International Journal of Injury Control and Safety Promotion

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/nics20

Defining machine learning algorithms as accident prediction models for Italian two-lane rural, suburban, and urban roads

Nicholas Fiorentini, Pietro Leandri & Massimo Losa

To cite this article: Nicholas Fiorentini, Pietro Leandri & Massimo Losa (2022): Defining machine learning algorithms as accident prediction models for Italian two-lane rural, suburban, and urban roads, International Journal of Injury Control and Safety Promotion, DOI: 10.1080/17457300.2022.2075397

To link to this article: https://doi.org/10.1080/17457300.2022.2075397

Published online: 25 May 2022.

INTERNATIONAL JOURNAL OF INJURY CONTROL AND SAFETY PROMOTION
https://doi.org/10.1080/17457300.2022.2075397

Taylor & Francis
Taylor & Francis Group

RESEARCH ARTICLE

Defining machine learning algorithms as accident prediction models for Italian two-lane rural, suburban, and urban roads

Nicholas Fiorentini, Pietro Leandri and Massimo Losa

Department of Civil and Industrial Engineering (DICI), Engineering School, University of Pisa, Pisa, Italy

ABSTRACT

Four Accident Prediction Models have been defined for Italian two-lane rural, suburban, and urban roads by exploiting different Machine Learning Algorithms. Specifically, a Classification and Regression Tree, a Boosted Regression Tree, a Random Forest, and a Support Vector Machine have been implemented to predict the number of Fatal and Injury crashes on a 905-km network, which experienced 5,802 FI crashes in 2008-2016. The dataset incorporates geometrical, functional, and environmental information. Several performance metrics have been computed, such as Determination Coefficient, Mean Absolute Error, Root Mean Square Error, and scatterplots. Outcomes suggest that Support Vector Machine outperforms the other Machine Learning Algorithms for predicting Fatal and Injury crashes. In Addition, the computation of Predictor Importance shows that traffic flow, the density of intersections, driveway density, and type of area are the most impacting factors on crash likelihood. Road authorities may use these findings for conducting reliable safety analyses.

ARTICLE HISTORY
Received 7 October 2021
Revised 21 April 2022
Accepted 5 May 2022

KEYWORDS
Accident prediction models; crash prediction models; machine learning algorithms; Fatal and injury crashes; two-lane roads; road safety

Introduction

Motivation and purpose

Road accidents prevention is of paramount importance worldwide, considering the dramatic number of casualties due to crash occurrences; the latest global report on road safety published in 2018 by the World Health Organization (2018) states:

> 'the number of annual road traffic deaths has reached 1.35 million. Road traffic injuries are now the leading killer of people aged 5-29 years',

and also

> 'the price paid for mobility is too high, especially because proven measures exist. Drastic action is needed to put these measures in place to meet any future global target that might be set and save lives'.

With these remarks as a guideline, academic researchers regularly propose methods for measuring the safety level of a road element by correlating a reference parameter (e.g., the crash number, crash rate, crash frequency, or severity of impacts) with crash conditioning factors related to the trinomial road-vehicle-environment. In most such studies, researchers exploit a specific statistical or data-driven modelling approach and attempt to relate one or more conditioning factors to the reference parameter; there is a general awareness that it is not trivial to recognise which factors impact the crash occurrence and in which magnitude.

Intending to provide further findings and support in this research field, this study aims to develop, compare, and emphasise the strengths and weaknesses of four Accident Prediction Models (APMs) deriving from non-parametric data-driven modelling in the context of minor Italian two-lane rural, suburban, and rural roads. We considered a comprehensive network that extends for more than 900 km and structured a dataset incorporating functional, geometrical, environmental, and crash information. It is worth mentioning that, to the best of our knowledge, the present network has not been investigated yet for defining regression-based APMs. Moreover, considering the available literature, few studies investigated minor roads and intersections belonging to the so-called secondary road network, i.e., the road network that links the primary road network (motorways and freeways) with local roads. Therefore, regional and local road authorities could adopt the outcomes of the present analysis as novel guidelines for implementing different data-driven APMs, identifying the most critical road sections, and comparing different maintenance interventions (in terms of both functional and horizontal, vertical, and cross-sectional geometry modification) on the road network they manage. Moreover, they can evaluate the impact of each predictor on crash likelihood. Therefore, data-driven APMs can be useful for supporting the decision-making processes of road authorities, especially when they need to justify infrastructure maintenance plans or new road works.

CONTACT Nicholas Fiorentini nicholas.fiorentini@phd.unipi.it

The paper moves on with a state-of-the-art subsection for recapping recent and relevant works on APMs by Machine Learning Algorithms (MLAs). Afterwards, in Chapter 2, the workflow, theory, and implementation process of the APMs are discussed. In Chapter 3, the study area and data collection are introduced. In Chapter 4, we report and discuss the leading findings, recognising the strengths and weaknesses of each APM. The paper wraps up with a conclusion and a reference section.

Accident prediction models by machine learning algorithms

Several works can be found in the literature on MLAs as APMs, mainly for major road facilities. Compared to statistical parametric approaches, a MLA allows for performing regression and classification analyses without substantially modifying its modelling process. Furthermore, MLAs are non-parametric models; accordingly, they do not require that the analyst fixes a distribution of the output target (i.e., the dependent variable) and specifies a functional form of the model that links inputs (i.e., independent variables) and output. These are undoubtedly significant advantages, which broadly extend the field of application of traditional statistical models. Nonetheless, MLAs have some limits; firstly, MLAs act as a 'black box', and the analyst has marginal control over the calibration process. Moreover, the outcomes are not usually high interpretable. Furthermore, there are no relations that allow quantifying the impact that each explanatory variable has on the output of the algorithm; some MLAs allow evaluating which inputs have the most significant impact qualitatively. Finally, several hyperparameters have to be specified before the training phase of each MLA; accordingly, their calibration requires a particular experience by the analyst.

Table 1 reports more than 35 works on road safety analyses by MLAs. Table 1 shows the Reference, the type of task (Classification, C, or Regression, R), the modelling purpose, the road network investigated, the MLAs implemented, and the performance metrics computed for performance assessment.

From Table 1, we can derive several considerations. Firstly, there are two types of MLAs in the road safety analysis: those used for classification purposes and those used for regression purposes. The first group is widely investigated and includes the MLAs used for road blackspots prediction and crash severity prediction. The second group, much less investigated, includes the MLAs employed to predict the number of accidents or the crash frequency on a road element. In addition, other road safety aspects have been investigated, such as crash angle prediction and crash risk prediction. In road blackspot detection and real-time crash prediction, the algorithms operate as binary classifiers, in which the output of the algorithm can be 'Accident Case' or 'Non-Accident Case'. In studies on crash severity prediction, the task to be solved is a multiclassification, in which the output of the algorithms can be, for instance, distributed within three classes (Property Damage Only Accident, Injury Accident, Fatal Accident). Conversely, in the studies where MLAs are trained to perform regressions, the output of such models is a continuous variable with real numbers.

Secondly, studies mainly offer the analysis of major roads, while there are few studies on minor roads and intersections; consequently, these latter road networks should constitute a promising research field that needs to be investigated.

Thirdly, considering that there are no theories or empiric rules for determining the best MLA for a specific task, authors generally use different algorithms and compare their performance, attempting to identify the most appropriate one. In several studies, authors implemented basic MLAs, such as Logistic Regression, K-Nearest Neighbour, Naive Bayes, and Classification and Regression Tree (CART). These algorithms are relatively simple to implement, do not require many hyperparameters to be tuned, and their outcomes are interpretable. Generally, they are adopted as benchmark algorithms compared with other more complex ones. Indeed, we identified numerous studies in which more sophisticated MLAs are calibrated, such as RF, Boosted Regression Tree (BRT), and Support Vector Machine (SVM). These algorithms are calibrated according to a somewhat complex process and require more hyperparameters to be tuned (especially for SVM-based algorithms). Finally, there are studies in which the authors implement Artificial Neural Networks. Depending on the architecture chosen for the neural network, these algorithms can be associated with Deep Learning Algorithms; consequently, the calibration procedure is even more complex than the previous ones and requires several hyperparameters to be carefully evaluated and tuned.

