A General Approach to Predict the Traffic Accidents Severity based on Deep Learning and Genetic Algorithms

Luis Pérez-Sala[a], Manuel Curado[a], José L. Oliver[b], Jose F. Vicent[a,*]

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

[b] *Department of Architecture, University of Alicante, Campus de San Vicente del Raspeig, Ap. Correos 99, E-03080, Alicante, Spain*

Abstract

According to the World Health Organization (WHO), thousands of people die in traffic accidents every year. A crucial problem is the prediction of the severity pf these accidents, which has been studied in many works that have to deal with issues such as unbalanced data. For this purpose, we propose a deep learning model to analyse the severity of accidents, the ultimate purpose of which is to distinguish whether an accident requires medical assistance or not. Our perspective is a general one such that the model is valid for any city. To this end, we present a model divided into three clearly differentiated stages. In the pre-processing step, a data treatment is conducted. Secondly, in the post-processing stage, genetic and boosting algorithms are employed to obtain the importance of all the dataset variables used in the prediction. Finally, we test the effectiveness and precision of the model by applying it to traffic accidents in six different cities. The experimental results show that the proposed model achieves better accuracy in all the cities compared to six state-of-the-art models. The results confirm the suitability and applicability of the proposed generalized deep learning model for detecting the severity of traffic accidents in real time.

Keywords: convolutional neural networks, genetic algorithm, data analysis, traffic accidents

1. Introduction

Artificial Intelligence (AI) has surpassed the realm of science fiction and is now seamlessly integrated into our daily lives. Its widespread applications in sectors such as healthcare, transportation, urban mobility, and sustainability have sparked a surge in research efforts under various perspectives, permeating numerous aspects of our existence. Although the AI has been under development for a decade, its practical implementation had to await advances in information technology. This necessitated progress in computer components, particularly fast processors, high-capacity memory and wireless networks. Thanks to these technological advances, artificial intelligence now encompasses a diverse range of techniques and resources that can be employed in various domains. Neural networks, AI planning, evolutionary algorithms, expert and knowledge systems, fuzzy logic, multi-agent systems, vector regression, data mining, and optimization techniques are just a few examples of the versatile tools that enable AI applications in diverse fields, such as urban mobility and road safety.

Daily, millions of road trips are made around the world, which has become an increasingly serious pollution problem, with solutions such as carpooling emerging to reduce this impact, such as in [1, 2]. The primary problem, however, is the excessive number of traffic accidents associated with people's mobility. There exists a large body of research on this topic, with studies focused on analysing the causes and severity

*Corresponding author.

Email addresses: lpg95@alu.ua.es (Luis Pérez-Sala), manuel.curado@ua.es (Manuel Curado), joseluis.oliver@ua.es (José L. Oliver), jvicent@ua.es (Jose F. Vicent)

of such accidents. In recent decades, the analysis of the severity of accidents has followed two different paths: statistical models and machine learning approaches [3, 4, 5]. In [6, 7, 8], the authors propose different models that are characterized by imposing specific assumptions on the data, presuming they adhere to a predetermined probability distribution. However, it is necessary to complete the input data to obtain error-free results.

In contrast, machine-learning models present similar or better performance than statistical models because they make any no assumptions about the input data. In the state of the art, several models have been applied to traffic accidents to evaluate certain problems, including the implementation of decision rules based on decision trees [9] that evaluate the significance of the characteristics, or the use of logistic regressions to classify their severity [10].

The classification of the severity of traffic accidents is key in the study of their prediction. In the literature, there are two main classifications: distinguishing between property damage only accidents and injury accidents [11], and between accidents involving slight injury and serious injury [12]. This decision is closely related to the level of detailed data available, which allows traffic accidents to be predicted under different approaches.

In light of these problems, this paper proposes to predict the severity of the traffic accidents, considering different situations, such as the temporality and the environmental conditions. The paper is thus structured as follows: in Section 2, we study the most recent related work on the prevention of traffic accidents. In Section 3, the methodology used is described in detail, while the experimental results of the proposed model are presented and discussed in Section 4. Finally, our conclusions are set out in Section 5.

2. Related Work

The task of predicting traffic accidents has been extensively studied in recent decades, with several methodologies being used to solve this problem. For instance, to classify traffic accidents efficiently, some approaches apply genetic algorithms, where knowledge about the drivers is highly important [13, 14].

In [15], the authors propose the training of fuzzy classifiers with evolutionary programming to find the important features and their relationships in the accident data. To increase the performance of the studies in the prediction of traffic accidents, it is common to combine different methodologies, such as genetic algorithms and pattern seeking with respect to artificial neural networks (ANN) and multilayer perceptrons [16].

These types of evolutionary methods are useful to optimize the input hyperparameters from external models to apply them to a specific problem, such as the severity of traffic accidents [17]. For instance, in [18] the authors propose the optimization of all the hyperparameters of the Support Vector Classifier (SVC) model using the Particle Swarm method.

Moreover, some works focus on seeking the most important characteristics to predict the severity of traffic accidents through data visualization or pattern analysis [19, 20]. For example, in [21], the authors use data visualization to detect patterns or variability in all the characteristics available in the accident datasets to interpret the causes of the accidents.

Another approach to solving traffic problems used in recent decades is game theory [22]. In this sense, in [23], the authors propose a dynamical model based on evolutionary game theory, studying the interactions between both traffic roles, pedestrians and drivers, using an evolutionary system.

Additionally, the machine learning methods allow classification problems to be solve, with this being very useful in the context of traffic accidents, such as neural networks for the prediction of accidents on highways [24, 25].

Focusing on deep learning models, the use of convolutional neural networks (CNN) solves problems in different areas, such as matrix segmentation, classification or language recognition, among others. For severity classification, in [26], the authors propose a traffic crash severity prediction framework using deep learning, where the method transforms crash data to images for predicting crash severity (fatal or injury crash only) using these learning approaches with the aim of helping to improve traffic safety.

CNNs have now become an extensively studied topic in several scientific areas, such as deep learning methods. These kinds of techniques require data input in a matrix form, implying the need to transform from categorical data to this format. Their uses focus on image recognition, computer vision, facial recognition, and speech recognition, among others [27, 28, 29, 30, 31, 32].

The original CNN is formed by a single-dimension [33]. The two-dimensional CNN then applies two-dimensional filters to the input to learn complex patterns, using them in tasks such as face recognition [34], classification of scanned documents [35] or the detection of dangerous weather events [36].

In the traffic context, several studies have used CNN methods for the prediction of road traffic accident severity [37, 38, 39, 40]. The main drawback of these previous studies is the low quality of the datasets [41]. Moreover, the difficulty of predicting the severity of the traffic accidents associated with the imbalance of data is analysed in [42]. However, in the case of traffic accidents, different solutions need to be studied to solve these problems, such as the use of re-sampling techniques or the definition of new classification metrics. In [43], a new model is presented to predict the severity of traffic accidents, using a set of characteristics such as environmental situations or driver gender, transforming this qualitative data into numerical matrices to feed evolutionary methods for hyperparameter optimization in two CNN architectures, both one- and two-dimensional. However, the accidents are classified into three categories (only property damages, moderate and fatal injuries), where data is imbalanced (there are few fatal accidents with respect to the remaining categories), and important characteristics such as temporality (day of the week and date) are not taken into account.

With all the above as the main motivation, the aim of this paper is to propose a new model to predict the need for medical assistance in traffic accident, taking into account a large set of characteristics from a general perspective, where the model can predict using a reduced number of general characteristics. It may be said that the ultimate goal is to propose a novel architecture that can infer the severity of traffic accidents in any city and in real time. To this end, a two-dimensional convolutional neural network with an optimized architecture is used.

The main contributions of the paper are as follows:

- We present a novel deep learning model to predict the severity of road accidents. It is essentially made up of three clearly differentiated processes. The first refers to the pre-processing of the data, and includes cleaning transformation, filtering of areas and resampling. The second stage refers to the post-processing and consists of the use of a genetic algorithm to optimize the hyperparameters used in the boosting model to calculate the weights of the variables. Finally, a novel optimized convolutional neural network 2D is applied to predict the severity of the accident.

- To test the efficiency of the presented model, it is applied to six cities using traffic accident datasets.

- We verify the effectiveness of the model presented by calculating accuracy metrics and comparing it with six well-known deep learning models.

3. Methodology

In [43], a model was proposed to predict the severity of road accidents were classified into three categories according to the severity of the injuries (minor, severe and fatal). Although the prediction results of the trained models are promising, there are certain weaknesses in the methodology proposed, caused mainly by the imbalance of the data. Thus, due to the nature of the data, there are few related to fatal or severe injury accidents compared to those involving slight injures. In addition, certain characteristics that could be important in the performance of the models, such as temporal and climatological features, are nor addressed.

In this context and in order to avoid these problems, we propose a new methodology. A flowchart of this methodology is presented in Figure 1 in which the three phases can be clearly observed: pre-processing, post-processing and model training. In this section, the different stages that make up this methodology are analysed. It is worth noting that the purpose is to present a generalized architecture that can infer the severity of traffic accidents by means of a two-dimensional convolutional model applicable to any city. This methodology focuses on predicting the severity of a road accident by distinguishing whether medical assistance is needed or not. It worth noting that the entire process -three steps– is carried out for each dataset.

