification is:
   - Hospital admission for more than 24 h: 3.
3. Fatal: fatalities within 24 h after the accident. The numerical assignment to this field is:
   - Died within 24 h: 4.

Once the data has been imported, they are cleaned by choosing the features used as explanatory variables in the predictions and also considering the values of the instances, since they may contain outliers.

We delete the following variables as they are not relevant in the predictive model: IncidentID, Date, Name and Street Number. In addition, the Alcohol and Drug columns have been merged into a new column due to the number of null values that existed in Drugs variable, so a new column is created that refers to alcohol or drug intoxication.

With this in mind, the following characteristics have been selected as explanatory variables: **Severity, Time, District, X coordinate, Y coordinate, Type of accident, Type of Road, Weather conditions, Vehicle, Person, Age, Gender and Alcohol or Drugs**. Therefore, final number of rows of the dataset is 54,364 rows.

It is necessary to carry out transformations on the data because of deep learning models used require with a well-defined and consistent set of data. This means that numerical and normalized input data are mandatory. Therefore, we must transform categorical to numerical variables.

First of all, it has been necessary to transform the X and Y coordinates to number, since these variables were initially of type String. In addition, the range of these variables is between 7 to 10 digits without any standardized decimal format. Because of this, it has been necessary to perform a process that analyzes each case and translates them into a standardized format.

As there is no service that offers the typology of the street given the name of the road or given the coordinates, it has been necessary to deal with this problem from another point of view, through the use of regular expressions [29]. In general it can be said that a regular expression is a type of pattern used to match a combination of characters in a text string. In this work, to classify the type of roads, a series of regular expressions have been designed that match certain criteria regarding to the values of this variable. This has been made in order to be able to distinguish different values of Type of Road. On the other hand, it is

Table 1
Variables of the dataset and their descriptions.

Attribute	Description
Incident ID	Incident identifier, if several records have the same file number, they are considered the same accident and each record represents each of the different people involved in it (Driver, Passenger or Pedestrian).
Date	Day, month and year in which the incident occurred.
Time	Hour and minute in which the incident occurred.
Type of Road	Type of road where the incident occurs.
Name	Name of the street where the incident occurs.
Street number	Street number where the incident occurred.
District	Name of the district where the incident occurred.
Type of accident	It can be: double collision, multiple collision, range, collision with an obstacle, run over, rollover, fall, or other causes.
Weather conditions	Weather conditions at the time of the incident.
Vehicle	Classification according to the types of vehicles.
Person	Role of the person involved: driver, passenger or pedestrian.
Age	Age range of the person involved.
Gender	Woman or man.
Severity	Physical consequences of the person involved, if they have needed health care, if they have been hospitalized or if they have been fatal.
X	X - UTM coordinate.
Y	Y - UTM coordinate.
Alcohol	If the person involved has tested positive for alcohol (Y or N).
Drugs	If the person involved has tested positive for drugs (Y or N).

Table 2
Numerical assignment of the dataset variables.

Features	Typing
Severity	0 **Slight** $(1,2,5,6,7)$; 1 **Severe** (3); 2 **Fatal** (4).
Time	1 Night (6 PM –6 AM); 2: Day (6 AM –6 PM).
District	Based on order of appearance.
X	UTM X Coordinate position.
Y	UTM Y Coordinate position.
Type of Accident	1 Head-on-size collision; 2 Rear-end collision; 3 Side crash; 4 Collision again fixed obstacle; 5 Pile-up; 6 Hitting a pedestrian; 7 Head-on collision; 8 Other; 9 Leaving the road; 10 Vehicle rollover; 11 Hitting an animal; 12 Falling.
Type of Road	1 Parking; 2 Airport; 3 Park; 4 Tunnel; 5 Industrial state; 6 Track; 7 Round; 8 Roundabout; 9 Gate; 10 Bridge; 11 Square; 12 Blvd.; 13 Crossing; 14 Roadway; 15 Road; 16 Avenue; 17 Highway; 18: Street.
Weather Conditions	1 Sunny; 2 Cloudy; 3 Light rain; 4 Heavy rain; 5 Hail; 6 Snowing; 7 Unknown
Vehicle	Based on order of appearance.
Person	1 Driver; 2 Passenger; 3 Pedestrian.
Age	1 Under 18 years of age; 2 From 18 to 25 years old; 3 From 25 to 65 years old; 4 Over 65 years old; 5 Unknown.
Gender	1 Male; 2 Female; 3 Unknown.
Alcohol or Drugs	1 Yes; 2 Not.

necessary to treat those values that do not match defined patterns. This has been done automatically, all of them coinciding with street names to which the word "street" has not been assigned. The Type of road typologies are described in Table 2.

The Time variable took on any value within a range, and the number of possible values within that range is infinite. Due to this, it has been necessary to discretize it according to intervals, distinguishing between night and day based on the time at which the incident occurred. Note that various ranges of values were tested, concluding that they had no influence on the final predictions.

Regarding to the Weather Conditions, a study of their values was made and it was found that all of them could be included in those described in the Table 2.

In the Age variable, several range values were tested showing no influence on the final predictions.

Finally, the encodings applied, in the predictive model, for the quantification of the variables is given in Table 2.

It is interesting to observe the degree of correlation between the variables, so that if two highly correlated variables are found, one of them can be deleted (see Fig. 2). The computation of the Pearson's correlation coefficient is used to determine the degree of correlation among the variables.

After analyzing the correlation matrix, it can be concluded that the most correlated variables are the X coordinate and the District, with a relatively median correlation of 0.44. However, it is not a value so high as to eliminate either of the two characteristics. Therefore, it can be concluded that none of the variables are left over and it is possible to move on to the next phase.

Data normalization is a necessary process when it comes to obtaining good results in predictive machine learning models. When a model is trained, there are features represented on different scales, and those that contain a higher range of numerical values, either due to the nature of the variable or because they are in another range, dominate those that are in a minor range, influencing negatively in the predictions of the machine learning models [30]. Thus, the normalization process aims to minimize the bias of those features whose contribution is greater when it comes to finding patterns in the data. There are different normalization techniques such as Mean Centered (MC), Variable Stability Scaling (VSS) or Min–Max Normalization (MMN) among others [31]. In this work, the Z-Score Normalization (ZSN) has been used because it achieves representations according to a normal distribution. To do this, the mean and standard deviation are used to re-scale the data so that their distribution is defined by a mean of zero and a unit standard deviation.

Another important aspect of any machine learning model is the data separation (split). This is done by splitting the total dataset into two subsets: training and test. The model is trained based on the training set and is evaluated with the test set. Then, the results on the test set allow us to compare the models based on the predictions on the samples that they have never seen. Commonly the ratio of training and test data is 80% and 20%, respectively. In our case, this proportion has been chosen, so we have a total of 43,603 accidents for the training set and 10,901 for the test set.