Finally, performance metrics are distinguished according to the task to be solved. For example, in the case of classification tasks, researchers ordinarily compute the Accuracy, Recall, Precision, F1, and Area Under the Receiver Operating Characteristic. These metrics are generally employed for assessing the reliability of each output class to be predicted. Researchers exploit different performance metrics in regression tasks, such as the Determination Coefficient (R^2), the Root Mean Square Error (RMSE), the Mean Absolute Error (MAE), and the Scatterplots.

Background on machine learning algorithms

In the following subsections, we briefly introduced the MLAs employed as APM in the present study, and the hyperparameters that need to be tuned are defined. We have intentionally kept this part narrow since several comprehensive tutorials on implementing such MLAs and pillar works are quoted. The modelling framework (training, validation, test, evaluation) is common to all MLAs and is discussed in the 'Methodology' Chapter. It is worth mentioning that a supervised learning approach trains these MLAs, i.e., algorithms learn from a certain percentage of samples (by knowing both the input features and the output target) during their training phase. By exploiting these data, the optimal hyperparameters can be recognised. In this

Table 1. Road Safety-related Analyses by MLAs.

Reference	Task	Purpose	Network	Algorithms	Evaluation Metrics
(Fiorentini & Losa, 2020b)	C	Blackspot Detection	Two-lane rural roads	LR, CART, RF, KNN, NB	Precision, Recall, F1, Accuracy, AUROC
(Basso et al., 2018)	C	Blackspot Detection	Urban Expressways	SVM, LR	Recall, FPR
(Dogru & Subasi, 2018)	C	Blackspot Detection	Freeways	SVM, ANN, RF	Accuracy, Recall, TNR
(Hossain & Muromachi, 2012)	C	Blackspot Detection	Expressways	RMNL, BBN	Accuracy, Recall, FPR
(P. Li et al., 2020)	C	Blackspot Detection	Urban Arterials	LSTM-CNN	Recall, FPR, AUROC
(Lv et al., 2009)	C	Blackspot Detection	Highways	SVM	OER, NDER, DNER
(Schlögl et al., 2019)	C	Blackspot Detection	Highways	LR, BQR, MARS, RF, ERT, EGB, SVM, BRNN	Accuracy, Recall, TNR, FPR, AUROC
(Theofilatos et al., 2019)	C	Blackspot Detection	Urban Motorways	KNN, NB, CART, RF, SVM, ANN, DNN	Accuracy, Recall, TNR, FPR, AUROC
(Xiao, 2019)	C	Blackspot Detection	Freeways	SVM, KNN, SVM-KNN ensemble	Accuracy, Recall, FPR
(Xiao & Liu, 2012)	C	Blackspot Detection	Freeways	SVM, SVM ensemble, SVM-MKL, SVM-MKL ensemble	Accuracy, Recall, FPR
(Yuan et al., 2019)	C	Blackspot Detection	Signalized Intersections	LSTM-RNN, CLM	Recall, FPR, AUROC
(You et al., 2017)	C	Blackspot Detection	Freeways	SVM	Accuracy, Recall, FPR, AUROC
(Fiorentini & Losa, 2020a)	C	Crash Severity Prediction	Motorways, A-class, B-class, and C-class UK roads	CART, KNN, LR, RF	Accuracy, TPR, FPR, TNR, Precision, F1
(Ijaz et al., 2021)	C	Crash Severity Prediction	Urban Roads	DJ, RF, CART	Accuracy, Precision, Recall, GM, AUROC
(Rahim & Hassan, 2021)	C	Crash Severity Prediction	Highway Work Zones	CNN, SVM	Precision, Recall
(Jamal et al., 2021)	C	Crash Severity Prediction	Rural Highways	XGBoost, LR, RF, CART	Accuracy, Precision, Recall, GM, F1, Specificity, K
(Jamal & Umer, 2020)	C	Crash Severity Prediction	Rural Highways	ANN	Accuracy, Specificity, Recall
(Abellán et al., 2013)	C	Crash Severity Prediction	Rural roads	CART	/
(Al Mamlook et al., 2019)	C	Crash Severity Prediction	Freeways	BRT, LR, RF, NB	Accuracy, Precision, Recall, F1, AUROC
(Mokhtarimousavi et al., 2019)	C	Crash Severity Prediction	Work zones	SVM, MNL	Accuracy, Recall, Specificity, Precision
(Chang & Wang, 2006)	C	Crash Severity Prediction	/	CART	Accuracy
(Chen et al., 2016)	C	Crash Severity Prediction	Urban, rural non-interstate, interstate roads	SVM	Accuracy
(Iranitalab & Khattak, 2017)	C	Crash Severity Prediction	/	KNN, SVM, RF, MNL	Accuracy, POCC, OPE, SPE
(Z. Li et al., 2012)	C	Crash Severity Prediction	Freeway diverge areas	SVM, OPM	Accuracy
(Zhang et al., 2018)	C	Crash Severity Prediction	Freeway diverge areas	KNN, CART, RF, SVM, MNL, OPM	Accuracy
(Abdelwahab & Abdel-Aty, 2001)	C	Crash Severity Prediction	Before, at, and After toll plazas	ANN, RBF-ANN, NLM, OLM	Accuracy
(Alkheder et al., 2017)	C	Crash Severity Prediction	/	ANN, OPM	Accuracy, AUROC
(Wahab & Jiang, 2020)	C	Crash Severity Prediction	/	ANN, PART, CART	Accuracy, Recall, Specificity, Precision, AUROC
(Delen et al., 2006)	C	Crash Severity Prediction	/	ANN	Accuracy
(Delen et al., 2017)	C	Crash Severity Prediction	/	ANN, SVM, CART, LR	Accuracy, Recall, Specificity, AUROC
(Wang et al., 2019)	C	Crash Risk Prediction	Expressways	ANN	Accuracy, TRP, FPR, AUROC
(Abdel-Aty & Haleem, 2011)	R	Crash Angle Prediction	Stop and right-of-way intersections	MARS, NBR	MAD, MSPE
(Singh et al., 2016)	R	Crash Frequency Prediction	Highways	M5T, NBR, RENB	R, RMSE, MAE, Scatterplot
(Singh et al., 2018)	R	Crash Frequency Prediction	Highways	SVM, NBR, RENB	R, RMSE, MAE, Scatterplot
(Singh et al., 2020)	R	Crash Frequency Prediction	Highways	DNN, GEP, RENB	R, RMSE, MAE, Scatterplot
(Zhang & Yang, 1997)	R	Crash Frequency Prediction	Highways	KNN, ANN, LR	APE

Acronyms of APMs: ANN = Artificial Neural Network, BBN = Bayesian Belief Network, BQR = Binary Quantile Regression, BRNN = Bayesian Regularized Neural Network, CART = Classification and Regression Tree, CLM = Conditional Logistic Model, DJ = Decision Jungle, DNN = Deep Neural Network, EGB = Extreme Gradient Boosting, ERT = Extremely Randomized Trees, GEP = Gene Expression Programming, KNN = K-Nearest Neighbor, LR = Logistic Regression, LSTM-CNN = Long Short-Term Memory Convolutional Neural Network, LSTM-RNN = Long Short-Term Memory Recurrent Neural Network, M5T = M5 model Tree, MARS = Multivariate Adaptive Regression Splines, NB = Naïve Bayes, NBR = Negative Binomial Regression, NLM = Nested Logit Model, OLM = Ordered Logit Model, OPM = Ordered Probit Model, PART = Rule Induction, RBF-ANN = Radial Basis Function Artificial Neural Network, RENB = Random Effect Negative Binomial, RF = Random Forest, RMNL = Random Multinomial Logit, SVM = Support Vector Machines, SVM-MKL = Multiple Kernel Support Vector Machines, XGBoost = Extreme Gradient Boosting.