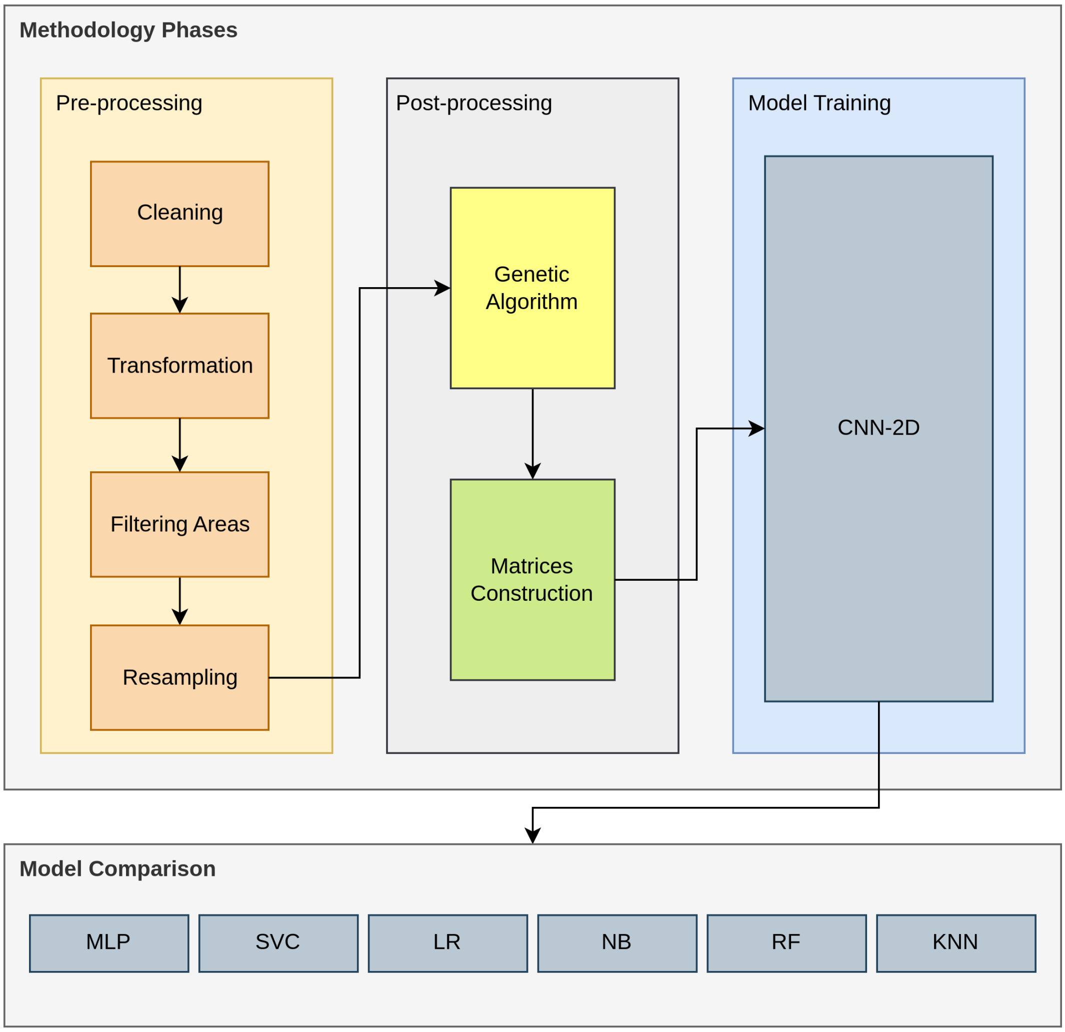


Figure 1: Methodology Phases.

Thus, in this section we are going to explain the three stages into which the model is divided.

3.1. Pre-processing Stage

This section explains the different sub-stages of the pre-processing phase of our proposed architecture: cleaning, transformation, filtering areas and resampling.

3.1.1. Cleaning

This is the first stage of the pre-processing in which the data undergoes through an initial cleaning process. This involves correcting the values of the dataset that are incorrectly formatted, in addition to eliminating from the dataset those records that contain null values, have outliers or are duplicated, such that the output of this phase is a uniform dataset that includes no inconsistent or incomplete information.

3.1.2. Transformation

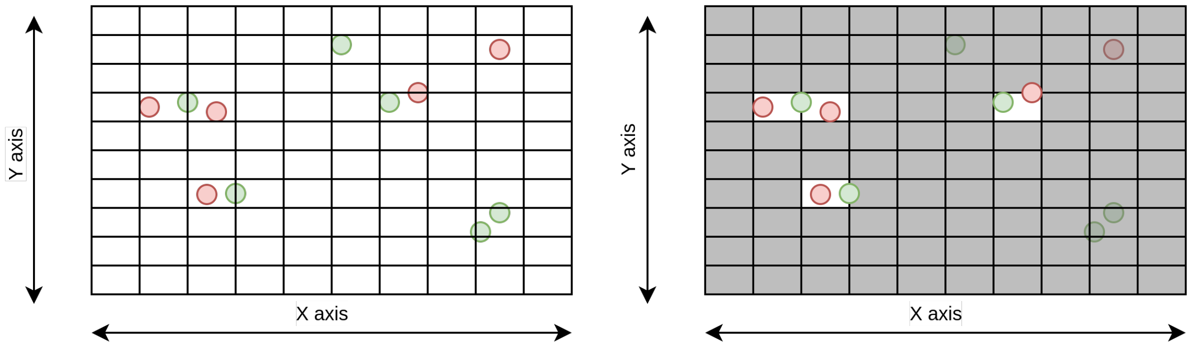
In this phase, the variables necessary for predicting the severity of accidents are selected. Thus, depending on the dataset available, there are potentially specific natural characteristics of each city that may be of interest. However, taking into account that the aim of this methodology is to be generalizable, it is proposed to select the variables that are easy to obtain and that do not involve a high data collection cost. For this reason, we select those characteristics that belong to one of the following five classifications:

1. **Location and Scale of Accident**. Features relating to the location of the accident, as well as data relating to the magnitude of the accident, such as its location or the number of vehicles involved in the accident.
2. **Driving limitations**. This encompasses the features that imply limitations to driving, either due to road conditions or by law, such as the state in which it is located or speed limits, among others.
3. **Temporal and environmental**. Characteristics related to the temporal and climatological context are included, such as the light conditions or the weather conditions at the time of the accident.
4. **Vehicle**. This includes those characteristics related to the accident vehicle, such as its weight or age.
5. **Victim**. This refers to characteristics related to the person involved in the accident, such as their age, or whether they are a pedestrian or a driver.

It should be pointed out that this categorization lends itself to many interpretations but theee are not decisive in the precision of the model because, in the second phase, that of post-processing, weights are assigned to each variable from each category. It is then possible to optimize the importance of the variables by assigning a weight using a boosting algorithm.

3.1.3. Filtering areas

One of the primary problems with road accident datasets is the imbalance of data with respect to the variable to be predicted. In our case, accident severity is imbalanced because of it is inherent to the nature of the problem. This entails a low ratio of fatal or serious accidents compared to minor accidents. Moreover, as seen in the related work section, some studies classify accidents into two or three categories, depending on the severity of the accident or the need for assistance. Therefore, with the aim of defining a generalizable model that can be applied to any type of city, the variable of accident severity is simplified into two classes, depending on whether medical assistance is needed or not. Additionally, a new fundamental stage of area filtering is introduced, which aims to reduce this imbalance by screening samples of minor accidents. In this new system, the surface of the cities is divided into rectangles, based on the coordinates of the accidents. In this way, only the areas in which both types of accident –with and without assistance– occurred are selected. If an area contains only minor accidents or attended accidents, this rectangle is discarded from the dataset (see Fig. 2). This means that all accidents in the discarded areas are removed from the database.



(a) Example of dataset accidents points.

(b) Example of filtered dataset accidents points.

Figure 2: Filtering areas example. Green points represent slight injury accidents while red points represent assistance accidents.

3.1.4. Resampling

Due to the nature of the data, even with area filtering applied, there are more occurrences of slight accidents than assistance accidents. An unbalanced dataset negatively affects the performance of the models, as they learn much more from the majority classes than from the minority classes. This results in the training of a biased model, which tends to infer new samples by classifying them as the majority class. To overcome this imbalance in the data, the original samples are initially split between training and test data. Subsequently, the Borderline SMOTE-II technique [44] is applied on the training data set, generating new synthetic samples of the minority class –accidents with assistance– until reaching the same number of records as the majority class –accidents without assistance–. This method generates new data by modifying the values of the minority class features that are close to the line of separability between the two classes. Once the last pre-processing step is applied, the dataset is ready to be normalized.

3.2. Post-processing Stage

Starting from a balanced and normalized dataset, this section explains the new methodology that operates on the data. The goal of this stage is to transform the data so that the 2D convolutional neural network can interpret them. Firstly, the process that determines the importance of each dataset characteristic is detailed. For this, a boosting algorithm is used whose parameters are optimized by a genetic algorithm. Subsequently, the process of matrix construction that serves as input to the convolutional neural network is applied.

3.2.1. Genetic Algorithm

To obtain the weights of each variable, we use a boosting algorithm. These algorithms have a series of hyperparameters that need to be optimized for their correct operation. It is in this phase that a genetic algorithm is used to optimize the input hyperparameters of the boosting model.

Genetic algorithms are methods used to search for approximate solutions to optimization problems. These algorithms work with heuristic functions, which are simpler approximations of the original objective to be minimised. This makes it possible to greatly reduce the resources needed for optimization thanks to their behaviour based on biological evolution, in which the individuals of a population represent possible solutions to the problem and, by simulating the evolution of the individuals over generations, increasingly optimized solutions to the problem are obtained. A genetic algorithm is composed of three fundamental stages that may vary depending on the implementation itself:

1. Selection: this is the stage of selecting individuals in the population at the beginning of each iteration of the algorithm. Each individual is evaluated on the basis of the heuristic function and assigned a score. The individuals with the best score are selected to reproduce and survive in the next population.

2. Crossover: this is the stage of reproduction. The best individuals mix their information with each other to give rise to new individuals that may contain the best information from both parents. Crossover strategies can be varied, such as each new individual being composed of half of the information from each of its parents, or a mixing ratio can be established.

3. Mutation: as in any biological population, diversity is the key to evolution. A population that shares a large amount of information between one produces stagnation in the successive individuals. This is why a mutation stage is introduced in which the values of the new individuals are randomly altered based on a probability, thus ensuring the diversity of the population and avoiding the stagnation that could occur. This stagnation is referred to as *local minima*.

By sequentially repeating these three stages, a cycle is generated in which the population tends to improve in each generation. The new individuals improve the score of their progenitors and after a defined number of iterations, the individual with the best score is selected, arriving at an optimized solution.