It is important to point out that the training phase is contextualized to a specific use case but only needs to be performed once supporting training from scratch without the need for pre-trained model weights.

2.2. Phase 2: Resampling

If the data corresponding to each of the three possible classes of Severity (Slight, Severe and Fatal) are analyzed, it can be shown that the dataset is clearly unbalanced with respect to the mentioned variable. There are 53,009 minor accidents, 1,271 severe and 84 fatal. This becomes a problem for classification models since they tend to predict the samples as those that belong to the majority of the test set.

Data imbalance is a problem widely studied over the years and there are numerous methods aimed at solving it through different sampling techniques. In this work, the data has been resampled using the Borderline Synthetic Minority Over-sampling Technique 2 (SMOTE-II) (see [32]). This algorithm has been used to generate more samples of

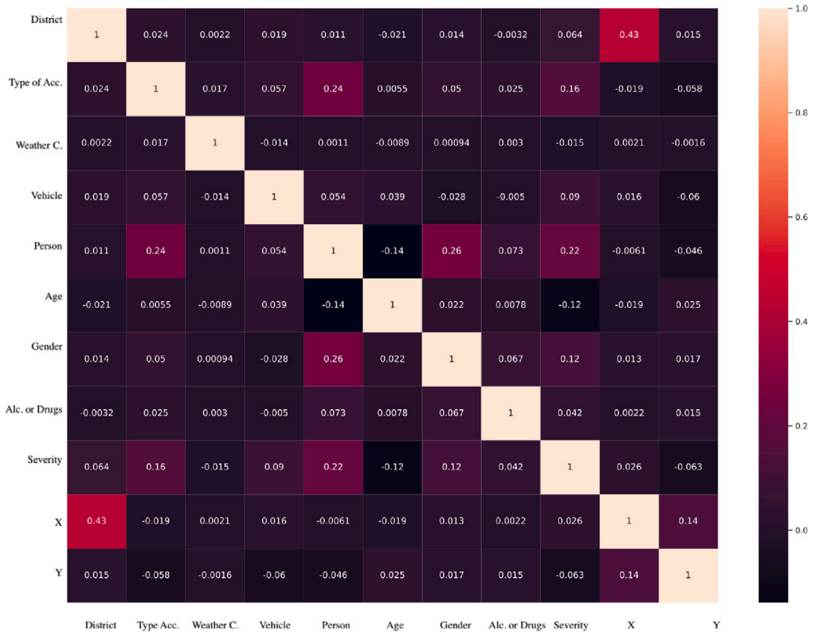


Fig. 2. Correlation matrix between the dataset variables.

accidents belonging to minority classes (Severe and Fatal) preventing the model from overfitting. Once the SMOTE-II algorithm is applied, 42,508 samples are obtained from each of the accident classes.

2.3. Phase 3: Genetic algorithm

Once all the data processing is done, the next step is to calculate the optimal hyperparameters of the Boosting Algorithm responsible for calculating the weights of the variables.

A Genetic Algorithm is a model that mimics the evolution of species in the field of biology, with the aim of finding a potentially optimal solution to a problem [33]. It is based on a series of phases such as initialization, in which each individual of the population, formed by the set of variables that are the object of optimization, is initialized. Evaluation, whereby each individual in the population is evaluated using a fitness function. Selection, where the individuals with the best fitness value are selected. Crossover, in this phase, information is exchanged between parents to give rise to new individuals. Finally, Mutation generates diversity in the population. By means of a genetic algorithm we can optimize the values of a fitness function.

In this work, a type of supervised classification algorithm has been used to calculate the weights assigned to each of the variables of the dataset. For the optimization of the hyperparameters included in this Boosting Algorithm, a genetic algorithms is used. In this algorithm, each individual is a specific solution to the hyperparameter values, in such a way that throughout the iterations the individuals evolve (through crossing and mutation) to give rise to new configurations of optimized hyperparameters.

The large number Boosting Algorithm hyperparameters makes the solution space enormous. Due to this, a subset of those that have the most influence on model training has been selected. More specifically, the following parameters have been optimized:

1. Maximum depth: It is the maximum height that the tree can take. If the decision tree reaches too deep it will tend to be over-fitting as it will learn complex relationships between the data that may be due to noise in the training data.
2. Minimum weight of children: It is the minimum weight that is established when creating a new node in the tree. When a decision tree is trained, it generates new nodes based on the maximum separability of the training data at each level. With the weight limit of the children, we establish a minimum threshold of samples that must belong to a node to carry out the separation. A low value in this parameter will allow to create nodes with fewer samples and therefore the model will tend to over-fitting.
3. Eta: Step size used to apply gradient descent to minimize loss of previous trees.
4. Gamma: A node is split only when the resulting split gives a positive reduction in the loss function. It specifies the minimum loss reduction required to make a split.
5. Alpha: It is a L1 regularization parameter, increasing its value makes the model more conservative.
6. Lambda: It is a L2 regularization parameter, increasing its value also makes the model conservative.

The other parameters are not optimized because of they are chosen by default since they do not affect the optimization result.

In evolutionary algorithms, the initialization and mutation of the values of individuals are given by a minimum and maximum limitation. If this restriction is not considered, the hyperparameters could take extreme values, thus reducing the training and prediction performance. Therefore parameters of the new solutions are within the specific range.

Once the individuals of the population have been randomly initialized, they are evaluated. The fitness function is the Micro F1-score metric and the goal is to optimize the aforementioned metric. For that purpose, in each generation it is checked if there is any individual in the population with better Micro F1-score that the best individual of the moment. Once the individuals of a generation have been evaluated, the 15 best will be crossed to give rise to new children, which have a mutation probability for each of their characteristics. Finally, the parameters of the individual with best Micro F1 score are the hyperparameters sought.

2.4. Phase 4: Weights of the features

Once the hyperparameters of the classification algorithm have been optimized by means of the genetic algorithm, they are included to calculate the weights of the variables used in the dataset by means of the Boosting Machine Learning Ensembles Algorithm (BMLEA). This algorithm is used for classification and regression and it builds a robust model by combining a series of weak models applying regularization

Table 3
Classification of the features (dataset variables) in categories.