Acronyms of performance metrics: APE = Absolute Prediction Error, AUROC = Area Under the Receiver Operating Characteristic, DNER = Dangerous Pattern to Normal Pattern Error Rate, FPR = False Positive Rate, GM = Geometrical Mean, K = Cohen's Kappa, MAD = Mean Absolute Deviation, MAE = Mean Absolute Error, MSPE = Mean Squared Prediction Error, NDER = Normal Pattern to Dangerous Pattern Error Rate, OER = Overall Error Rate, OPE = Overall Prediction Error, POCC = Predicted Overall Costs of Crashes, R = Correlation Coefficient, RMSE = Root Means Square Error, SPE = Specific Prediction Error, TNR = True Negative Rate, TPR = True Positive Rate.

phase, the Goodness-of-Fit performance is evaluated. Once trained, MLAs make predictions on the unknown remaining percentage of samples during their test phase (by knowing the input features only). In this phase, the predictive performance is evaluated.

Classification and regression tree

Breiman (Breiman et al., 1984) introduced the CART algorithm, a hierarchical process that identifies a tree-based structure composed of homogeneous zones called nodes. Starting from the root node, the tree-based structure takes form. Intermediate levels are composed of branch nodes, while the last level is composed of leaf nodes. The training samples are iteratively split according to specific decision rules, passing from one level to the next one. A decision rule is a binary function that splits the samples according to a value (the so-called cut-point) of one of the input features. Decisions are taken at each node except at the last level; at each leaf node, the CART predicts the average value of the output targets belonging to each observation falling in that leaf node. Indeed, leaf nodes are considered homogeneous or pure. To define the decision rules for each node, CART exploits the Recursive Partitioning (RP) algorithm (Breiman et al., 1984; Loh & Shih, 1997). For each CART node, the RP algorithm identifies the most appropriate split (i.e., the decision rule) for dividing the sample of observations falling into that node into two subsamples (i.e., the branch nodes). The algorithm maximises the impurity decrease of that node by considering an impurity metric; in regression tasks, such impurity at each node is represented by the MSE of the samples belonging to the node. RP is stopped when observations in a node have a difference between target values lower than a minimum MSE threshold). RP starts in the root node; once the best decision rule is identified, the RP is implemented in next-level nodes. When nodes become leaf nodes, CART predicts the target output. In this study, an MSE equal to 10^{-6} represents the termination criterion.

In the CART algorithm, the analyst has to specify four hyperparameters: (1) the fixed number of the input features random sampled for identifying the decision rules at each node, (2) the maximum number of splits, which implies the level of complexity of the tree, (3) the minimum leaf node size, i.e., the minimum number of observation that can belong to a leaf node, and (4) the minimum parent size for each child node, i.e., the minimum number of observation that can belong to a branch node.

Boosted regression tree

Schapire (1999) introduced the BRT as an ensemble learner, i.e., a MLA composed of several base learners. Boosting technique is applied to tree-based learners; indeed, BRT usually relies on several CART predictions in an aggregate way. Boosting is an iterative procedure. At each iteration, a new CART is trained specifically with those samples that, in the previous iteration, are responsible for the higher residuals. Once the number of CARTs composing the BRT is reached, the BRT algorithm ends, and the final prediction is equal to the sum of the prediction of the first trained CART and a weighted value of all the other CART predictions. The BRT algorithm adopts a stage-wise procedure, i.e., the existing CARTs are left unchanged as the model is enlarged. The leading analytical relationship of BRT can be found in the comprehensive tutorials of Elith et al. and Schapire (Elith et al., 2008; Schapire, 2002).

In the BRT algorithm, the analyst has to specify three hyperparameters: (1) the number of CART belonging to the ensemble, (2) the learning rate, i.e., the ratio of learning from each new CART, and (3) the minimum size of the leaf nodes.

Random forest

RF algorithm (Breiman, 2001) is an ensemble learner based on the parallel aggregation of several CARTs. In the RF algorithm, each CART has the same importance concerning the final prediction; once trained, RF makes predictions on unknown new samples by averaging the predicted target output of each CART. The strength of RF lies in the fact that several uncorrelated CARTs (i.e., CARTs with high variance and low bias) are aggregated. In order to train uncorrelated trees, the RF algorithm exploits the Bootstrap Aggregation (Bagging) process; several subsets of the training set are defined with replacement, and then the RF exploits each of them in a different CART training. Moreover. RF exploits the Feature Randomness process; at each CART node, the algorithm randomly chooses a fixed number of input features to identify the best decision rule.

In the RF algorithm, the analyst has to specify two hyperparameters: (1) the number of CART belonging to the ensemble learner and (2) the fixed number of the input features randomly sampled for identifying the decision rules at each node.

Support vector machines

SVM was defined in the early sixties by the research of Vapnik and Learner (Vapnik & Lerner, 1963) as a classification learner. Subsequently, the SVM has also been extended to regression tasks (A. Smola, 1996; Vapnik et al., 1997). The core idea of SVM is to map the input features into a high-dimensional feature space through a non-linear kernel function. Therefore, the SVM detects an optimal hyperplane in such a new feature space. In the SVM for regression, the optimal hyperplane lies close to as many points as possible.

Readers may find a comprehensive tutorial on how to train an SVM for regression in the work of Smola (A. J. Smola et al., 2004). Many kernel functions can be implemented for mapping input features into the high-dimensional feature space; according to several studies (Iranitalab & Khattak, 2017; Mokhtarimousavi et al., 2019; Singh et al., 2018; Theofilatos et al., 2019), the

radial basis kernel function, also called Gaussian kernel (Aizerman et al., 1964; Micchelli, 1986), has been implemented.

In the SVM algorithm, the analyst has to specify four hyperparameters: (1) the so-called Box Constraint in order to define the trade-off between the flatness of the hyperplane and the amount up to which errors higher than a certain threshold are accepted, (2) the value of such a threshold, (3) the value of the standard deviation of the Gaussian kernel function, and (4) if the input features have to be standardised before the training phase.

Methodology

Workflow

Figure 1 shows the leading modelling steps of the present research.

Firstly, relevant information related to traffic flow, road geometry, road context, and crash history of the road network has been provided by the Tuscany Region Road Administration (TRRA). Therefore, a Geographic Information System (GIS) environment (ESRI., 2011) has been exploited to define the dataset of road elements and link the collected information. Next, the infrastructure-related features (i.e., the predictors) have been exploited for training each MLA to predict the number of Fatal and Injury (FI) crashes that occurred in the nine year-period considered (i.e., the target output). It is worth mentioning that the software MATLAB R2020b (The MathWorks Inc, 2020) has been exploited for training and testing the MLAs. Four MLAs have been calibrated; the hyperparameters have been identified during the training phase (exploiting 70% of samples) employing the Bayesian Optimization Algorithm (BOA). Accordingly, such MLAs have been tested by exploiting the remaining 30% of samples. To assess the Goodness-of-Fit and the generalisation capabilities of MLAs, some numerical (R2, RMSE, MAE) and graphical (scatterplots) metrics have been computed and plotted, respectively. As a result, we compared the MLAs and recognised specific strengths and weaknesses by these metrics. Finally, the impact of each predictor has been evaluated by computing the importance of each predictor.

Modelling framework and hyperparameters estimation

The modelling framework of supervised MLAs usually includes three stages: training phase, validation phase, and testing phase. The validation phase is performed during training; accordingly, two datasets are required, namely the training and test sets. Therefore, the whole database has been divided randomly into two parts: we exploited 70% of samples (i.e., 1,270 road elements) for training MLAs, while the remaining 30% (i.e., 544 road elements) for testing the algorithms. The validation phase involves a 10-Fold Cross-Validation (CV) procedure (Kohavi, 1995); this phase is essential for training MLAs with the highest generalisation capabilities. Moreover, such a process should make algorithms less prone to overfit data. By implementing the 10-Fold CV, the training set has been split into 10 folds. Therefore, the MLA is trained (i.e., the set of hyperparameters is tuned) using 9 folds according to a specified algorithm. Subsequently, the MLA is validated on the remaining one, i.e., the remaining fold is used for testing the algorithm, and a loss function (i.e., the MSE) is computed. The process is iterated 10 times, identifying 10 sets of hyperparameters; this allows exploiting each sample for both training and validation of a MLA, ensuring that the random split in training/validation does not suffer from selectivity biases. The hyperparameters that provide the least loss function constitute the most appropriate hyperparameters sought for the MLA.