Genetic algorithms need to establish a number of parameters in order to run, such as the fitness function by which the quality of the individuals are evaluated, the size of the population, the number of individuals that reproduce for the next generation, the rate at which the parents divide to give rise to a new individual, the probability with which a new individual mutates and the range of values allowed for each variable of this new individual. These parameters are set to obtain a quality solution representing the hyperparameters of the boosting model.

Once the hyperparameters of the classification algorithm have been optimized using the genetic algorithm, they are used to calculate the weights of the variables used in the dataset by means of the deep learning boosting algorithm. The input of this model are the optimized hyperparameters and the resampling data of the traffic accidents. The output are the weight values of these variables, which typify the degree of importance of each feature of the dataset.

3.2.2. Matrix Construction

Convolutional neural networks interpret input data in the form of matrices. These networks are widely used for image classification, such that in order to apply a convolutional network to a tabular dataset it is necessary to apply transformation that converts this data, originally in a row and column format, to a matrix. This conversion requires the application of an algorithm that positions the values of each of the accident variables in strategic positions within the matrix, since the position in which each feature is assigned influences the final performance of the convolutional neural network. This is because these networks are trained to find patterns within the matrices, and, if the position of the features that influence the categorization of the accident are not in the relevant positions within the matrix, the training of the network can be affected by information losses, negatively affecting its final performance.

To solve this problem, and with the aim of feeding the convolutional neural network, it is proposed to calculate the weight of the variables, in order for those that are most important to describe the dataset to have more privileged positions in the matrices. To find the position of each feature, the weight of each one is calculated by means of a boosting algorithm based on decision trees. The hyperparameters of this algorithm are optimized during successive generations of a genetic algorithm, where each individual in the population represents a combination of these hyperparameters. These optimized values of the hyperparameters are used to train the final algorithm with the aim of calculating the weight of the features, representing the importance of each one of them has had during its training.

Based on the weight of each variable, the total weight of each category is calculated by adding up each of their characteristics. As a result of this process, the order of importance of the categories is obtained, as shown in Figure 3.

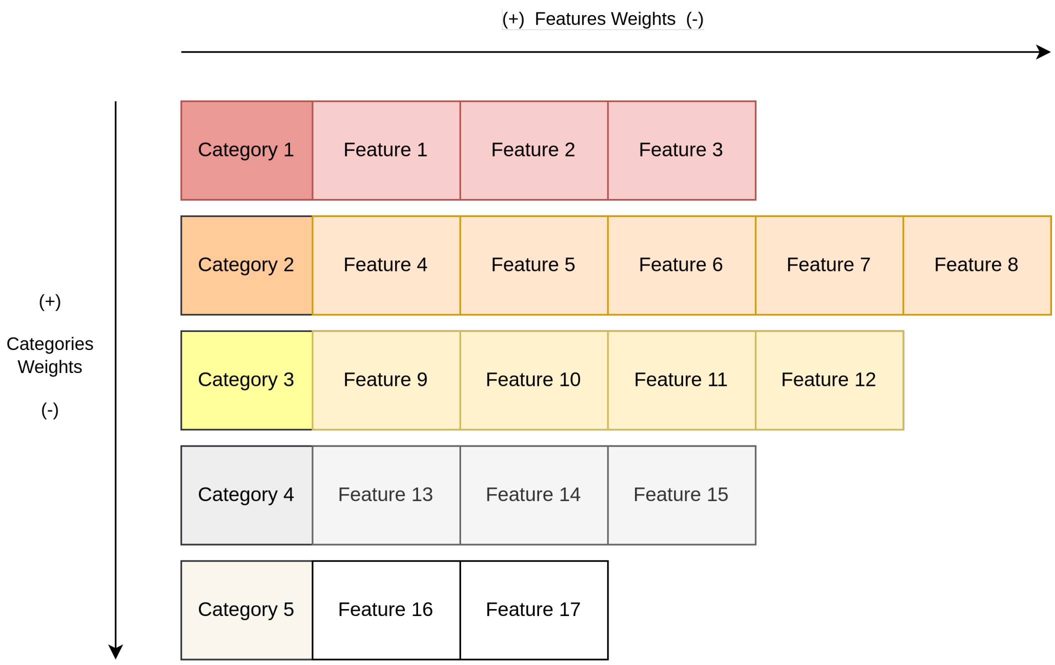


Figure 3: Category and feature weights.

Finally, the following algorithm is applied to position the individual features in the coordinates of the matrix:

1. The weight of each category is calculated, taking into account that this weight is the sum of each of its individual characteristics.
2. Each category is associated with a row of the matrix according to its weight, where that with the greatest weight is positioned in the middle row, the second most important characteristic is associated to the row immediately above, the next one to the row immediately below and so on (see Fig. 5).
3. Each of the characteristics within its category is associated in each column of the row following the same procedure as above. The most important feature of a category is positioned in the centre of the category, the second most important feature is positioned immediately to its left while the next most important feature occupies the place to its left and so on (see Fig. 5).

The result of this process is a transformation of initially tabular data into an $n \times n$ matrix, where the variables most important to the data are in the central positions, as shown in Figure 5.

Category 4	0	Feature 14	Feature 13	Feature 15	0
Category 2	Feature 7	Feature 5	Feature 4	Feature 6	Feature 8
Category 1	0	Feature 2	Feature 1	Feature 3	0
Category 3	Feature 12	Feature 10	Feature 9	Feature 11	0
Category 5	0	Feature 17	Feature 16	0	0

Figure 4: Categories and feature positions.

3.3. The convolutional neural network for prediction of traffic accidents severity

The new model proposed presents an architecture of four convolutional layers of two dimensions each, with a kernel size of $3x3$ and a ReLU activation function. It is worth noting that a batch normalization is applied to the output of each convolutional layer.

The first convolutional layer of the network consists of 64 kernels, the second of 512, the third of 128 and the fourth of 256. These kernels contain the weights that are trained during the model fitting phase from the known output of the labelled data, learning which multiplications on the data are those that minimize the defined loss function of the network (binary cross entropy) thanks to back propagation process. The output of each convolutional layer are the feature maps, which are the result of applying the multiplication of these filters on their input. The stride, or number of units that advance the kernels for a feature map, is 1. Padding is also applied in the convolutions, that is, if the kernel multiplication exceeds the limits of the matrix, zeros will be added to these limits in order to carry out the convolution. The resulting feature maps of the last layer go through a flattening layer, which transforms the data to a single dimension once the convolutions are finished. Each of these flattened data is interconnected with the 256 defined nodes of the dense layer (Fully Connected Network). Finally, the dense layer is connected to a final dense layer with Sotfmax activation function, which gives the probability of each new sample belonging to one of the two classes.

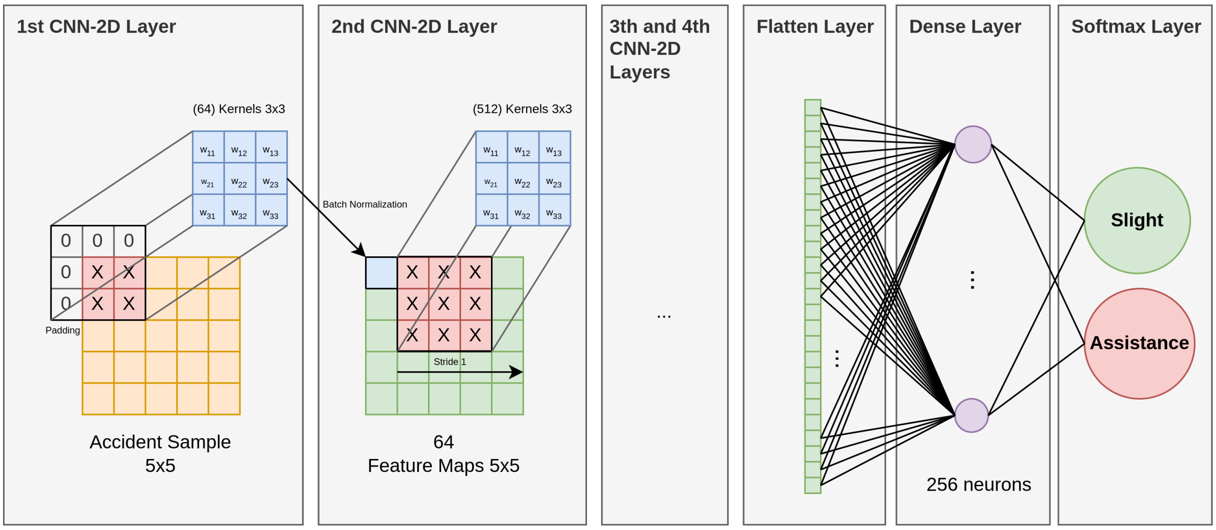


Figure 5: Proposed CNN-2D architecture summary.

Figure 5 shows a schematic of the operation of the new convolutional network proposed. The hyperparameters of each layer were calculated based on a hyperparameter optimization procedure using GridSearchCV, where different combinations of hyperparameters of the network are tested over a defined range.

3.4. Performance evaluation and model comparison

This section identifies six deep learning models against which the new two-dimensional convolutional neural network model is compared. Similarly, the results obtained for each of the dataset and the metrics used for comparison are defined.

3.4.1. Performance evaluation

To validate the effectiveness of the proposed 2D convolutional neural network model, we perform a comparison with different deep learning models. In order to assess the performance of the models, the following commonly known metrics are used:

Metrics	Description
Precision	It is the ratio between the number of true positive classified to the total number of positive (true or false). This metric measures the accuracy in classifying a sample as positive [45]: $$\text{Precision} = \frac{TP}{TP + FP}.$$
Recall	It is the ratio between the number of true positive classified as positive to the total number of positive. This metric measures the ability to detect positive samples [45]: $$\text{Recall} = \frac{TP}{TP + FN}.$$
F1-Score	It is a measure of a model's accuracy that is used to evaluate binary classification systems by combining Precision and Recall. It is defined as the harmonic mean of the model's precision and recall [45]: $$\text{F1-Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}.$$
Binary Cross Entropy	It is the loss function on which the neural networks are optimized when the number of classes to be predicted is 2. This metric is defined as the negative average of the log of corrected predicted probabilities [46]: $$\text{Binary Cross Entropy} = \frac{-1}{N}\sum_{i=1}^{N}(y_i * \log p_i + (1 - y_i) * \log(1 - p_i)).$$

Table 1: Metrics used in the comparison.