Category	Features
Accident	- X - Y - Time - Type of accident - Severity
Road	- Type of road - District
Weather	- Weather conditions
Vehicle	- Vehicle
Driver	- Person - Gender - Age - Alcohol or Drugs

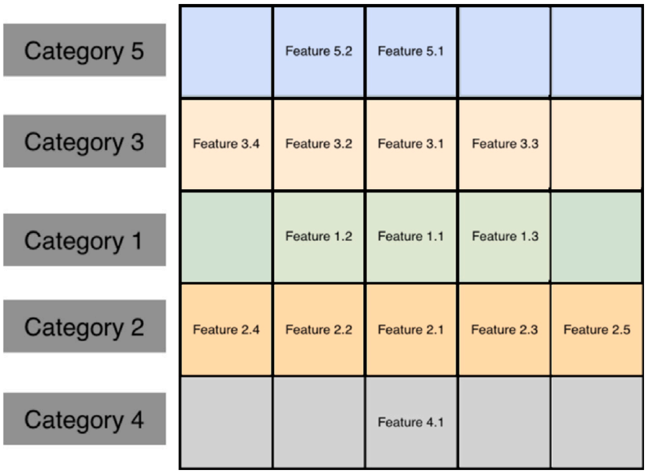


Fig. 3. Example of positioning the elements in a matrix. Categories are assigned in rows based on their weight, and Features are assigned in the columns of the corresponding Category based on their weight.

techniques to its loss function [34]. The inputs of the BMLEA algorithm are the optimized hyperparameters and the resampling traffic accidents dataset while the output are, for each row of the input dataset, the weights of the twelve different characteristics analyzed.

The weights obtained play a key role in our proposal because of they indicate the degree of importance assigned to each feature of the dataset. To calculate these weights, the training of N sequential decision trees is used, in which each of them tries to minimize the error produced at the end of the classification of its predecessor tree. In such a way that they are sequentially nested with the aim of minimizing the prediction error. It reduces the individual influence of each generated tree and its leaves in order to give rise to subsequent trees that manage to improve the model.

In the BMLEA algorithm, the calculation of the weights of each of the input characteristics is based on obtaining, numerically, the influence of each variable when constructing each of the N sequential classifiers. The value of this influence is calculated by analyzing the performance obtained by the classification of each characteristic, calculated by means the Gini Index, with respect to the number of samples that this characteristic has managed to divide.

Remark that for each row of the input resampling dataset twelve values are obtained, one per variable.

2.5. Phase 5: Construction of the matrices

Convolutional neural networks (CNN) learn patterns using arrays as input data. This implies the application of techniques that position each feature in an element of the matrix, maximizing the accident information. Therefore, with the normalized and resampled data and the optimized hyperparameters, we must transfer each variable of the accident to a matrix that is the input of the convolutional neural network. To achieve this goal the features of a traffic accident are divided into a series of categories: Accident characteristics, Road conditions, Weather conditions, Vehicle characteristics and Driver characteristics (see Table 3). Summarizing, an assignment of each the explanatory variables of the dataset in one of the five categories is made.

As can be observed in Table 3, we have five categories and in each of them there are, at most, five features. This data structure may be represented by a matrix where each category is placed in rows and each feature in columns.

Once the hierarchy of characteristics of the traffic accidents has been defined, as well as the weights associated with each of them, the input matrices of the convolutional networks are constructed. Remark that a matrix of size (5×5) is constructed for each set of variables that represent an accident, that is 54,364.

The assignment of the rows of the matrix to the categories is done in an interleaved way based on their weight. The most important category is positioned in the central row of the matrix, the second category is positioned above it and the third below and so on. It is necessary to point out that the importance of each category is given by the sum of the weights of their features.

Once the categories have been assigned to the rows, the same procedure is carried out with the features at the column level. These features are assigned in a position within the row of their category, where the one that has the most importance is positioned in the center, the second is positioned to its left, the third to the right and so on.

The purpose of positioning the most influential categories and features in the central areas of the matrix is due to the fact that the CNN's use the kernels to go through the matrix based on a displacement. Therefore, these kernels convolve more times at the positions where the most influential features are found.

An example of the construction of these matrices is shown in Fig. 3 and it can be summarized in the following steps:

1. Generation of n arrays of 5×5 initialized to 0, where n in the number of elements of the dataset.
2. Allocation of a row to each category based on their weight.
3. Assignment of feature, based on its weight, within its Category's row.

Note that all features are normalized under the Z-Score Normalization model, so that, each array contains normalized values of each feature.

2.6. Phase 6: Convolutional neural network model

Neural networks (NN) are models that emulate the behavior of the brain when processing information, training on a set of data to identify patterns between them [35]. A convolutional neural network (CNN) is a neural network that uses convolution [36] and this is a mathematical operation that allows to merge sets of data. The convolution is applied to the input data to filter the information and produce a feature map. This filter is called a kernel and their dimensions can vary, although 3×3 dimension kernels are common. To perform convolution, the kernel goes over the input matrix, doing matrix multiplication element after element.

In this section, we analyze the two convolutional neural network architectures 1D-CNN and 2D-CNN, respectively, that have been applied in the propose model. It is worth mentioning that the operation of the *1D-CNN* and *2D-CNN* only differs in the size of the kernel and the way in which it moves due to its dimensionality; therefore both architectures are detailed in the same way.

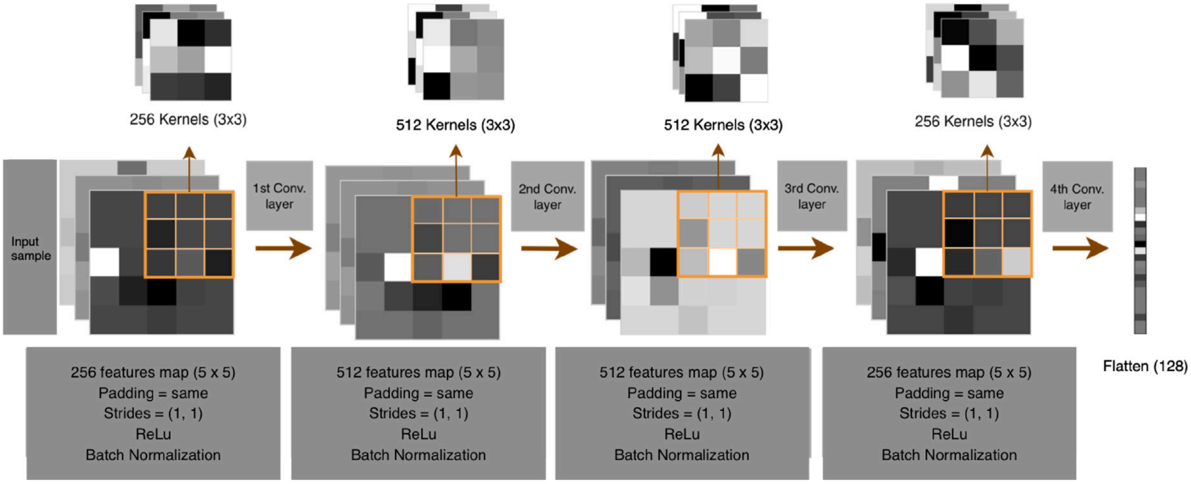


Fig. 4. Architecture of the 2D-Convolutional neural network.