These hyperparameters can be searched for in each fold by implementing different manual and automatic approaches. In the present paper, we carried out an optimisation procedure according to the BOA algorithm (Martin et al., 1999). The BOA provides a very flexible and efficient optimisation procedure for optimising an unknown function in the hyperparameter space since it is fully automatic; indeed, the analyst has to specify only the number of iterations. Moreover, BOA does not suffer from the curse of dimensionality, which could be easily an issue in Grid Search and Random Search algorithms.

Exploiting the Bayes theorem, BOA merges prior information concerning the distribution of the function to be optimised with sample information (the evidence, i.e., at the beginning of a BOA iteration, a set of evaluations of

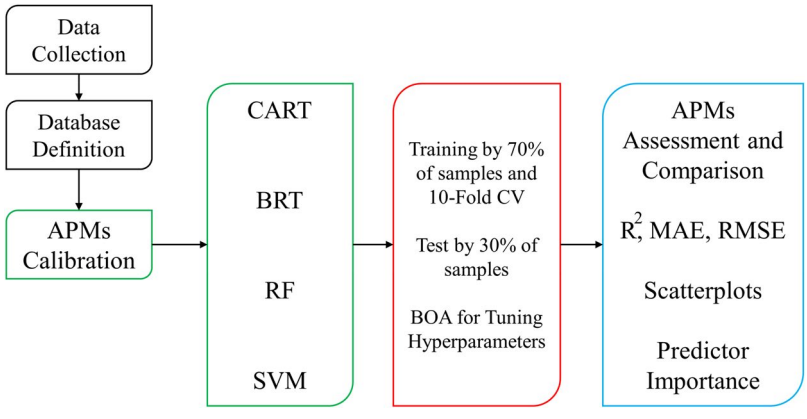

Figure 1. Workflow.

the function is computed by imposing different random values of the hyperparameters) to obtain posterior information of the function distribution. Accordingly, BOA searches for the best set of hyperparameters that maximises the function according to an MSE criterion. It is generally assumed that the Gaussian Process (Rasmussen & Williams, 2006) is suitable as the prior distribution of such a function. Some comprehensive tutorials on BOA implementation can be found in the literature (Fiorentini et al., 2020; Snoek et al., 2012; Wu et al., 2019). The BOA wraps up the process when it reaches the specified number of iterations. At each iteration, the 10-Fold CV is implemented. In this study, MLAs have been optimised by 30 iterations of the BOA. Accordingly, the training phase of each MLA exploits 270 folds for training and 30 folds for validation, ensuring that the best set of hyperparameters has been discovered.

As a last modelling step concerning MLAs with a tree-based structure, it is possible to evaluate the importance of each input feature in predicting the target output. The so-called Predictor Importance (PI) is computed; PI quantifies the level of correlation between the target output and each input factor. In the present research, PI could be of interest to analysts for understanding the potential implication of road-related characteristics of FI crash occurrence. As a general rule, a high PI I corresponds to a high correlation between an input and the target; however, the PI is also related to the splitting process of the RP algorithm. Indeed, all inputs chosen by RP at the first levels of a tree gain much PI than inputs chosen at the last levels. Therefore, the PI values should be assessed carefully and qualitatively. The math relation for computing the PI of an independent input feature X_m is reported in Equation 1 (Breiman, 2001).

$$PI(X_m) = \frac{1}{N_T} \sum_{T} \sum_{t \in T: v(s_t) = X_m} p(t) \Delta i(s_t, t) \tag{1}$$

Where N_T is the number of trees composing the tree-based algorithm (for CART, $N_T = 1$), $p(t)\Delta i(s_t,t)$ is the weighted impurity decreases, in which the weight $p(t)$ is the proportion n_t/n of observations falling in the node t , and $v(s_t)$ is the independent variable used in split s_t . In the present case, $v(s_t) = X_m$. Summations are extended to all the trees T composing the algorithm and to all t nodes. Readers may find comprehensive tutorials on how to compute $PI(X_m)$ in the research of Breiman and Louppe (Breiman, 2001; Louppe et al., 2013).

Model assessment

MLAs have been evaluated by some numerical and graphical performance metrics. As for the former typology, we exploited three well-known and widely-used metrics, as suggested by the literature quoted in the Introduction Chapter. Specifically, we computed R^2, RMSE, and MAE. They can be computed by the following Equations 2–4:

$$R^2 = 1 - \frac{\sum_{i=1}^{N} \left(Y_{i_{predicted}} - Y_{i_{observed}} \right)^2}{\sum_{i=1}^{N} \left(Y_{i_{observed}} - \overline{Y}_{observed} \right)^2} \tag{2}$$

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^{N} \left[\left(Y_{i_{predicted}} - Y_{i_{observed}} \right) \right]^2} \tag{3}$$

$$MAE = \frac{1}{N} \sum_{i=1}^{N} \left| Y_{i_{predicted}} - Y_{i_{observed}} \right| \tag{4}$$

Where $Y_{i_{predicted}}$ is the predicted value of the target output Y for the i -th element, $Y_{i_{observed}}$ is the observed value of the target output Y for the i -th element, and $\overline{Y}_{observed}$ is the averaged observed value of the target output Y . The coefficient R^2 identifies the amount of variance that a MLA can explain. RMSE quantifies the standard deviation of the residual distribution (a residual is defined as $e_i = Y_{i_{predicted}} - Y_{i_{observed}}$). Therefore, RMSE shows the accuracy of the predictions. Finally, as the name suggests, the MAE parameter provides a measure of the average magnitude of the residuals. Also, the R^2, RMSE, and MAE parameters allow comparing MLAs.

For what concerns the graphical performance metrics, we considered the scatterplots, i.e., graphs representing the relation between the actual values (observed crashes) and predicted targets (predicted crashes by MLAs). Therefore, scatterplots allow for identifying potential clustering of the points and outliers. Furthermore, scatterplots can recognise if the training and testing phases have been performed correctly and if MLAs suffer from overfitting issues.

Study area and data

Input features

The present research focuses on two-lane roads managed by the TRRA. The analysed network comprises all the roads managed by the TRRA, with an extension of about 905 km. Highway Safety Manual (American Association of State Highway and Transportation Officials (AASHTO),), 2010) introduces two-lane roads as single carriageway roads with one lane for each travel direction. In Italy, two-lane roads may cross rural or urban areas. From a geometrical point of view, two-lane roads in rural areas should present a lane width of 3.75 m, a paved shoulder of 1.50 m, a minimum planimetric radius with a curve of 118 m and a maximum longitudinal gradient of 7%, according to the Italian standards (Ministry of Infrastructure & Transport, 2001). On the other hand, two-lane roads in urban areas should present a lane width of 3 m, a paved shoulder of 0.50 m, a sidewalk of 1.50 m, a minimum planimetric radius with a curve of 50 m and a maximum longitudinal gradient of 8%.

Nonetheless, existing two-lane roads may have a significantly reduced cross-section. Figure 2 reports the study area (Italy and the Tuscany Region), the road network analysed, and the typical cross-section that a new two-lane road (both rural and urban) should have. Figure 2a has been adapted

from (Fiorentini et al., 2020), and Figure 2c has been adapted from (Ministry of Infrastructure & Transport, 2001).