It should be underlined that the metric that will be used to measure the effectiveness of the models in the experimental phase is the F1-Score, since it combines both precision and recall in such a way that it is possible to evaluate the quality of a model based on a single metric.

3.4.2. Model comparison

To validate the effectiveness of the proposed 2D convolutional neural network model, we performed a comparison with different deep learning models. We are aware there are many models to compare with, but those selected were chosen for their popularity and use in deep learning. These models are listed in Table 2.

Model	Description
Naive Bayes (NB)	It is a supervised learning algorithm based on Bayes theorem. It is used for solving classification problems with high-dimensional training dataset [47]
Support-Vector Classifier (SVC)	Actually, it is a set of supervised learning methods that can be used for classification or regression. It is effective in high dimensional spaces and its objective is to fit to the data, returning a "best fit" hyperplane that divides, or categorizes, the data [48].
K-Nearest Neighbors (KNN)	It is a supervised learning technique that assumes the similarity between the new and available data. This means that, when new data appears, it can be easily classified. This algorithm can be used for regression and classification and it is a non-parametric algorithm [49].
Random Forest(RF)	It is a supervised learning algorithm that builds the "forest" using an ensemble of decision trees. It is a flexible, easy-to-use machine learning algorithm that can be used for both classification and regression tasks [50].
Logistic Regression (LR)	It is a statistical analysis method that aim to predict a binary outcome. It predicts a dependent data variable by analyzing the relationship between one or more existing independent variables [51].
Multilayer Perceptron (MLP)	It is one of the most common neural network models used. It consists of interconnected neurons transferring information to each other. The network can be divided into three main layers: input , hidden and output [52].

Table 2: Definition of the models used in the comparison.

Regarding the six models proposed for comparison, an interesting feature is the inference time for new accident predictions. The Naive Bayes model is the fastest in terms of inference, followed by Random Forest, Logistic Regression and Multilayer Perceptron. In the last positions are the Support-Vector Classifier and K-Nearest Neighbours. However, it should be noted that our model is designed to be executed in real time, and so the execution time is relatively small.

4. Experimental results

This section presents all the information regarding the chosen datasets, the results of the proposed methodology on these datasets, as well as a comparison of the proposed convolutional network with six models from the state of the art.

4.1. Experimental setup

The application of the proposed methodology based on the chosen dataset is described in this subsection. For this purpose, we first explain, the nature of the data, the categorization of its values and the selection of the variables chosen to predict accident severity. Subsequently, the calculation of variable weights is applied to construct the matrices that from the input of the proposed new convolutional model.

4.1.1. Dataset

The chosen dataset is from the UK Government's Department for Transport dataset [53]. This dataset describes accidents that occurred across a set of cities and districts in the UK, where for each of the accidents information such as the date on which it occurred, the speed limit of the road, the number of vehicles involved, weather conditions at that time, among many others, is included. We chose accidents between 2005-2019 across cities of different sizes, namely Southwark, Manchester, Birmingham, Liverpool, Sheffield and Cornwall. This gave us a sufficiently representative sample of $345,229$ accidents with a total of 81 characteristics.

4.1.2. Feature classification

As has discussed in Transformation Section 3.1.2, in order to build a procedure applicable to any type of city, the five categories proposed for the convolutional network architecture encompass variables that are easy to acquire and easy to access. This fits the data sets for the six UK cities listed in the previous section. Therefore, in this process of classifying variables, the characteristics that cannot be classified in any of the five proposed categories for the reasons specified above are discarded from the original data set.

Once this first filtering has been applied, a correlation analysis is carried out between variables to check which characteristics are unnecessary, since the inclusion of pairs of highly correlated variables presents redundant information and negatively affects the training of the models. For this purpose, a threshold of ± 0.5 is established, after which two characteristics are considered to be highly related and those that exceed this threshold are eliminated from the dataset. Finally, 17 characteristics were obtained, with the highest correlation being road surface-weather conditions, with a score of 0.47. Table 3 shows the quantification of the resulting features included in the proposed categories.

Classification	Feature	Typing	Value
Location & Scale Accident	Easting	Real Number	OSGR East Coordinate
	Northing	Real Number	OSGR North Coordinate
	Time	0	Between 06:00 and 18:00
		1	Between 18:00 and 06:00
	1st Road Class	0	Motorway
		1	A(M)
		2	A
		3	B
		4	C
		5	Unclassified
	Number of Vehicles	0-N	Depending on the number of vehicles involved
Driving Limitations	Surface	0	Dry
		1	Wet / Damp
		2	Snow
		3	Frost / Ice
		4	Flood
	Speed Limit	0-70	Depending on the speed limit (mph) of the road
Temporary & Environmental	Lighting Conditions	0	Daylight: street lights present
		1	Darkness: no street lighting
		2	Darkness: street lights present and lit
		3	Darkness: street lights present but unlit
		4	Darkness: street lighting unknown
	Weather Conditions	0	Fine without high winds
		1	Raining without high winds
		2	Snowing without high winds
		3	Fine with high winds
		4	Raining with high winds
		5	Snowing with high winds
		6	Fog or mist
		7	Other
	Day on Week	0-6	Depending of the day on week
	Week on Year	0-51	Depending of the year on week
Vehicle	Vehicle Type	0-17	Depending on the weight of the vehicle
	Vehicle Age	0-N	In order of vehicle age
	Point of Impact	0	Did not impact
		1	Front
		2	Back
		3	Offside
		4	Nearside
		5	Unknown (self reported)
Victim	Casualty Class	0	Driver/Rider
		1	Passenger
		2	Pedestrian
	Casualty Sex	0	Male
		1	Female
	Casualty Age	0	Younger than 18
		1	Between 18 and 25
		2	Between 25 and 65
		3	Older than 65

Table 3: Classification of variables.

4.1.3. Filtering and oversampling cities

In this section, the slight and assistance accident samples are filtered for each of the six cities chosen by means of the subdivision by filtering areas. Table 4 shows, for each coordinate axis, the size and the resulting total number of areas for each city.

Areas Split (OSGR Units)			
City	**Axis**	**Areas Number**	**Areas Size**
Southwark	X	529	10
	Y	487	20
Manchester	X	791	14
	Y	1069	20
Birmingham	X	3519	12
	Y	1557	17
Liverpool	X	2107	12
	Y	717	21
Sheffield	X	1896	12
	Y	1115	18
Cornwall	X	7777	14
	Y	5242	20

Table 4: Data distribution for all cities.

Note that the size of the areas was defined by means of a bisection procedure, where the sizes for both axes were delimited on the basis of the results obtained from the average of the models.

Table 5 shows the number of accidents of each class by city, for each of the implementation phases of the proposed methodology. The column *Original* shows the number of raw records, i.e. the number of accidents in the original dataset. The *Filtered* column shows the data after filtering by area, so that the original sample set is reduced to a subset in which only minor accidents and those requiring assistance coexist, resulting in a dataset that is notably less unbalanced than the original. The last column *Oversampled* shows the data resulting from the application of the Borderline SMOTE-II algorithm on the training set of data, in order to fit the models with as many balanced samples as possible.

Data Distribution				
City	**Severity**	Original	Filtered	Oversampled
Southwark	Slight	27105	4251	2973
	Assistance	3109	1256	2973
Manchester	Slight	48771	4548	3178
	Assistance	4570	1466	3178
Birmingham	Slight	108723	4092	2838
	Assistance	11187	2063	2838
Liverpool	Slight	49291	3640	2554
	Assistance	5161	1192	2554
Sheffield	Slight	43579	2060	1447
	Assistance	5887	1638	1446
Cornwall	Slight	32994	3171	2218
	Assistance	4852	2845	2217

Table 5: Data distribution for the selected cities.

4.1.4. Genetic algorithm

Once the final data is available, the search for hyper-parameters of the boosting model is applied by running a genetic algorithm. The hyperparameters to be optimized for the boosting algorithm are the learning rate of the algorithm (ETA), the maximum depth of the tree (Max Depth), and the sum of the minimum child weight (Min Child Weight). The fitness function to be optimized by the genetic algorithm is the F1-Score resulting from the execution of the validation of the boosting model on the accident test set, i.e. on the accidents not seen during training. The parameters chosen, in the genetic algorithm, to optimize the fitness function can be seen in Table 6.

Hyperparameter	**Value**
Population	50
Parents Mating	10
Generations	50
Crossover Index	Random
Mutation Probability	0.4
Fitness Function	Boosting Algorithm F1-Score

Table 6: Genetic algorithm hyperparameters setup.

Ii is also worth noting that the hyperparameters values of the genetic algorithm are selected on the basis of optimizing the convergence of the heuristic function as efficiently as possible through numerous experiments and validations.

Table 7 shows the maximum and minimum values allowed for each variable to be optimized, both the limits in the initialization of new individuals and the limits when a mutation occurs.

Hyperparameter	**Limit**	**Min**	**Max**
ETA	Initial	0.01	1
	Mutation	-0.2	0.2
Max Depth	Initial	1	25
	Mutation	-3	3
Min Child Weight	Initial	0.01	20
	Mutation	-4	4

Table 7: Boosting models hyperparameters limits.

Finally, Table 8 shows the hyperparameters resulting from each boosting model for each city after running the genetic algorithm.

Genetic Algorithm Result Values			
City	**ETA**	**Maximum Depth**	**Minimum Children Weight**
Southwark	0.84	10	1.51
Manchester	0.01	1	0.01
Birmingham	0.32	1	0.01
Liverpool	0.38	1	0.01
Sheffield	0.96	1	17.9
Cornwall	0.59	1	0.01

Table 8: Resulting boosting model hyperparameters after executing the genetic algorithm.