The architecture consists of four convolutional layers with kernel sizes of 1 × 3 for 1D-CNN and 3 × 3 for 2D-CNN. These kernels are projected into 256 or 512 channels to form the convolutional filter associated with each layer. A batch normalization process is applied to the output of each of the feature maps.

The kernel padding has been set to 1 for both types of networks, so that convolutions will be applied by adding zeros to the boundaries of the arrays, and strides to 1 for 1D-CNN and $\{1, 1\}$ for 2D-CNN. Therefore, the shifting of the kernels is done pixel by pixel in both convolutional networks.

At the output of each convolutional layer, the activation function Rectified Linear Unit (ReLU) is applied.

The output of the last layer of the convolution transforms the generated feature map matrix of size 5 × 5 into a layer that will flatten the matrix to a one-dimensional vector of 1 × 25. Next, a dense layer is applied connecting each of the 25 nodes of the Flatten layer with the 128 nodes of the dense layer, which generates the logits before applying the last Softmax activation function that returns the predicted class.

To exemplify the architecture of the proposed CNN, the case of 2D-CNN is shown in Fig. 4.

During the training, both convolutional networks have been set to a 23 batch size of 32 samples, and dropout layers of 0.2 rate have been present between each convolutional layer.

2.7. Quality measures

There are quite a few metrics used to evaluate machine learning classification models. However, the most popular are:

Precision: This is the percentage of success when classifying a class, representing the percentage of correct accidents of each type with respect to the total number of accidents predicted.

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

Recall: This is the percentage of total identifications of the model for a class and it represents the percentage of correct identification.

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

F1-Score: This is a metric that summarizes Precision and Recall in a single value to show how well the classification of true positives is carried out regarding to the cost of false negatives that this entails. It is the most used ranking metric.

F1 Score $= 2 \cdot \frac{\text{Precision}\cdot\text{Recall}}{\text{Precision}+\text{Recall}}$.

Confusion Matrix: In this type of matrix, the X axis shows the labels predicted by the model and the Y axis shows the labels of the true classes. In this way, the confusion matrix allows visualizing how the model classifies each one of the classes and observing to which labels each sample is assigned on any dataset.

Table 4
Optimized values of the parameters after applying the genetic algorithm.

Hyperparameters	Value
Max deep	2
Minimum weight of children	1.6
ETA	0.007
Gamma	0.3
Alpha	0
Lambda	1

3. Numerical results

In order to carry out a discussion and a comparative analysis, the results of each of the model phases and the final experiments are detailed. It should be noted that we start from typified, normalized and balanced data.

Firstly, the optimization of the hyperparameters of the Boosting algorithm by means of a genetic algorithm is analyzed. Fig. 5 shows the evolution of three hyperparameters throughout the course of the iterations. As it can be seen, the hyperparameters try different values until they converge approximately at iteration 42.

It should be noted that, to simplify Fig. 5, only the three hyperparameters that vary the most are shown (*eta, max depth* and *min child weight*).

As the best individual obtained in the execution of a genetic algorithm depends on the initialization of the population, 80 executions of the algorithm with its different initialization have been carried out. Note that an individual of the genetic algorithm is made up of several variables representing the hyperparameters. Then, we calculate the mean of the best hyperparameters found in each execution of the genetic algorithm, obtaining the values shown in Table 4.

Once the main optimal hyperparameters have been obtained, the Boosting Machine Learning Ensembles Algorithm (BMLE) is executed, with the aim to calculate the weights of the dataset variables. After applying this algorithm, the weight of the variables for each accident are obtained. For instance, Table 5 shows the weight of a specific register (accident) in which it can be observed that the Features Person, Type of Road and Gender are the most influenced, obtaining weights of 0.177, 0.127 and 0.111, respectively.

Two facts must be pointed out: on the one hand, each category's weight is obtained adding the weights of each feature. On the other hand, the results obtained by the BMLE algorithm have been obtained by establishing a seed on the division of the training and test data, in order to be able to reproduce the values of the weights in later experiments.

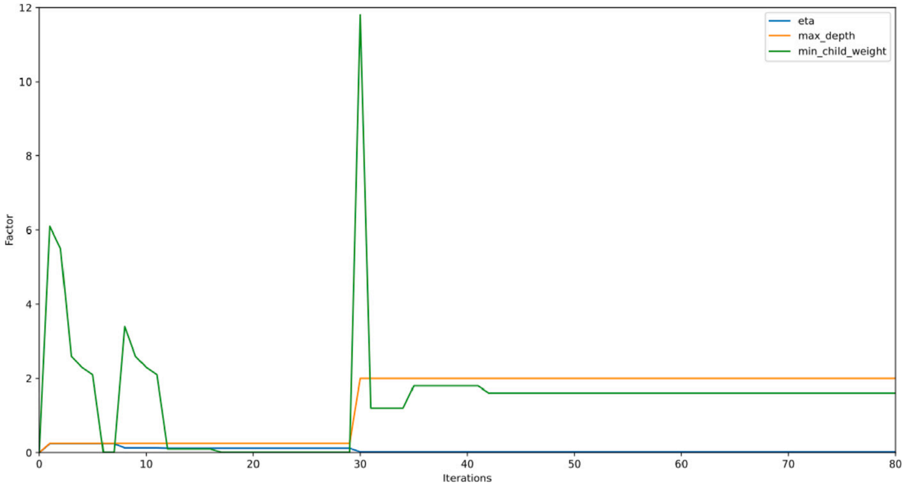


Fig. 5. Evolution of hyperparameters throughout the iterations.

Table 5
Example with the weights of all the characteristics studied, as well as the weights of the five categories.

Category	Weight category	Feature	Weight feature
Accident	0.299	- X coordinate - Y coordinate - Time - Type of accident - Severity	0.071 0.066 0.055 0.051 0.057
Road	0.187	- District - Type of Road	0.059 0.127
Weather	0.050	- Weather conditions	0.050
Vehicle	0.070	- Vehicle	0.070
Driver	0.394	- Person - Gender - Age - Alcohol or drugs	0.177 0.111 0.050 0.056

Once the weights are calculated, they are placed in a matrix 5 × 5 in the positions given by a hierarchy of categories. As can be seen below, the construction of the matrices is made positioning categories in rows and features in columns.

$$\begin{bmatrix} 0.0 & 0.0 & 0.05 & 0.0 & 0.0 \\ 0.0 & 0.059 & 0.128 & 0.0 & 0.0 \\ 0.050 & 0.111 & 0.177 & 0.056 & 0.0 \\ 0.055 & 0.066 & 0.071 & 0.057 & 0.051 \\ 0.0 & 0.0 & 0.070 & 0.0 & 0.0 \end{bmatrix}.$$

Summarizing, the process followed by a record of the original dataset, until obtaining a matrix of characteristics, is as follows: First, the values of the original typified data are normalized according to the ZSN criterion to become an observation whose values are bounded in a range based on the normal distribution. Then, the weights of the features are obtained by means of a BMLE algorithm. The construction of the matrices is made and, finally, all these matrices are the input to two convolutional neural networks (1D-CNN and 2D-CNN). Finally, once the matrices have been constructed, the convolutional neural network described in the propose model is used.