The dataset is composed of 1,814 road sections concerning 22 two-lane roads. The input features are described in the following list;

1. Road Area Type (*AREA*): Road Area Type is a categorical variable assuming three values, namely *RURAL* , *SUBURBAN* , and *URBAN* . Table 2 reports the frequency of *AREA* within the investigated road network. The Italian Ministry of Infrastructure and Transport defines an urbanised area as a set of buildings marked along the road by the appropriate start and end signals. A set of buildings means a continuous grouping, even if interspersed with streets, squares, gardens, or similar, consisting of no less than twenty-five buildings and areas with vehicular or pedestrian access on the road (Ministry of Infrastructure & Transport, 1992). Therefore, in this paper, the term urban area is used to identify the road context of a two-lane road that crosses these areas. Conversely, the term rural areas (or rural road context) is used for road stretches that do not cross urban areas; suburban areas (or suburban road context) represent a transition between urban and rural areas and include the administrative boundaries of the urban area itself. Accordingly, the related road contexts were defined as urban, rural, or suburban.
2. Average Annual Daily Traffic (*AADT*). The *AADT* was detected yearly through 71 survey stations located on the analysed road network. To be inserted as an input feature within the MLAs, the *AADT* of each road element is equal to the average of the *AADT* s detected;
3. Average Carriageway Width (*WC*);
4. Average Longitudinal Gradient (*G*);
5. Driveway Density (*DD*);
6. Density of Intersections (*DI*);
7. Vertical Tortuosity Index (*VTI*), computed as in Equation 5

$$VTI_i = \frac{\sum_{j=1}^{r_v} \frac{L_j}{Rv_j}}{L_i} \tag{5}$$

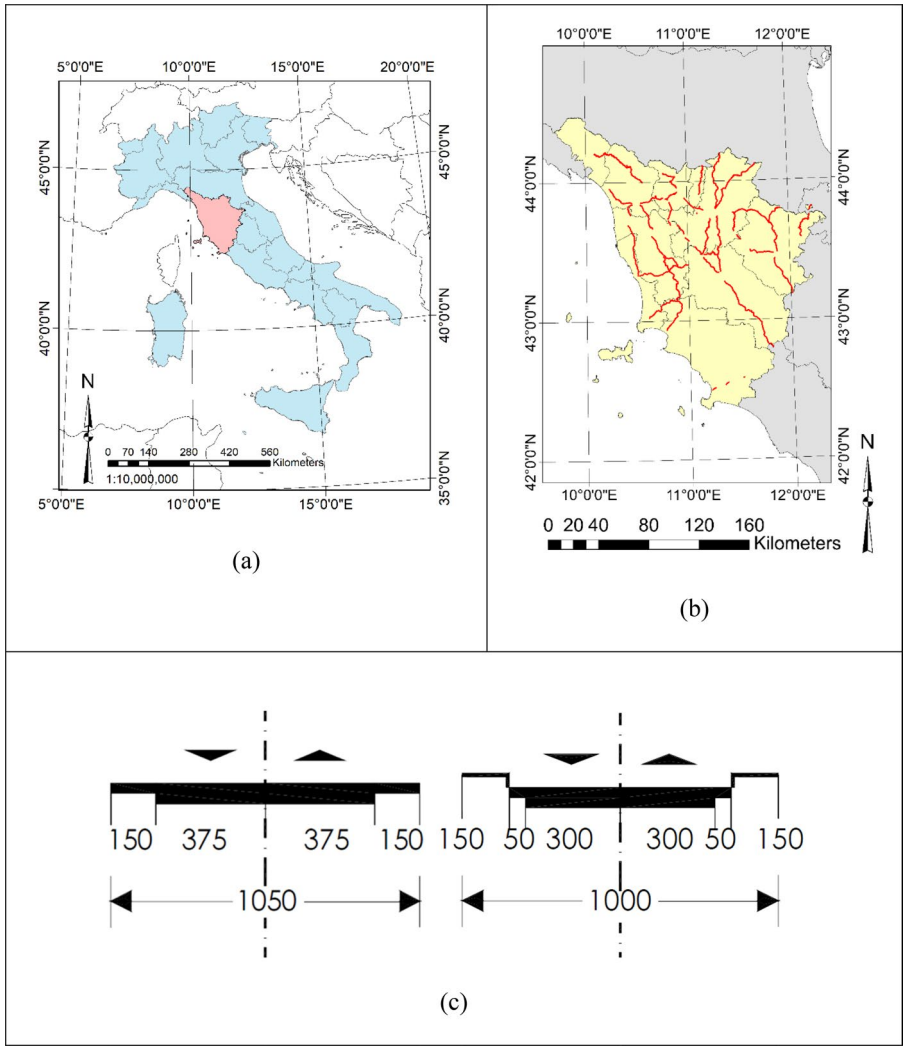

Figure 2. Study area and road network analysed: (a) Italy (light blue) and the Tuscany Region (pink); (b) Two-lane roads (solid red lines) crossing the Tuscany Region; (c) Cross-sections of new two-lane rural roads (left) and cross-section of new two-lane urban roads (right). Measures are expressed in terms of [cm].

Table 2. Road Area Type frequencies.

Area Type	Frequency
RURAL	1,146 (63.2%)
SUBURBAN	351 (19.3 %)
URBAN	317 (17.5%)

Where L_j is the length of the j-th segment in the i-th road section, Rv_j is the radius of the j-th vertical curve in the i-th road section, r_v is the number of vertical curves composing the i-th road section, and L_i is the length of the i-th road section;

8. Horizontal Tortuosity Index (*HTI*), computed as in Equation 6:

$$HTI_i = \frac{\sum_{j=1}^{r_p} \frac{L_j}{R_j}}{L_i} \quad (6)$$

Where L_j is the length of the j-th segment in the i-th road section, R_j is the radius of the j-th circular curve in the i-th road section, r_p is the number of circular curves composing the i-th road section, and L_i is the length of the i-th road section;

Table 3 provides the descriptive statistics of the input features.

It is worth mentioning that road elements have a fixed length, equal to 500 meters, according to the resolution of crash data provided by TRRA. A further explanation of such input features and definition of road elements can be found in (Fiorentini & Losa, 2020b), in which some MLAs able to identify road blackspots have been defined. Such Algorithms demonstrated a reliable relation between output response (crash likelihood susceptibility) and these input features. The Authors found an Accuracy of 70% in identifying critical road sites.

Output feature

The output feature of the MLAs is the number of FI crashes that may occur on a road element in the investigated 9-year period. Indeed, TRRA provided the crash history concerning FI crashes that occurred in the network analysed from 2008 to 2016. It is worth noting that Property Damage Only crashes have not been considered since they are neither registered by law enforcement nor considered by the Italian standards for road safety analyses on two-lane roads (Ministry of Infrastructure & Transport, 2012).

Accordingly, MLAs have been calibrated by an amount of 5,802 FI crashes. Such crashes caused 8,500 injuries and 162 deaths. In order to estimate the number of deaths or the number of injuries as a result of FI accidents, the prediction of MLAs can be multiplied by the average number of deaths per FI crash (0.028) or the average number of injuries per FI crash (1,465).

Results and discussion

Performance of accident prediction models

The Goodness-of-Fit and the predictive performance are evaluated. A first evaluation concerns the quality of MLAs through scatterplots. Figure 3 shows the scatterplots of both the training phase and test phase.

Some relevant aspects can be drawn from Figure 3. First, Figure 3a shows that the CART algorithm is not adequately trained and is unsuitable for learning the hidden patterns between features that lead to FI crashes. Indeed, it is possible to recognise several points arranged on horizontal lines. This aspect reflects that CART suffers from the approximation process through a piecewise-constant function. The test phase is aligned with the training phase, showing the CART weakness in providing reliable predictions. We may suppose that the low level of complexity of the CART does not allow identifying the relationships between input features and target output.