4.2. Model validation

To evaluate the presented model, a test set was used, which was previously separated from the augmented training set, so that it can be considered a sufficiently representative data set to validate its quality, since they are data that not seen during their learning process. Table 5 shows the number of slight and assistance accident samples for each of the areas with which the models were validated.

Test Samples Number			
City	**Accident Type**	**Amount**	**Total**
Southwark	Slight Assistance	1267 385	1652
Manchester	Slight Assistance	1341 464	1805
Birmingham	Slight Assistance	1235 612	1847
Liverpool	Slight Assistance	1080 370	1450
Sheffield	Slight Assistance	601 473	1074
Cornwall	Slight Assistance	950 855	1805

Table 9: Test set distribution for all cities.

4.3. Model comparisons

This section shows a comparison of the proposed CNN-2D model with the six state-of-the-art models (Naive Bayes, Support-Vector Classifier, K-Nearest Neighbours, Random Forest, Logistic Regression and Multilayer Perceptron). The comparison is performed for each of the six city dataset, namely Southwark, Manchester, Birmingham, Liverpool, Sheffield and Cornwall. The resulting metrics obtained are shown for both the training data, with the aim of understanding how the models are fitted to the data, and for the validation set, to show how they generalise based on test set. The quality of the CNN-2D model is also compared with that of the faster model in terms of inference.

4.3.1. Southwark

		Train			Test		
Model	**Severity**	Precision	Recall	F1	Precision	Recall	F1
NB	Slight	0.57	0.77	0.655	0.841	0.77	0.803
	Assistance	0.645	0.419	0.508	0.407	0.519	0.456
SVC	Slight	0.909	0.82	0.862	0.925	0.773	0.843
	Assistance	0.836	0.917	0.875	0.516	0.795	0.626
KNN	Slight	0.748	0.559	0.64	0.898	0.558	0.688
	Assistance	0.648	0.812	0.721	0.353	0.792	0.488
RF	Slight	0.767	0.444	0.563	0.889	0.437	0.586
	Assistance	0.609	0.865	0.715	0.307	0.821	0.447
LR	Slight	0.603	0.633	0.618	0.846	0.631	0.723
	Assistance	0.614	0.584	0.599	0.339	0.623	0.439
MLP	Slight	0.966	0.951	0.959	0.936	0.882	0.908
	Assistance	0.952	0.967	0.959	0.674	0.8	0.732
CNN2D	**Slight**	0.996	0.995	0.996	0.972	0.95	**0.961**
	Assistance	0.995	0.996	0.996	0.847	0.909	**0.877**

Table 10: Southwark models metrics.

Analysing the results of the set of tests in Table 10, it can be seen that the best results for both accident categories are presented by the CNN-2D model, obtaining a considerably significant improvement over the next best model (MLP) in the assistance accident category, with an increase of 14.5% in terms of F1-Score, while the improvement in slight accidents over the next best model, again the MLP, is 5.3%.

Comparing the quality of the CNN-2D model with respect to the fastest model in terms of inference (Naive-Bayes), we obtain an increase in the F1-Score of 15.8% for slight accidents and 42.1% for assistance accidents.

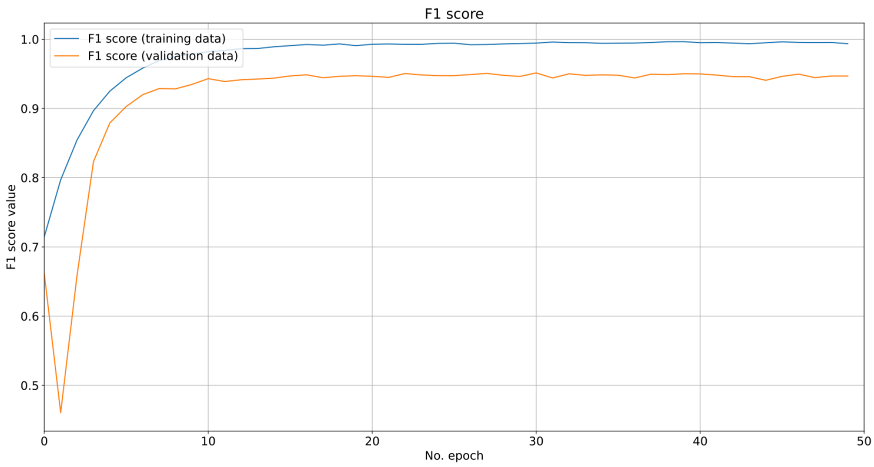


Figure 6: Evolution of the F1-score of the CNN-2D in training and test set for Southwark.

An analysis of the training graph in Figure 6 shows the evolution of the F1-Score metric over the epochs of the network training. The maximum value over the validation set is reached around epoch 30, while the network does not stop learning from the training set, showing its maximum value at epoch 37. Hence, this number of epochs is sufficiently representative to choose the model trained with these parameters. If the network were to continue training, it could fall into a situation of overfitting, leaving a model that knows the training data by heart.

4.3.2. Manchester

		Train			Test		
Model	**Severity**	Precision	Recall	F1	Precision	Recall	F1
NB	Slight	0.626	0.801	0.702	0.861	0.803	0.831
	Assistance	0.723	0.521	0.605	0.523	0.625	0.57
SVC	Slight	0.89	0.839	0.864	0.906	0.809	0.855
	Assistance	0.848	0.896	0.872	0.579	0.759	0.657
KNN	Slight	0.783	0.676	0.726	0.896	0.671	0.768
	Assistance	0.715	0.813	0.761	0.449	0.776	0.569
RF	Slight	0.705	0.814	0.755	0.899	0.814	0.854
	Assistance	0.78	0.66	0.715	0.578	0.735	0.647
LR	Slight	0.682	0.767	0.722	0.881	0.77	0.822
	Assistance	0.734	0.643	0.685	0.513	0.7	0.593
MLP	Slight	0.955	0.904	0.929	0.886	0.854	0.869
	Assistance	0.909	0.957	0.932	0.617	0.681	0.648
CNN2D	**Slight**	0.997	0.988	0.993	0.899	0.908	**0.904**
	Assistance	0.988	0.998	0.993	0.727	0.705	**0.716**

Table 11: Manchester model metrics.

Comparing the results of the new CNN-2D model on the basis of Table 11, we obtain an increase in performance, relative to the next best models for each of these classes, for slight accidents of 3.5% over the MLP model, and an increase of 5.9% over the assistance accidents of the SVC model. On the other hand, comparing the fastest model (Naive-Bayes) with the CNN-2D, an improvement of 7.3% was obtained over slight accidents and 14.6% over assistance accidents.

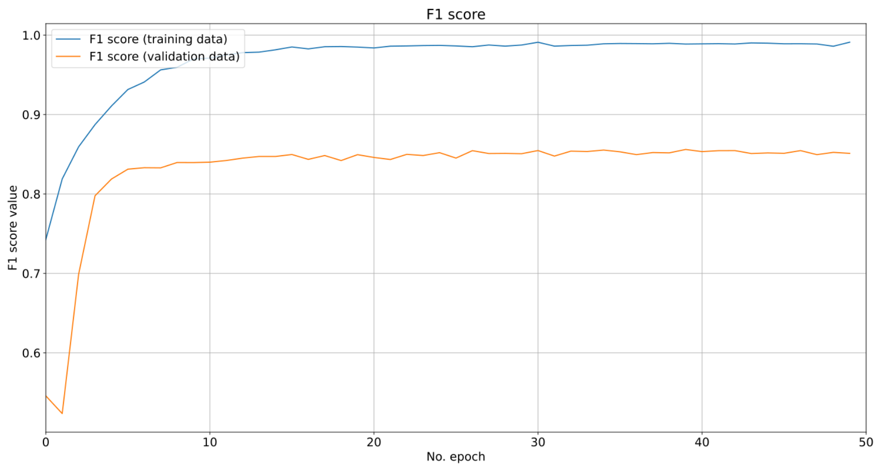


Figure 7: Evolution of the F1-score of the CNN-2D in training and test set for Manchester.

Looking at the results of the evolution of the training metrics shown in Figure 7, the maximum value of F1-Score of the validation set is observed around epoch 39 and subsequently remains constant, while, over the training set, this metric increasing constantly. If we were to continue training the network over the epochs, an overfitting situation would occur, where the model would not be able to generalize over new data.

4.3.3. Birmingham

		Train			**Test**		
Model	**Severity**	Precision	Recall	F1	Precision	Recall	F1
NB	Slight	0.629	0.796	0.703	0.82	0.805	0.812
	Assistance	0.722	0.53	0.611	0.62	0.644	0.632
SVC	Slight	0.825	0.818	0.821	0.838	0.778	0.807
	Assistance	0.819	0.826	0.823	0.609	0.696	0.649
KNN	Slight	0.725	0.653	0.687	0.815	0.613	0.7
	Assistance	0.684	0.752	0.717	0.479	0.719	0.575
RF	Slight	0.676	0.823	0.743	0.826	0.046	0.087
	Assistance	0.774	0.606	0.68	0.337	0.98	0.502
LR	Slight	0.659	0.712	0.684	0.828	0.704	0.761
	Assistance	0.686	0.631	0.658	0.541	0.704	0.612
MLP	Slight	0.915	0.881	0.897	0.828	0.783	0.805
	Assistance	0.899	0.899	0.899	0.605	0.672	0.637
CNN2D	**Slight**	0.993	0.978	0.985	0.837	0.851	**0.844**
	Assistance	0.978	0.993	0.985	0.689	0.667	**0.678**

Table 12: Birmingham model metrics.

Considering the results for the test set shown in Table 12, the best generalization metrics for both types of accidents in terms of F1-Score are again presented by CNN-2D. In terms of incremental improvement over the next best model in slight accidents, the Naive Bayes, a difference of 3.2% is obtained over this class.

Regarding the quality of the prediction of assistance accidents, there is an improvement of 2.9% over the second best model in this class, the SVC.

Comparing the CNN-2D model with the Naive-Bayes model, which is the fastest, a difference of 3.2% is obtained for slight accidents, while for assistance accidents, it is 4.6%.