The objective of the training process of a convolutional neural network is to evolve from a low-performance network to a high-precision one. To do this, a series of characteristics must be predefined, such as the percentage of use of the original dataset or the number of training phases. In the proposed model, 80% of the original dataset is used and, in terms of training phases, there are two: one forward and one backward. In the forward phase, the input passes completely through the network and in the backward phase, the gradients are propagated backwards and the weights are updated.

Table 6
Training metrics for 1D-CNN and 2D-CNN.

Metric/Severity	1D-CNN			2D-CNN		
	Slight	Serious	Fatal	Slight	Serious	Fatal
Precision	0.701	0.696	0.754	0.488	0.646	0.966
Recall	0.724	0.523	0.917	0.974	0.299	0.524
F1-score	0.712	0.597	0.828	0.650	0.409	0.679

Translated into loss function terms, the objective of the training process is that the loss function is much smaller at the end of the training than at the beginning. This is possible since the loss function can be modeled by adjusting their weights. All this leads us to redefine the training problem in terms of minimizing the loss function.

Thus, the evolution of the loss function and the prediction results based on confusion matrices and classification metrics are studied.

Figs. 6 and 7 show the evolution of the F1-score metric over the 100 times for the 1D and 2D convolutional neural networks. Visualizing the one-dimensional convolutional (Fig. 6), it can be verified that the training F1-score increases slightly over the epochs, experiencing ups and downs as the model is trained, initially starting from a training value less than 0.58 and reaching up to 0.68.

On the other hand, Fig. 7 shows the training and validation graph of the two-dimensional Convolutional neural network. We observe that the trend of the loss function on the training dataset is stable. It can be seen how the network in the first execution starts with a F1-score of 0.62 until reaching 0.78 at the time 100, so it can be deduced that this network achieves a better performance on the training set regarding the one-dimensional convolutional network.

Tables 6 and 7 detail the metrics resulting from the classification of the networks for the training and test sets. Remark that, for the training set, the 1D-CNN model obtains a better F1-score on the classification of all accident classes compared to the 2D-CNN network. However, when the metrics of the test data are analyzed, the model that presents the best F1-score for Slight and Serious accidents is 2D-CNN, with 0.950 and 0.148 respectively, while in Fatal both networks are tied with 0.004.

Fig. 8 shows the confusion matrices for the training and test sets of both models, where predictive trends can be analyzed. For the 2D-CNN training data, there tends to be a greater propensity to predict observations as Slight accidents compared to the 1D-CNN network. This causes 1D-CNN to correctly classify more Serious and Fatal accidents. For the test dataset, the 2D-CNN network classifies more observations as Slight accidents than the 1D-CNN model, predicting fewer Serious accidents but with greater confidence.

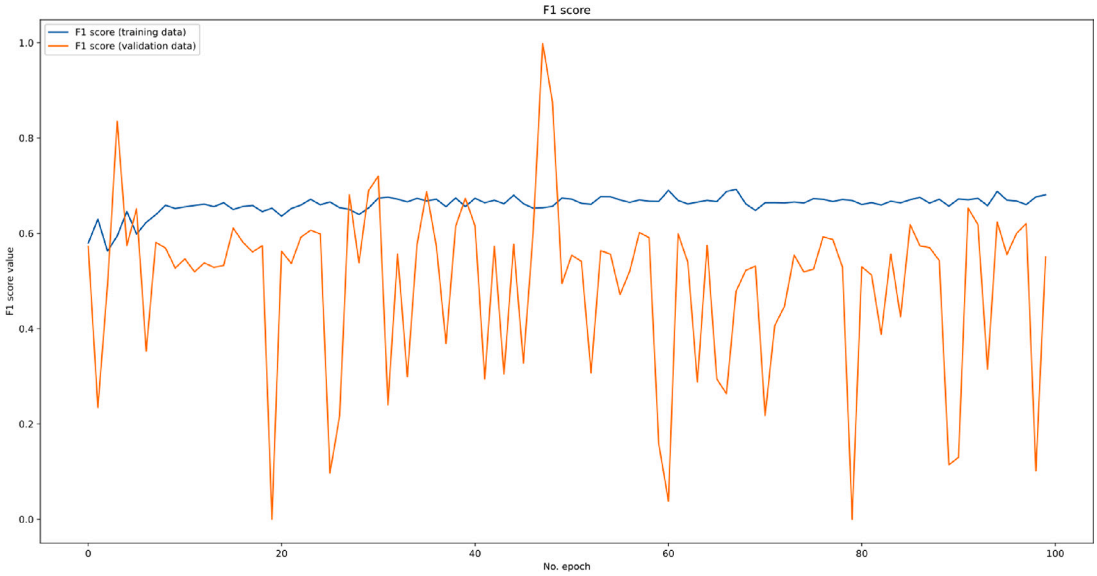


Fig. 6. Evolution of the F1-score of the 1D-CNN in the training and test set.

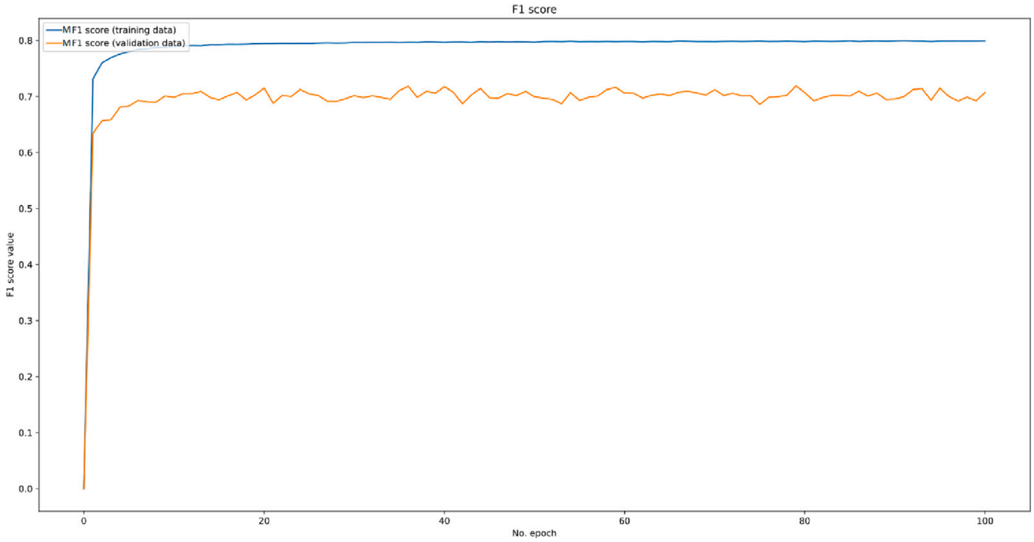


Fig. 7. Evolution of the F1-score of the 2D-CNN in training and test set.