Figures 3b and c show similar behaviour for the BRT and RF algorithms; indeed, both algorithms express an appropriate training phase, in which the observation-prediction pairs align on the 45-degree line. Road elements with a high number of observed FI crashes tend to be slightly underestimated by the algorithms. The RF seems to have better Goodness-of-Fit than the BRT, probably because in RF, several uncorrelated CARTs work parallelly, and each has the same importance in the final prediction. On the contrary, in BRT, the final prediction is most related to the first trained CART of the ensemble that, as in Figure 3a, could suffer from a piecewise-constant function approximation. The learning rate has a meaningful impact in the training phase, but it does not fully alleviate such an issue. The test phase of both models shows that points are more scattered and not well aligned on the 45-degree line. Consequently, we expect the BRT and RF to suffer from overfitting issues during the training phase, which is reflected in a conspicuous decrease in performance in the test phase. Accordingly, the prediction of such algorithms on new data should be evaluated carefully.

Finally, as regards Figure 3d, it is noted that the SVM shows similar scatterplots for both the training and test phase. During training, points are more dispersed than in the BRT and RF algorithms; nonetheless, SVM shows similar

Table 3. Descriptive statistics of the input features.

	AADT	W_c	*G*	*HTI*	*VTI*	*DD*	*DI*
Statistic	[vehicles/day]	[m]	[%]	[1/km]	[1/km]	[accesses/km]	[intersections/km]
Mean	6,631.92	6.57	1.48	0.0057	0.0007	13.85	2.09
Std. Dev.	4,981.06	.70	2.10	0.0064	0.0008	16.06	2.82
Minimum	565	4.64	.15	0	0	0	0
Maximum	29,164	9.68	15.63	0.097	0.015	85.89	15.90

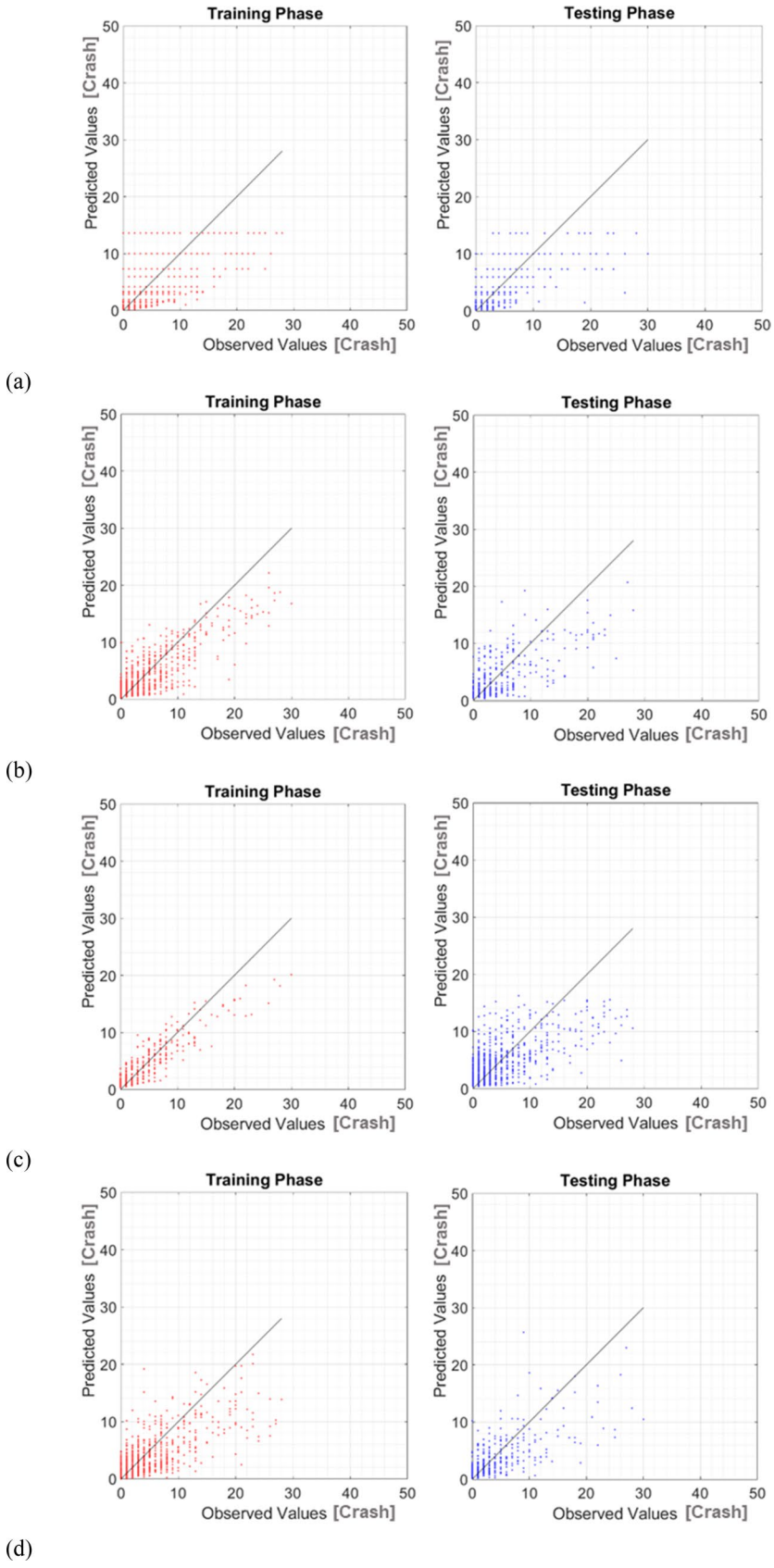

Figure 3. Scatterplots: (a) CART; (b) BRT; (c) RF; (d) SVM.

behaviour in the test phase, revealing no drops in performance on new data. Consequently, we can suppose that the SVM does not suffer from overfitting issues and can identify the hidden patterns between inputs and the target output. Therefore, the SVM could be considered an appropriate algorithm for making predictions on new data.

To support the above considerations and as a statistical evaluation, we can compute some numerical metrics that

allow comparing MLAs concisely. Indeed, such metrics should report the qualitative outcomes of Figure 3 in an aggregate and quantitative manner of the performance of MLAs. Therefore, Table 4 reports the R^2, RMSE, and MAE for both the training and test phase of the MLAs.

Table 4 demonstrates that the performance of MLAs is comparable. The CART reports the lowest performance for both phases and all metrics. However, it is worth saying that the performances are not significantly different from SVM in the training phase and all the MLAs in the test phase. This aspect shows the significant importance of scatterplots, demonstrating that CART predictions are not comparable with those of the other MLAs. Consequently, it is generally beneficial to evaluate both numerical and graphical metrics.

Table 4 emphasises the issues of overfitting in BRT and RF by the significant decrease in performance of all parameters. In RF, for instance, the parameter R^2 moves from 0.859 in the training phase to 0.552 in the test phase, while the RMSE and MAE nearly double their value. The same considerations can be extended to the BRT. Finally, Table 4 shows that the SVM performance is similar in both phases, with a slight drop in the transition between training and test. It is confirmed that the SVM does not suffer from overfitting and that the numerical metrics of the test phase appropriately represent its predictive performance. Therefore, a prediction made by the SVM results in an average error of about 2.07 FI crashes, while the residual distribution has a standard deviation of about 3.41 FI crashes. It is worth mentioning that most of this error is made in predictions on road elements with a high number of FI crashes. Conversely, predictions should be more accurate on road elements with fewer FI crashes, as shown in Figure 3d.

As the third piece of information derived from the calibrated algorithms, tree-based MLAs provide the importance of each input feature in predicting the target output. Figure 4 reports the computation of the Predictor Importance parameter through histograms.

Figure 4 reports valuable information, which can be summarised as follows. The parameter with the most significant impact on FI crash occurrence is *AADT*. This aspect results in all MLAs. It is worth mentioning that no specified functional form between inputs and output and no statistical distribution is associated with the target output; consequently, the models can identify that *AADT* plays a fundamental role in the phenomenon examined, as widely suggested in the literature. The *DD* parameter assumes a significant effect in all models, especially in the CART and BRT, in which the parameter takes the second position in importance. Furthermore, the *AREA* parameter assumes substantial influence in all MLAs (third position in CART and BRT, fourth position in RF), confirming the importance of considering the context of the road element.