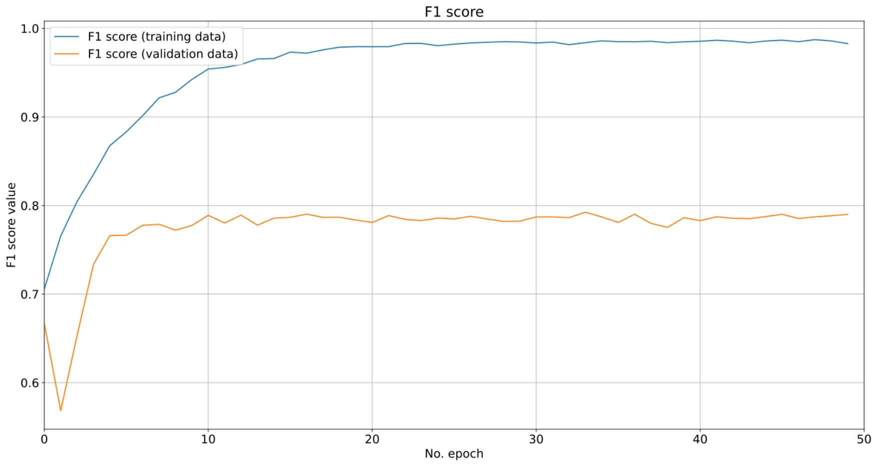


Figure 8: Evolution of the F1-score of the CNN-2D in training and test set for Birmingham.

The training plot in Figure 8 shows that the performance of the network on the training set, the data it trains on, reaches its maximum value at epoch 46, while the maximum value on the validation set, data that the network does not see during the adjustment, occurs at epoch 33; at later epochs it remains constant. This means that the network is unable to generalize further from epoch 33 onwards.

4.3.4. Liverpool

		Train			**Test**		
Model	**Severity**	Precision	Recall	F1	Precision	Recall	F1
NB	Slight	0.645	0.792	0.711	0.858	0.776	0.815
	Assistance	0.731	0.565	0.637	0.488	0.624	0.548
SVC	Slight	0.88	0.873	0.876	0.888	0.827	0.856
	Assistance	0.874	0.881	0.877	0.579	0.695	0.631
KNN	Slight	0.783	0.627	0.696	0.903	0.627	0.74
	Assistance	0.689	0.827	0.752	0.424	0.803	0.555
RF	Slight	0.714	0.757	0.735	0.886	0.735	0.804
	Assistance	0.741	0.696	0.718	0.484	0.724	0.58
LR	Slight	0.682	0.769	0.72	0.873	0.754	0.809
	Assistance	0.735	0.641	0.685	0.486	0.681	0.568
MLP	Slight	0.948	0.919	0.933	0.875	0.832	0.853
	Assistance	0.921	0.95	0.935	0.571	0.651	0.609
CNN2D	**Slight**	0.999	0.984	0.991	0.88	0.874	**0.877**
	Assistance	0.984	0.999	0.991	0.639	0.651	**0.645**

Table 13: Liverpool model metrics.

Analysing the results in Table 13, we can observe an improvement in the CNN-2D model over the next best models in the slight and assistance categories by the next best model, the SVC in both cases, of 2.6% and 1.4% respectively. The CNN-2D model obtains a performance increase over the fastest Naive-Bayes model on slight and assistance accidents of 6.2% and 9.7% respectively.

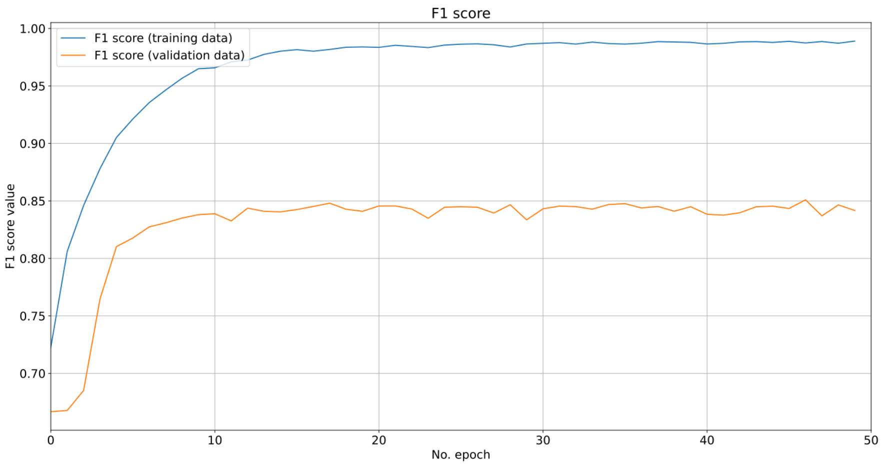


Figure 9: Evolution of the F1-score of the CNN-2D in training and test set for Liverpool.

Fig. 9 shows the training and validation data during model fitting across epochs. The highest F1-Score on the training set is obtained at epoch 50, while on the validation set on 46, from that epoch, the model stops generalizing about the validation set.

4.3.5. Sheffield

		Train			**Test**		
Model	**Severity**	Precision	Recall	F1	Precision	Recall	F1
NB	Slight	0.611	0.764	0.679	0.699	0.782	0.738
	Assistance	0.685	0.514	0.588	0.674	0.573	0.619
SVC	Slight	0.843	0.809	0.826	0.805	0.769	0.786
	Assistance	0.816	0.85	0.833	0.722	0.763	0.742
KNN	Slight	0.682	0.704	0.693	0.742	0.719	0.73
	Assistance	0.694	0.672	0.683	0.657	0.683	0.669
RF	Slight	0.626	0.838	0.717	0.698	0.859	0.77
	Assistance	0.756	0.5	0.602	0.746	0.529	0.619
LR	Slight	0.659	0.693	0.675	0.747	0.714	0.73
	Assistance	0.676	0.641	0.658	0.656	0.693	0.674
MLP	Slight	0.88	0.874	0.877	0.798	0.802	0.8
	Assistance	0.875	0.881	0.878	0.747	0.742	0.744
CNN2D	**Slight**	0.973	0.997	0.985	0.794	0.857	**0.824**
	Assistance	0.997	0.972	0.985	0.798	0.717	**0.755**

Table 14: Sheffield model metrics.

Based on the results presented in Table 14, the CNN-2D model again presents the best metrics for slight and assistance accidents over the next best model for both cases, the MLP, with a difference of 2.4% and 1.1% respectively. The performance increase over the fastest inference model (Naive-Bayes) is 8.6% and 13.6% for the slight and assistance accidents respectively.

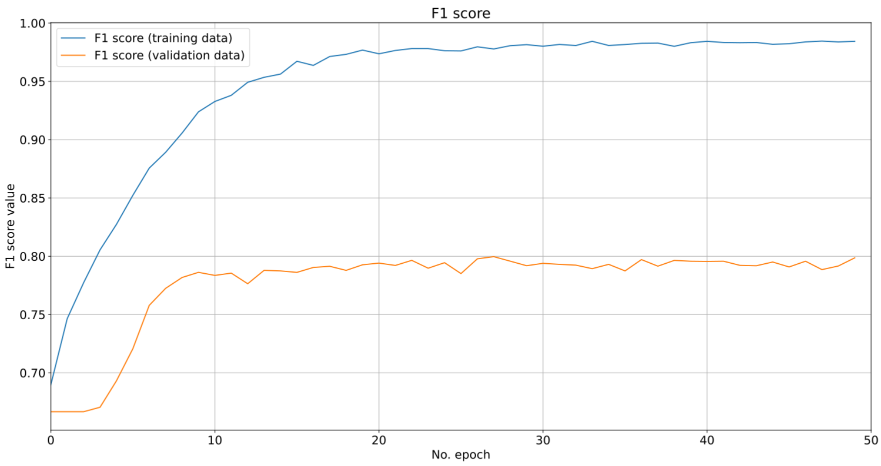


Figure 10: Evolution of the F1-score of the CNN-2D in training and test set for Sheffield.

Figure 9 shows the training and validation data for this city during model fitting across epochs. The highest F1-Score over the training set is obtained at epoch 50, while over the validation set at epoch 28, from that epoch onwards, the model stops generalizing over the validation set.

4.3.6. Cornwall

		Train			Test		
Model	**Severity**	Precision	Recall	F1	Precision	Recall	F1
NB	Slight	0.576	0.797	0.669	0.622	0.782	0.693
	Assistance	0.671	0.413	0.511	0.661	0.473	0.551
SVC	Slight	0.721	0.824	0.769	0.695	0.769	0.73
	Assistance	0.794	0.682	0.734	0.709	0.625	0.664
KNN	Slight	0.626	0.752	0.683	0.656	0.729	0.691
	Assistance	0.689	0.551	0.612	0.656	0.574	0.613
RF	Slight	0.623	0.862	0.723	0.676	0.299	0.415
	Assistance	0.776	0.478	0.591	0.519	0.841	0.642
LR	Slight	0.643	0.675	0.658	0.677	0.675	0.676
	Assistance	0.658	0.625	0.641	0.64	0.642	0.641
MLP	Slight	0.804	0.862	0.832	0.709	0.765	0.736
	Assistance	0.851	0.79	0.82	0.714	0.65	0.681
CNN2D	**Slight**	0.967	0.989	0.978	0.774	0.766	**0.77**
	Assistance	0.988	0.966	0.977	0.743	0.751	**0.747**

Table 15: Cornwall model metrics.

The results in Table 15 show that CNN-2D is the model that best generalizes with respect to slight and assistance accidents, showing an increase of 3.4% and 6.6% respectively, taking as a reference the MLP, which has the second with the best metrics for both classes. The improvement in the CNN-2D model over the faster Naive-Bayes model is 7.7% for slight accidents and 19.5% for assistance accidents.