Table 7
Test metrics for 1D-CNN and 2D-CNN.

Metric/Severity	1D-CNN			2D-CNN		
	Slight	Serious	Fatal	Slight	Serious	Fatal
Precision	0.984	0.031	0.002	0.982	0.097	0.002
Recall	0.429	0.394	0.333	0.919	0.313	0.1
F1-score	0.596	0.058	0.004	0.950	0.148	0.004

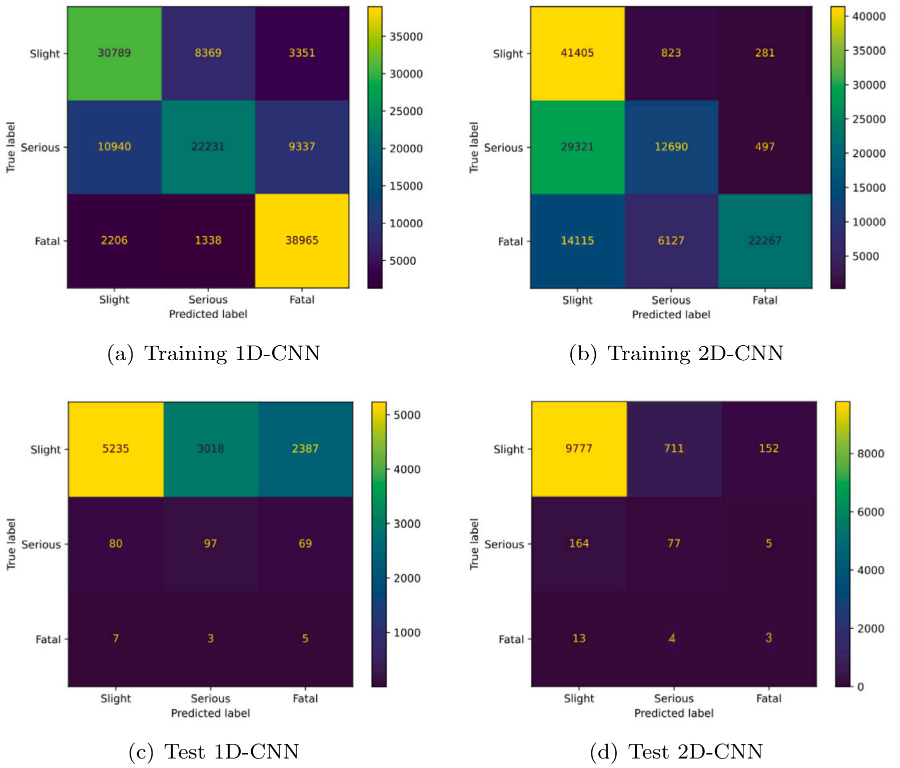
4. Model comparisons

To verify the effectiveness of the presented model, it is compared with three deep learning models. This is very important since the objective of a predictive model is the classification of future data. The three machine learning models are:

- Naive Bayes (NB): It is a probabilistic classifier based on the Bayes theorem, which assumes certain independent assumptions about the predictors to carry out the classifications [37]. To use this classifier with real values, the Gaussian distribution of the characteristics is assumed.
- Support-Vector Classifier (SVC): It is a model oriented to multiple classifications, which are based on projecting the input data in a multidimensional space [38].
- K-Nearest Neighbors (KNN): It is an algorithm widely used for classification and regression tasks [39].

All the experiments have been executed under a server with a Dual AMD Rome 7742 CPU (128 cores) and with a 40 GigaByte DGX NVIDIA A100 GPU. In addition, the propose model and the three machine leanings models used to conduct the comparison, have been implemented in Tensorflow and Scikit-learn, including GridSearchCV, to optimize models hyperparameters.

In Table 8 is shown the resulting classification metrics for each of the classes predicted on the test set. These reports show the information with which we evaluate the models since it explains how they behave regarding new data. As can be seen in this table, the KNN model obtains better results in all measures of all classes except Recall in Serious accidents in which the GNB is a little better.



(a) Training 1D-CNN

(b) Training 2D-CNN

(c) Test 1D-CNN

(d) Test 2D-CNN

Fig. 8. Confusion matrices for convolutional neural networks.

Table 8
Test metrics classification for GNB, SVC and KNN.

Metric/Severity	GNB			SVC			KNN		
	Slight	Serious	Fatal	Slight	Serious	Fatal	Slight	Serious	Fatal
Precision	0.980	0.025	0	0.979	0.029	0	0.982	0.042	0.001
Recall	0.369	0.699	0	0.644	0.411	0	0.689	0.382	0.067
F1-score	0.536	0.048	0	0.777	0.054	0	0.810	0.076	0.002

If we analyze the Precision metric, it can be observed that the model that presents the best average for Slight classes is the 1D-Convolutional Neural Network (1D-CNN) with 0.984, followed by 2D-Convolutional Neural Network (2D-CNN) and KNN model with 0.982. In addition, the 2D-CNN also offers the best metric for Serious accidents 0.097, with great difference with respect to the model that follows KNN with 0.042. Regarding to the Fatal accidents, both models 1D-CNN and 2D-CNN have similar value, obtaining 0.002.

Regarding to the Recall metric, the best average for Slight classes is the 2D-CNN with 0.919, followed by KNN model with 0.689. Moreover, the GNB model offers the best metric for Serious accidents 0.699. In Fatal accidents, 2D-CNN have the best value with 0.1.

It is necessary to point out that the F1-score is a way of combining the Precision and Recall metrics, and it is defined as the harmonic mean of the model's Precision and Recall. Taking this into account, if we analyze the F1-score of the reports, the model that presents the best average for Slight classes is the 2D-CNN, reaching 0.950, well above the following KNN model, which offers a value of 0.810. In addition, the 2D-CNN also offers the best metric for Serious accidents 0.148, reaching twice the performance compared to the model that follows it, the KNN with 0.076. Regarding to the Fatal accidents, the models with best classification are both 1D and 2D CNN, obtaining 0.004, double that KNN, which are the next best models in this class with 0.002.