We can also appreciate how the Predictor Importance values identified by CART and BRT are very similar, confirming that the training process of the BRT is strongly linked to the first trained CART of the ensemble. Considering the performance of these algorithms (Figure 4 and Table 4), we should expect that the Predictor Importance values computed with the RF algorithm are the most reliable among MLAs. Unlike CART and BRT, the RF also recognised the *DI* as an additional significant predictor (second position in importance). Finally, all MLAs provide a marginal influence of *VTI* and *HTI*, recognising that these parameters have a limited impact on the target output. No consideration can be derived for the *L* parameter since it assumes a constant value among road elements.

Comparing the performance and characteristics of MLAs, we recognise the SVM as the most appropriate MLA for predicting the number of FI crashes that may occur on Italian two-lane roads. Nonetheless, the advantages of the other MLAs are not negligible, such as the computation of predictor importance. Therefore, it is always a good modelling strategy to implement different MLAs to find the most appropriate one. Also, we demonstrated how using a comprehensive set of performance metrics is of great importance. Both graphical and numerical metrics should be computed and compared; as demonstrated, scatterplots revealed that BRT and RF suffer from overfitting issues while CART predictions are unsuitable. Numerical metrics allow comparing MLAs and concisely quantifying the performance, but they may hide specific weaknesses of the models.

Table 4. Performance metrics of Machine Learning Algorithms.

	Training Phase			Test Phase		
Model	R^2	RMSE	MAE	R^2	RMSE	MAE
CART	0.511	3.266	2.052	0.497	3.514	2.118
BRT	0.709	2.513	1.636	0.548	3.358	2.180
RF	0.859	1.735	1.050	0.552	3.380	2.235
SVM	0.561	3.119	1.914	0.510	3.417	2.074

Future works and research perspectives

There are some research directions to be pursued with future works. For instance, new data-driven models (such as Artificial Neural Networks) can be implemented; therefore, we can evaluate whether more sophisticated models lead to better predictive performance than the calibrated APMs. In addition, it could be interesting to compare the performance of calibrated APMs with traditional statistical regressions, developing local Safety Performance Functions (American Association of State Highway and Transportation Officials (AASHTO),), 2010). Based on statistical count models such as the Negative Binomial Regression, these models allow obtaining a non-black-box solution, showing how predictors are related to each other and the crash count. Furthermore, these models allow quantifying the impact of geometric and functional predictors on crash likelihood. Consequently, there would be the opportunity for a quantitative, and not only qualitative, assessment of the importance of predictors. Indeed, we believe that even basic research, endorsed by considerable scientific experiences, is significant. It would verify whether innovative data-driven models can effectively identify which parameters are more related to accidents and whether these models have higher predictive performance than traditional statistical models.

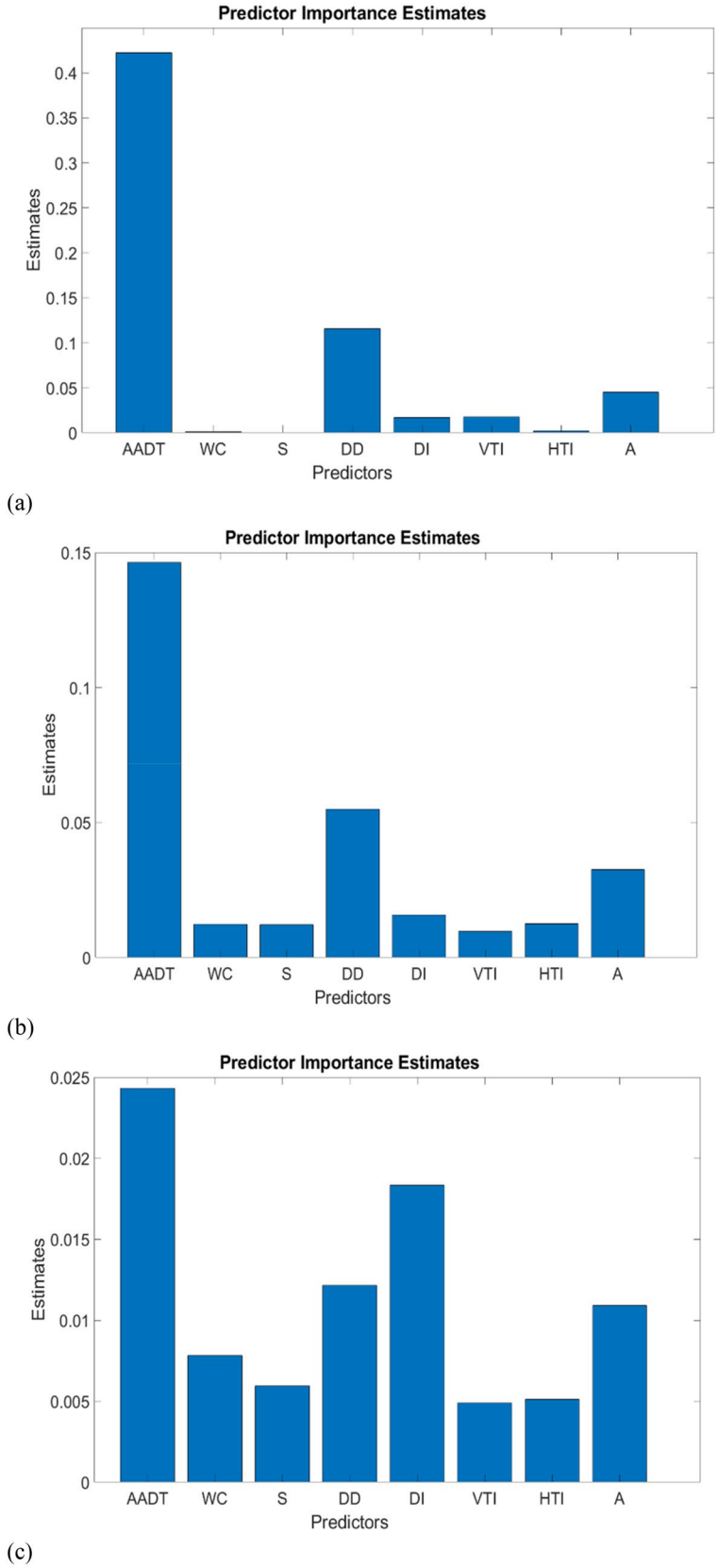

Figure 4. Predictor Importance: (a) CART; (b) BRT; (c) RF.

Conclusions

This paper focused on the calibration, evaluation, and comparison of four Accident Prediction Models for predicting Fatal and Injury crashes on Italian two-lane roads, implementing some Machine Learning Algorithms. Considering geometrical and functional predictors and incorporating the type of area crossed by roads (i.e., urban, suburban, and

rural), the models were calibrated with 1,814 road sections and evaluated by numerical and graphical metrics. We have recognised that tree-based Machine Learning Algorithms are suitable for identifying the most impacting explanatory variables, but they may suffer from overfitting issues during the training phase. In the context of the analysed network, we have shown that the Support Vector Machine is the most appropriate algorithm for predicting Fatal and Injury crashes. Moreover, by computing the Predictor Importance, we demonstrate that traffic flow, the density of intersections, driveway density, and the type of area are the most impacting infrastructure-related features on crash likelihood. These results can support the decision-making processes of road authorities regarding road planning, road monitoring, and road safety level evaluation.

Author contribution

Nicholas Fiorentini: Conceptualisation, Data Curation, Formal Analysis, Investigation, Methodology, Resources, Software, Validation, Visualization, Writing Original Draft, Review & Editing. Pietro Leandri: Conceptualisation, Methodology, Supervision, Validation, Review & Editing. Massimo Losa: Conceptualisation, Methodology, Supervision, Investigation, Validation, Review & Editing, Project Administration, Funding Acquisition.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This work is supported by the Tuscany Region CMRSS (Regional Road Safety Monitoring Centre) project promoted by the Tuscany Region as partial fulfilment of the National Plan of Road Safety (PNSS), funded by the Ministry of Infrastructure and Sustainable Mobility, Italy.