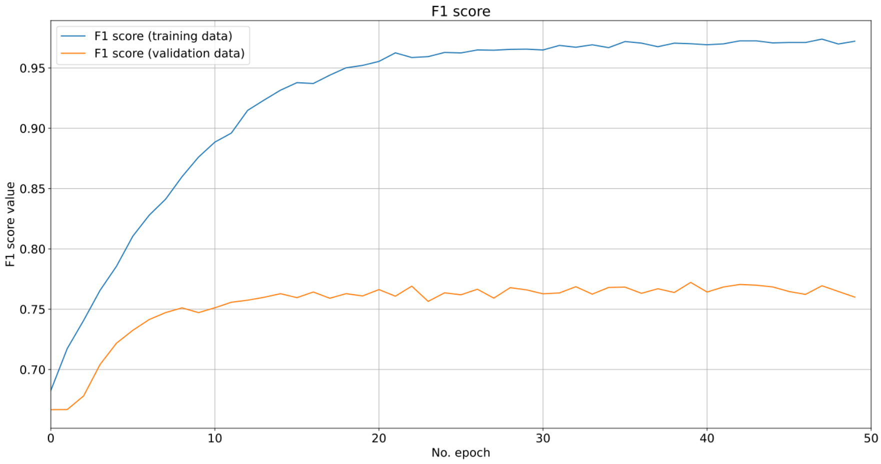


Figure 11: Evolution of the F1-score of the CNN-2D in training and test set Cornwall.

Figure 11 shows the evolution of the network on the training and validation data. It is observed that the performance for the training set does not stop increasing over the epochs, reaching its maximum value in 48. However, the validation set yields this in epoch 39; in subsequent epochs, this value remains constant, and it can be inferred that the network no longer generalizes from this epoch.

All the experiments carried out in this research were executed on a machine with Intel Core i7-12700F CPU with VENGEANCE LPX 64GB (2 x 32GB) DDR4 DRAM 3200MHz RAM and an NVIDIA GeForce RTX 3090 Ti graphics card.

4.3.7. Performance Comparison Summary

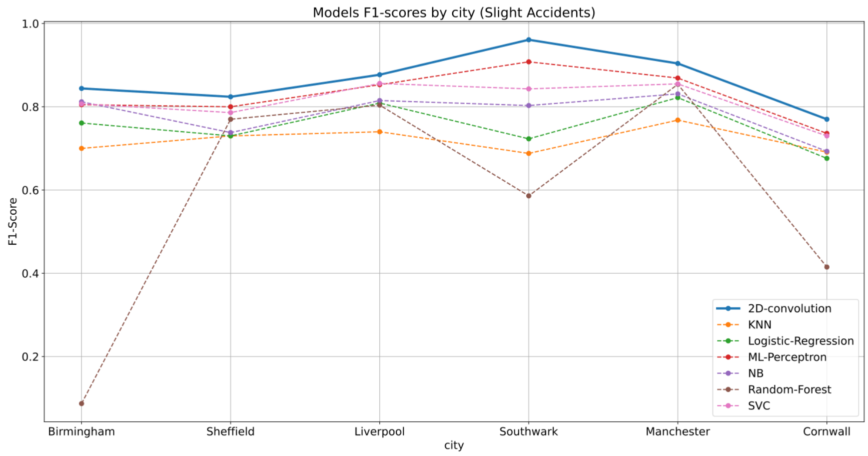


Figure 12: Models Slight F1-Score by city

If we analyse the F1-Score obtained for each city for slight accidents (see Fig. 12), it can be observed that the proposed 2D-Convolutional Neural Network (CNN-2D) model generally obtains the best results with the second position varying between the MLP and SVC model, depending on the city. Similarly, it is observed that the NB, KNN, RF and LR models alternate consecutively among the following positions depending on the city, evidencing that none of these is able to generalize consistently across the different cities. Finally, it can also be seen that, in Southwark, the non-deep learning models have more difficulty in achieving decent metrics, showing a drop in performance compared to the last city shown as a reference (Liverpool).

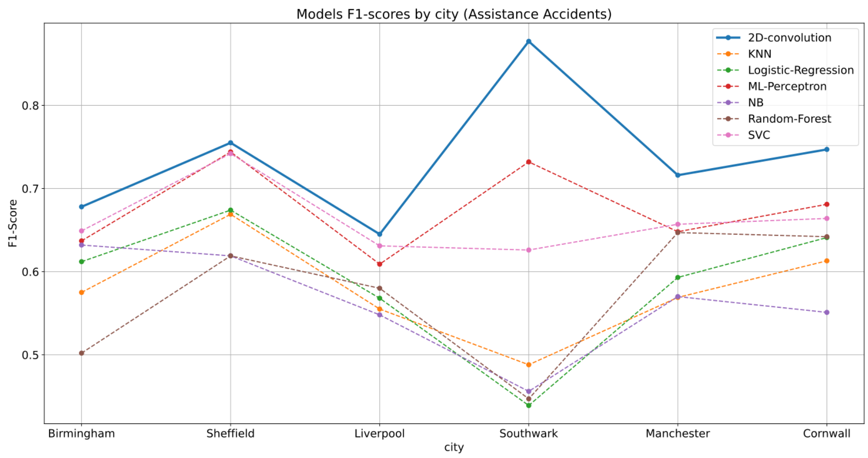


Figure 13: Models Assistance F1-Score by city.

Figure 13 shows the F1-Score values obtained from each model in each of the cities for assistance accidents. It can be observed that the best performing model based on this metric is the 2D-Convolutional Neural

Network (CNN-2D), typically followed by the MLP, while the SVC, NB, KNN, RF and LR models again alternate between the next positions. The deep learning based models, again in the city of Southwark obtain a notable difference with respect to the rest of the models.

The figures show the evolution of the CNN-2D model for each training epoch. It can be observed that the model generalizes as it learns from the training data offered at each stage. The evolution of the F1-Score as the epochs go by remains constant while the training one is increased.

Figure 13 shows the values of the F1-Score, for accidents that require assistance, obtained for each model and in each of the cities. It can again be seen that the model with the best performance, based on this metric, is the 2D-Convolutional Neural Network (CNN-2D) followed by the MLP, while the NB, KNN, RF and LR models alternate again between the following positions. We can conclude that the models based on deep learning, again in the city of Southwark, obtain a notable difference with respect to the rest of the models.

Model	Mean F1-Score
CNN-2D	**0.736**
MLP	0.675
SVC	0.662
LR	0.588
KNN	0.578
RF	0.573
NB	0.563

Table 16: Mean F1-Scores considering all cities.

5. Conclusions

This work analysed different fields of artificial intelligence, with the aim being to provide a generalized model that performs predictions of traffic accidents severity. This paper thus presents a framework based on two-dimensional convolutional neural networks, with datasets from several cities in the UK, the variables of which are grouped into categories. The transformations on the variables of the dataset are a critical point, and so we study these to verify whether 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 datasets are clearly unbalanced and, so the data were resampled using the Borderline Synthetic Minority Over-sampling Technique II (SMOTE-II). The proposed model then calculates a series of weights of the variables of the dataset by means of a boosting framework. This algorithm has, as input, several hyperparameters optimized by a genetic algorithm and the resampling data of the datasets. Its output is the weights of the variables used. With these weights, a set of matrices was constructed (one per accident) that serve as input to the two dimensional convolutional neural networks (2D CNN).

We tested the effectiveness and generality of the model using it on six different UK city datasets. The analysis and discussion of the results obtained reveals that, after training the model, it presents the best results in all selected metrics (precision, recall and F1 score) in the prediction of slight and assistance accidents. Thus, taking into account the F1 score of all cities for both types of severity, an improvement is observed in each of them, even reaching 14.5% in Southwark's assistance accidents.

It is should be highlighted that, among the main advantages of the proposed architecture, are its scalability and generality. That is, it can be applied to different data sets without the need for major changes to the implementation. Thus, as it is a general approach applicable to different datasets, a possible future work would be to apply it to other data related to the quality of life of the cities' inhabitants. Another advantage of the method is its speed, which involves the use of the model in real time.

In summary, in relation to the most recently published research [43], a considerable improvement has been achieved in the study and results of the models. By grouping the number of output classes of the networks in two (slight and assistance) and filtering the cities into areas so that the models are less sensitive to data bias, a practical and generalizable methodology has been achieved for any city with the necessary data. Furthermore, there is room for improvement of the two-dimensional convolutional network, optimizing the hyperparameters of this architecture for each city.

References

[1] Bin Cao, Weizheng Zhang, Xuesong Wang, Jianwei Zhao, Yu Gu, and Yan Zhang. A memetic algorithm based on two_arch2 for multi-depot heterogeneous-vehicle capacitated arc routing problem. Swarm and evolutionary computation, 63:100864, 2021.

[2] Sheng Su, Dongwen Xiong, Haijie Yu, and Xiaohua Dong. A multiple leaders particle swarm optimization algorithm with variable neighborhood search for multiobjective fixed crowd carpooling problem. Swarm and Evolutionary Computation, 72:101103, 2022.

[3] Kenny Santos, João P Dias, and Conceição Amado. A literature review of machine learning algorithms for crash injury severity prediction. Journal of safety research, 80:254–269, 2022.

[4] Zekun Yang, Wenping Zhang, and Juan Feng. Predicting multiple types of traffic accident severity with explanations: A multi-task deep learning framework. Safety science, 146:105522, 2022.

[5] Samuel Olugbade, Stephen Ojo, Agbotiname Lucky Imoize, Joseph Isabona, and Mathew O Alaba. A review of artificial intelligence and machine learning for incident detectors in road transport systems. Mathematical and Computational Applications, 27(5):77, 2022.

[6] Sylvia S Zajac and John N Ivan. Factors influencing injury severity of motor vehicle–crossing pedestrian crashes in rural connecticut. Accident Analysis , Prevention, 35(3):369–379, 2003.

[7] Mohammad Abrari Vajari, Kayvan Aghabayk, Mohammad Sadeghian, and Nirajan Shiwakoti. A multinomial logit model of motorcycle crash severity at australian intersections. Journal of Safety Research, 73:17–24, 2020.

[8] Zhibin Li, Pan Liu, Wei Wang, and Chengcheng Xu. Using support vector machine models for crash injury severity analysis. Accident Analysis , Prevention, 45:478–486, 2012.