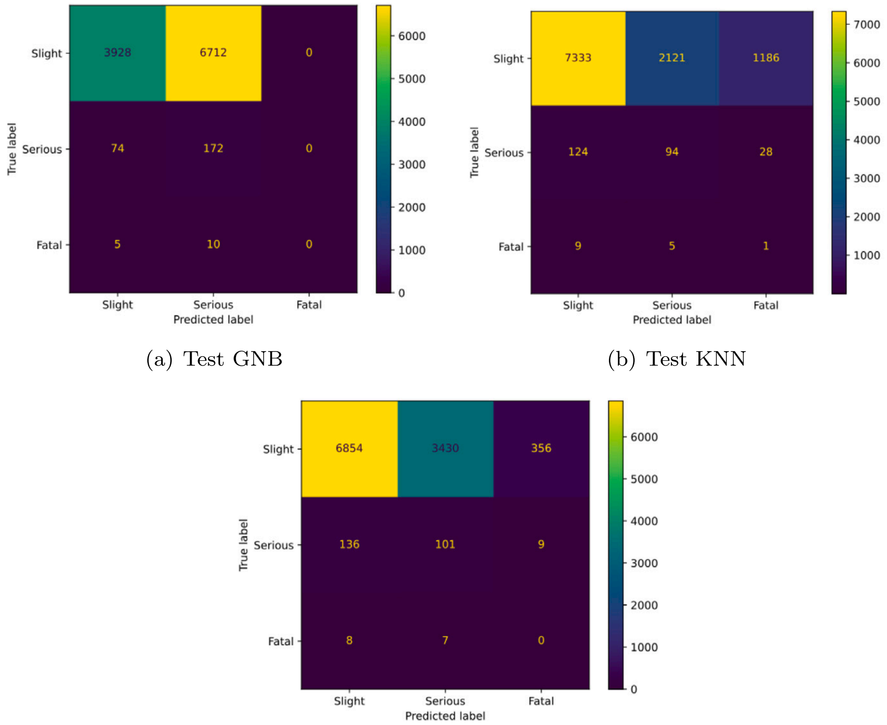
We can conclude that the proposed model, based on convolutional neural networks presents better predictions regarding the F1-score metric, which is a combination of Precision and Recall.

Fig. 9 shows the confusion matrices applied to the test set, showing a visual representation of the classification metrics. This makes it possible to deal with the fact that the GNB and SVC models do not correctly classify any of the Fatal accident observations, while KNN classifies one. With regard to Slight accidents, the 2D-CNN model is the one that correctly classifies the highest number, as well as Serious accidents. This is due to the nature of the networks applied to the complexity of the problem, each one of them finds different patterns depending on the characteristic maps resulting from the convolutions of each network.

As can be seen in the results of the experiments, the two proposed convolutional architectures outperform the rest of the reference models for each of the accident classes (Slight, Severe and Fatal). The advantage of having these two new architectures is that each of them works better depending on the type of class to be predicted. This allows the design of a system in which the two trained networks are used to combine their results, in such a way that the 1D-CNN network would be used to predict Slight and Fatal accidents while the 2D-CNN network would classify Serious accidents.

5. Conclusions

Various types of models have been proposed to achieve the prediction of the severity of traffic accidents, from statistical models to models based on machine learning. In this paper we present a framework based on one and two-dimensional convolutional neural networks that uses a dataset related to the city of Madrid (Spain), with variables grouped into categories such as Accident, Road, Weather, Vehicle or



(a) Test GNB

(b) Test KNN

(c) Test SVC

Fig. 9. Confusion matrices applied to the test set.

Drivers. The transformations on the variables of the dataset are a critical point; thus, we carry out a study of them to check if other typifications on the variables have a positive effect on the performance of the classifications. Once the data are preprocessed it can be seen that the dataset is clearly unbalanced and, due to this, the data has been resampled using the Borderline Synthetic Minority Over-sampling Technique 2 (SMOTE-II). Then, the proposed model calculates a series of weights of the variables of the dataset. For that, it is used a type of Boosting algorithm that has, as input, the optimized hyperparameters (by means a Genetic Algorithm) and the resampling data of the traffic accidents and, as an output, the intended weights. With these weights a set of matrices are constructed (one per accident) that serve as input to convolutional neural networks (1D and 2D CNN).

After training the model, a comparison with three deep learning models are made, verifying that our model presents the best results in the three selected metrics (Precision, Recall and F1 score) in the prediction of Slight, Fatal and Severe accidents.

Among the advantages of the proposed architecture is its scalability since it can be applied to other datasets without the need to make major changes to the implementation. The proposed architecture is trained based on a series of predefined attributes. Because of this, it is possible to perform a fine tuning to apply it in another location, if similar characteristics are available, with a relatively small amount of data (one-shot learning). This leads us to think, as future work, about applying it to other datasets, as well as increasing, if possible, the number of features in the dataset. Another advantage of the proposed predictive model is its application in real time. With the technical specifications mentioned in the paper, severity predictions of an accident are generated in split seconds. Thus, the process can be used in real time even if the institutions that use it have limited technological resources.

CRediT authorship contribution statement

Luis Pérez-Sala: Design and implementation of the research, Methodology, Formal analysis of the results, Writing of the manuscript. **Manuel Curado:** Design and implementation of the research, Methodology, Formal analysis of the results, Writing of the manuscript. **Leandro Tortosa:** Design and implementation of the research, Methodology, Formal analysis of the results, Writing of the manuscript. **Jose F. Vicent:** Design and implementation of the research, Methodology, Formal analysis of the results, Writing of the manuscript.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The data is available on the website mentioned in the paper.

Acknowledgement

Financial support provided under grant number PID2020-112827GB-I00 funded by MCIN/AEI/10.13039/501100011033.

References

[1] Zajac Sylvia S, Ivan John N. Factors influencing injury severity of motor vehicle–crossing pedestrian crashes in rural connecticut. Accid Anal Prev 2003;35(3):369–79.

[2] Abrari Vajari Mohammad, Aghabayk Kayvan, Sadeghian Mohammad, Shiwakoti Nirajan. A multinomial logit model of motorcycle crash severity at Australian intersections. J Saf Res 2020;73:17–24.

[3] Li Zhibin, Liu Pan, Wang Wei, Xu Chengcheng. Using support vector machine models for crash injury severity analysis. Accid Anal Prev 2012;45:478–86.

[4] Abellán Joaquín, López Griselda, de Oña Juan. Analysis of traffic accident severity using Decision Rules via Decision Trees. Expert Syst Appl 2013;40(15):6047–54.

[5] Rezapour Mahdi, Mehrara Molan Amirarsalan, Ksaibati Khaled. Analyzing injury severity of motorcycle at-fault crashes using machine learning techniques, decision tree and logistic regression models. Int J Transp Sci Technol 2020;9(2):89–99.

[6] Hashmienejad Seyed Hessam-Allah, Hasheminejad Seyed Mohammad Hossein. Traffic accident severity prediction using a novel multi-objective genetic algorithm. Int J Crashworthiness 2017;22(4):425–40.

[7] Beshah Tibebe, Ejigu Dejene, Krömer Pavel, Sn'el V'clav, Plato Jan, Abraham Ajith. Learning the classification of traffic accident types. In: 2012 fourth international conference on intelligent networking and collaborative systems. 2012, p. 463–8.

[8] Kunt Mehmet, Aghayan Iman, Noii Nima. Prediction for traffic accident severity: Comparing the artificial neural network, genetic algorithm, combined genetic algorithm and pattern search methods. Transport 2011;26:353–66.

[9] Gu Xiaoning, Li Ting, Wang Yonghui, Zhang Liu, Wang Yitian, Yao Jinbao. Traffic fatalities prediction using support vector machine with hybrid particle swarm optimization. J Algorithms Comput Technol 2018;12(1):20–9.

[10] Li Kun, Xu Haocheng, Liu Xiao. Analysis and visualization of accidents severity based on LightGBM-TPE. Chaos Solitons Fractals 2022;157:111987.

[11] Chen Liang, Sun Jingjie, Li Kun, Li Qiaoru. Research on the effectiveness of monitoring mechanism for "yield to pedestrian" based on system dynamics. Physica A 2022;591:126804.

[12] Wang Junhua, Kong Yumeng, Fu Ting. Expressway crash risk prediction using back propagation neural network: A brief investigation on safety resilience. Accid Anal Prev 2019;124:180–92.

[13] Milletari Fausto, Navab Nassir, Ahmadi Seyed-Ahmad. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In: 2016 fourth international conference on 3D vision. IEEE; 2016, p. 565–71.

[14] Li Qing, Cai Weidong, Wang Xiaogang, Zhou Yun, Feng David Dagan, Chen Mei. Medical image classification with convolutional neural network. In: 2014 13th international conference on control automation robotics & vision. IEEE; 2014, p. 844–8.

[15] Bantupalli Kshitij, Xie Ying. American sign language recognition using deep learning and computer vision. In: 2018 IEEE international conference on big data. IEEE; 2018, p. 4896–9.

[16] Rahim Md Adilur, Hassan Hany M. A deep learning based traffic crash severity prediction framework. Accident Analysis and Prevention 2021;154:106090.

[17] Sharma Alok, Vans Edwin, Shigemizu Daichi, Boroevich Keith, Tsunoda Tatsuhiko. DeepInsight: A methodology to transform a non-image data to an image for convolution neural network architecture. Sci Rep 2019;9.

[18] Rawat Waseem, Wang Zenghui. Deep convolutional neural networks for image classification: A comprehensive review. Neural Comput 2017;29(9):2352–449.

[19] Kiranyaz Serkan, Avci Onur, Abdeljaber Osama, Ince Turker, Gabbouj Moncef, Inman Daniel J. 1D convolutional neural networks and applications: A survey. Mech Syst Signal Process 2021;151:107398.

[20] Gao Shengyao, Wang Xueren, Miao Xuhong, Su Changwei, Li Yibin. ASM1D-GAN: An intelligent fault diagnosis method based on assembled 1D convolutional neural network and generative adversarial networks. J Signal Process Syst 2019;91(10):1237–47.

[21] Ince Turker, Kiranyaz Serkan, Eren Levent, Askar Murat, Gabbouj Moncef. Real-time motor fault detection by 1-D convolutional neural networks. IEEE Trans Ind Electron 2016;63(11):7067–75.

[22] Kiranyaz Serkan, Avci Onur, Abdeljaber Osama, Ince Turker, Gabbouj Moncef, Inman Daniel J. 1D convolutional neural networks and applications: A survey. 2019.

[23] Lawrence Steve, Giles C Lee, Tsoi Ah Chung, Back Andrew D. Face recognition: A convolutional neural-network approach. IEEE Trans Neural Netw 1997;8(1):98–113.

[24] Tensmeyer Chris, Martinez Tony. Analysis of convolutional neural networks for document image classification. 2017.

[25] Liu Yunjie, Racah Evan, Correa Joaquin, Khosrowshahi Amir, Lavers David, Kunkel Kenneth, et al. Application of deep convolutional neural networks for detecting extreme weather in climate datasets. 2016, arXiv preprint arXiv:1605.01156.

[26] Laiou Alexandra, Papadimitriou Eleonora, Yannis George, Milotti Alberto. Road safety data and information availability and priorities in South-East European regions. Transp Res Procedia 2017;25:3703–14, World conference on transport research - WCTR 2016 Shanghai. 10-15 July 2016.

[27] Fiorentini Nicholas, Losa Massimo. Handling imbalanced data in road crash severity prediction by machine learning algorithms. Infrastructures 2020;5(7).

[28] Portal de Datos Abiertos del Ayuntamiento de Madrid. Accidentes de tráfico de Madrid. 2022, https://datos.madrid.es/portal/site/egob.

[29] Goyvaerts Jan, Levithan Steven. Regular expressions cookbook. Oreilly and associate series, O'Reilly Media, Incorporated; 2012.

[30] Data School. Comparing supervised learning algorithms. 2015.

[31] Dalwinder Singh, Birmohan Singh. Investigating the impact of data normalization on classification performance. Appl Soft Comput 2020;97:105524.

[32] Chawla Nitesh V, Bowyer Kevin W, Hall Lawrence O, Kegelmeyer Philip W. SMOTE: Synthetic minority over-sampling technique. J Artificial Intelligence Res 2002;16:321–57.

[33] Lingaraj Haldurai. A study on genetic algorithm and its applications. Int J Comput Sci Eng 2016;4:139–43.

[34] A. Ganaie Mudasir, Minghui Hu, Malik Shreshth A, Tanveer Maham, Suganthan Ponnuthurai N. Ensemble deep learning: A review. Eng Appl Artif Intell 2022;115:105151.

[35] Emmert-Streib Frank, Yang Zhen, Feng Han, Tripathi Shailesh, Dehmer Matthias. An introductory review of deep learning for prediction models with big data. Frontiers Artif Intell 2020;3.

[36] Li Zewen, Liu Fan, Yang Wenjie, Peng Shouheng, Zhou Jun. A survey of convolutional neural networks: Analysis, applications, and prospects. IEEE Trans Neural Netw Learn Syst 2021;1–21.

[37] James Gareth, Witten Daniela, Hastie Trevor, Tibshirani Robert. An introduction to statistical learning, vol. 112. Springer; 2013.

[38] Cortes Corinna, Vapnik Vladimir. Support-vector networks. Mach Learn 1995;20(3):273–97.

[39] Isaac Triguero, García-Gil Diego, Maillo Jesus, García Salvador, Herrera Francisco. Transforming Big Data into smart data: an insight on the use of the k-nearest neighbors algorithm to obtain quality data. Data Min Knowl Discov 2019;9(2):e1289.