[9] Joaquín Abellán, Griselda López, and Juan de Oña. Analysis of traffic accident severity using decision rules via decision trees. Expert Systems with Applications, 40(15):6047–6054, 2013.

[10] Mahdi Rezapour, Amirarsalan Mehrara Molan, and Khaled Ksaibati. Analyzing injury severity of motorcycle at-fault crashes using machine learning techniques, decision tree and logistic regression models. International Journal of Transportation Science and Technology, 9(2):89–99, 2020.

[11] Juan Li, Fengxiang Guo, Yanning Zhou, Wenchen Yang, and Dingan Ni. Predicting the severity of traffic accidents on mountain freeways with dynamic traffic and weather data. Transportation Safety and Environment, page tdad001, 2023.

[12] Hyeonchoel Jeong, Inhi Kim, Keejun Han, and Jungeun Kim. Comprehensive analysis of traffic accidents in seoul: Major factors and types affecting injury severity. Applied Sciences, 12(4):1790, 2022.

[13] Seyed Hessam-Allah Hashmienejad and Seyed Mohammad Hossein Hasheminejad. Traffic accident severity prediction using a novel multi-objective genetic algorithm. International journal of crashworthiness, 22(4):425–440, 2017.

[14] Holman Ospina-Mateus, Leonardo Augusto Quintana Jiménez, Francisco J Lopez-Valdes, Shyrle Berrio Garcia, Lope H Barrero, and Shib Sankar Sana. Extraction of decision rules using genetic algorithms and simulated annealing for prediction of severity of traffic accidents by motorcyclists. Journal of ambient intelligence and humanized computing, 12(11):10051–10072, 2021.

[15] Tibebe Beshah, Dejene Ejigu, Pavel Krömer, V'clav Sn´el, Jan Plato, and Ajith Abraham. Learning the classification of traffic accident types. In 2012 Fourth International Conference on Intelligent Networking and Collaborative Systems, pages 463–468, 2012.

[16] Mehmet Kunt, Iman Aghayan, and Nima Noii. Prediction for traffic accident severity: Comparing the artificial neural network, genetic algorithm, combined genetic algorithm and pattern search methods. Transport, 26:353–366, 12 2011.

[17] Amir Mohammadian Amiri, Amirhossein Sadri, Navid Nadimi, and Moe Shams. A comparison between artificial neural network and hybrid intelligent genetic algorithm in predicting the severity of fixed object crashes among elderly drivers. Accident Analysis & Prevention, 138:105468, 2020.

[18] Xiaoning Gu, Ting Li, Yonghui Wang, Liu Zhang, Yitian Wang, and Jinbao Yao. Traffic fatalities prediction using support vector machine with hybrid particle swarm optimization. Journal of Algorithms & Computational Technology, 12(1):20–29, 2018.

[19] Jun Yuan, Changjian Chen, Weikai Yang, Mengchen Liu, Jiazhi Xia, and Shixia Liu. A survey of visual analytics techniques for machine learning. Computational Visual Media, 7:3–36, 2021.

[20] Afamdi O Ezenwa. Trends and characteristics of road traffic accidents in nigeria. Journal of the Royal Society of Health, 106(1):27–29, 1986.

[21] Kun Li, Haocheng Xu, and Xiao Liu. Analysis and visualization of accidents severity based on lightgbm-tpe. Chaos, Solitons Fractals, 157:111987, 2022.

[22] György Szabó and Gabor Fath. Evolutionary games on graphs. Physics reports, 446(4-6):97–216, 2007.

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

[24] Junhua Wang, Yumeng Kong, Ting Fu, and Joshua Stipancic. The impact of vehicle moving violations and freeway traffic flow on crash risk: An application of plugin development for microsimulation. PLoS one, 12(9):e0184564, 2017.

[25] Junhua Wang, Yumeng Kong, and Ting Fu. Expressway crash risk prediction using back propagation neural network: A brief investigation on safety resilience. Accident Analysis , Prevention, 124:180–192, 2019.
[26] A deep learning based traffic crash severity prediction framework. Accident Analysis , Prevention, 154:106090, 2021.
[27] Ossama Abdel-Hamid, Abdel-rahman Mohamed, Hui Jiang, Li Deng, Gerald Penn, and Dong Yu. Convolutional neural networks for speech recognition. IEEE/ACM Transactions on audio, speech, and language processing, 22(10):1533–1545, 2014.
[28] Qirong Mao, Ming Dong, Zhengwei Huang, and Yongzhao Zhan. Learning salient features for speech emotion recognition using convolutional neural networks. IEEE transactions on multimedia, 16(8):2203–2213, 2014.
[29] Pawel Swietojanski, Arnab Ghoshal, and Steve Renals. Convolutional neural networks for distant speech recognition. IEEE Signal Processing Letters, 21(9):1120–1124, 2014.
[30] Alok Sharma, Edwin Vans, Daichi Shigemizu, Keith Boroevich, and Tatsuhiko Tsunoda. Deepinsight: A methodology to transform a non-image data to an image for convolution neural network architecture. Scientific Reports, 9, 08 2019.
[31] Waseem Rawat and Zenghui Wang. Deep convolutional neural networks for image classification: A comprehensive review. Neural computation, 29(9):2352–2449, 2017.
[32] Turker Ince, Serkan Kiranyaz, Levent Eren, Murat Askar, and Moncef Gabbouj. Real-time motor fault detection by 1-d convolutional neural networks. IEEE Transactions on Industrial Electronics, 63(11):7067–7075, 2016.
[33] Serkan Kiranyaz, Onur Avci, Osama Abdeljaber, Turker Ince, Moncef Gabbouj, and Daniel J. Inman. 1d convolutional neural networks and applications: A survey, 2019.
[34] Steve Lawrence, C Lee Giles, Ah Chung Tsoi, and Andrew D Back. Face recognition: A convolutional neural-network approach. IEEE transactions on neural networks, 8(1):98–113, 1997.
[35] Chris Tensmeyer and Tony Martinez. Analysis of convolutional neural networks for document image classification, 2017.
[36] Yunjie Liu, Evan Racah, Joaquin Correa, Amir Khosrowshahi, David Lavers, Kenneth Kunkel, Michael Wehner, William Collins, et al. Application of deep convolutional neural networks for detecting extreme weather in climate datasets. arXiv preprint arXiv:1605.01156, 2016.
[37] Md Ebrahim Shaik, Md Milon Islam, and Quazi Sazzad Hossain. A review on neural network techniques for the prediction of road traffic accident severity. Asian Transport Studies, 7:100040, 2021.
[38] Yafang Liu, Chaozhong Wu, Jianghui Wen, Xinping Xiao, and Zhijun Chen. A grey convolutional neural network model for traffic flow prediction under traffic accidents. Neurocomputing, 500:761–775, 2022.
[39] Lu Wenqi, Luo Dongyu, and Yan Menghua. A model of traffic accident prediction based on convolutional neural network. In 2017 2nd IEEE international conference on intelligent transportation engineering (ICITE), pages 198–202. IEEE, 2017.
[40] Jiyao An, Li Fu, Meng Hu, Weihong Chen, and Jiawei Zhan. A novel fuzzy-based convolutional neural network method to traffic flow prediction with uncertain traffic accident information. Ieee Access, 7:20708–20722, 2019.
[41] Alexandra Laiou, Eleonora Papadimitriou, George Yannis, and Alberto Milotti. Road safety data and information availability and priorities in south-east european regions. Transportation Research Procedia, 25:3703–3714, 2017. World Conference on Transport Research - WCTR 2016 Shanghai. 10-15 July 2016.
[42] Nicholas Fiorentini and Massimo Losa. Handling imbalanced data in road crash severity prediction by machine learning algorithms. Infrastructures, 5(7), 2020.
[43] Luis Pérez-Sala, Manuel Curado, Leandro Tortosa, and Jose F Vicent. Deep learning model of convolutional neural networks powered by a genetic algorithm for prevention of traffic accidents severity. Chaos, Solitons & Fractals, 169:113245, 2023.
[44] Hui Han, Wen-Yuan Wang, and Bing-Huan Mao. Borderline-smote: a new over-sampling method in imbalanced data sets learning. pages 878–887, 2005.
[45] Reda Yacouby and Dustin Axman. Probabilistic extension of precision, recall, and f1 score for more thorough evaluation of classification models. In Proceedings of the First Workshop on Evaluation and Comparison of NLP Systems, pages 79–91, Online, Nov 2020. Association for Computational Linguistics.
[46] Usha Ruby and Vamsidhar Yendapalli. Binary cross entropy with deep learning technique for image classification. Int. J. Adv. Trends Comput. Sci. Eng, 9(10), 2020.
[47] Shenglei Chen, Geoffrey I. Webb, Linyuan Liu, and Xin Ma. A novel selective naïve bayes algorithm. Knowledge-Based Systems, 192:105361, 2020.
[48] Shamshe Alam, Sanjay Kumar Sonbhadra, Sonali Agarwal, and P. Nagabhushan. One-class support vector classifiers: A survey. Knowledge-Based Systems, 196:105754, 2020.
[49] Hemingway Huawei. ¿qué es knn ó k-nearest neighbor?, 2020.
[50] Matthias Schonlau and Rosie Yuyan Zou. The random forest algorithm for statistical learning. The Stata Journal, 20(1):3–29, 2020.
[51] Abhik Das. Logistic regression. In Encyclopedia of Quality of Life and Well-Being Research, pages 1–2. Springer, 2021.
[52] Felipe Arias Del Campo, María Cristina Guevara Neri, Osslan Osiris Vergara Villegas, Vianey Guadalupe Cruz Sánchez, Humberto de Jesús Ochoa Domínguez, and Vicente García Jiménez. Auto-adaptive multilayer perceptron for univariate time series classification. Expert Systems with Applications, 181:115147, 2021.
[53] Department for Transport. Reported road collisions, vehicles and casualties tables for great britain, 2023. Available at https://www.gov.uk/government/statistical-data-sets/reported-road-accidents-vehicles-and-casualties-tables-for-great-britain.