ORCID

Nicholas Fiorentini http://orcid.org/0000-0002-8769-8610
Pietro Leandri http://orcid.org/0000-0002-9032-7749
Massimo Losa http://orcid.org/0000-0001-7354-8534

References

Abdel-Aty, M., & Haleem, K. (2011). Analyzing angle crashes at unsignalized intersections using machine learning techniques. *Accident; Analysis and Prevention*, *43*(1), 461–470. https://doi.org/10.1016/j.aap.2010.10.002

Abdelwahab, H. T., & Abdel-Aty, M. A. (2001). Artificial neural networks and logit models for traffic safety analysis of toll plazas. *Transportation Research Record*, 1784(1), 115–125.

Abellán, J., López, G., & De Oña, J. (2013). Analysis of traffic accident severity using decision rules via decision trees. *Expert Systems with Applications*, *40*(15), 6047–6054. https://doi.org/10.1016/j.eswa.2013.05.027

Aizerman, M. A., Braverman, E. A., & Rozonoer, L. (1964). Theoretical foundations of the potential function method in pattern recognition learning. *Automation and Remote Control*, *25*, 821–837.

Al Mamlook, R. E., Kwayu, K. M., Alkasisbeh, M. R., & Frefer, A. A. (2019, April). Comparison of machine learning algorithms for predicting traffic accident severity. In *2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology, JEEIT 2019 - Proceedings* (pp. 272––276). https://doi.org/10.1109/JEEIT.2019.8717393

Alkheder, S., Taamneh, M., & Taamneh, S. (2017). Severity prediction of traffic accident using an artificial neural network. *Journal of Forecasting*, *36*(1), 100–108. https://doi.org/10.1002/for.2425

American Association of State Highway and Transportation Officials (AASHTO). (2010). *Highway safety manual* (1st ed.).

Basso, F., Basso, L. J., Bravo, F., & Pezoa, R. (2018). Real-time crash prediction in an urban expressway using disaggregated data. *Transportation Research Part C: Emerging Technologies*, *86*, 202–219. https://doi.org/10.1016/j.trc.2017.11.014

Breiman, L. (2001). Random forests. *Machine Learning*, *45*(1), 5–32. https://doi.org/10.1023/A:1010933404324

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). *Classification and regression trees* (2nd ed.). Chapman & Hall/CRC, Taylor & Francis Group.

Chang, L. Y., & Wang, H. W. (2006). Analysis of traffic injury severity: An application of non-parametric classification tree techniques. *Accident; Analysis and Prevention*, *38*(5), 1019–1027. https://doi.org/10.1016/j.aap.2006.04.009

Chen, C., Zhang, G., Qian, Z., Tarefder, R. A., & Tian, Z. (2016). Investigating driver injury severity patterns in rollover crashes using support vector machine models. *Accident; Analysis and Prevention*, *90*, 128–139. https://doi.org/10.1016/j.aap.2016.02.011

Delen, D., Sharda, R., & Bessonov, M. (2006). Identifying significant predictors of injury severity in traffic accidents using a series of artificial neural networks. *Accident; Analysis and Prevention*, *38*(3), 434–444. https://doi.org/10.1016/j.aap.2005.06.024

Delen, D., Tomak, L., Topuz, K., & Eryarsoy, E. (2017). Investigating injury severity risk factors in automobile crashes with predictive analytics and sensitivity analysis methods. *Journal of Transport & Health*, *4*, 118–131. https://doi.org/10.1016/j.jth.2017.01.009

Dogru, N., & Subasi, A. (2018). *Traffic accident detection using random forest classifier* [Paper presentation]. 2018 15th Learning and Technology Conference, L and T 2018 (pp. 40–45). https://doi.org/10.1109/LT.2018.8368509

Elith, J., Leathwick, J. R., & Hastie, T. (2008). A working guide to boosted regression trees. *The Journal of Animal Ecology*, *77*(4), 802–813. https://doi.org/10.1111/j.1365-2656.2008.01390.x

ESRI. (2011). *ArcGIS Desktop: Release 10.5*. Environmental Systems Research Institute.

Fiorentini, N., & Losa, M. (2020a). Handling imbalanced data in road crash severity prediction by machine learning algorithms. *Infrastructures*, *5*(7), 61. https://doi.org/10.3390/infrastructures5070061

Fiorentini, N., & Losa, M. (2020b). Long-term-based road blackspot screening procedures by machine learning algorithms. *Sustainability*, *12*(15), 5972. https://doi.org/10.3390/su12155972

Fiorentini, N., Maboudi, M., Leandri, P., Losa, M., & Gerke, M. (2020). Surface motion prediction and mapping for road infrastructures management by PS-InSAR measurements and machine learning algorithms. *Remote Sensing*, *12*(23), 3976. https://doi.org/10.3390/rs12233976

Hossain, M., & Muromachi, Y. (2012). A Bayesian network based framework for real-time crash prediction on the basic freeway segments of urban expressways. *Accident; Analysis and Prevention*, *45*, 373–381. https://doi.org/10.1016/j.aap.2011.08.004

Ijaz, M., Ian, L., Zahid, M., & Jamal, A. (2021). A comparative study of machine learning classifiers for injury severity prediction of crashes involving three-wheeled motorized rickshaw. *Accident; Analysis and Prevention*, *154*(February), 106094. https://doi.org/10.1016/j.aap.2021.106094

Iranitalab, A., & Khattak, A. (2017). Comparison of four statistical and machine learning methods for crash severity prediction. *Accident; Analysis and Prevention*, *108*, 27–36. https://doi.org/10.1016/j.aap.2017.08.008

Jamal, A., & Umer, W. (2020). Exploring the injury severity risk factors in fatal crashes with neural network. *International Journal of Environmental Research and Public Health*, *17*(20), 7466–7422. https://doi.org/10.3390/ijerph17207466

Jamal, A., Zahid, M., Tauhidur Rahman, M., Al-Ahmadi, H. M., Almoshaogeh, M., Farooq, D., & Ahmad, M. (2021). Injury severity prediction of traffic crashes with ensemble machine learning techniques: A comparative study. *International Journal of Injury Control and Safety Promotion*, *28*(4), 408–420. https://doi.org/10.1080/17457300.2021.1928233

Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In Ijcai (Vol. 14, pp. 1137–1145).

Li, P., Abdel-Aty, M., & Yuan, J. (2020). Real-time crash risk prediction on arterials based on LSTM-CNN. *Accident Analysis & Prevention*, *135*, 105371. https://doi.org/10.1016/j.aap.2019.105371

Li, Z., Liu, P., Wang, W., & Xu, C. (2012). Using support vector machine models for crash injury severity analysis. *Accident; Analysis and Prevention*, *45*, 478–486. https://doi.org/10.1016/j.aap.2011.08.016

Loh, W.-Y., & Shih, Y.-S. (1997). Split selection methods for classification trees. *Statistica Sinica*, *7*, 815–840.

Louppe, G., Wehenkel, L., Sutera, A., & Geurts, P. (2013). Understanding variable importances in forests of randomized trees. In *Advances in Neural Information Processing Systems*.

Lv, Y., Tang, S., Zhao, H., & Li, S. (2009). *Real-time highway accident prediction based on support vector machines* [Paper presentation]. 2009 Chinese Control and Decision Conference, CCDC 2009 (pp. 4403–4407). https://doi.org/10.1109/CCDC.2009.5192409

Martin, P., David, E. G., & Erick, C.-P. (1999). BOA: The Bayesian optimization algorithm. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)* (pp. 525–532).

Micchelli, C. A. (1986). Algebraic aspects of interpolation. In *Proceedings of Symposia in Applied Mathematics* (pp. 81–102). https://doi.org/10.1090/psapm/036/864367

Ministry of Infrastructure and Transport. (1992). *Decreto Legislativo N. 285 del 30/04/1992, Nuovo Codice della Strada*.

Ministry of Infrastructure and Transport. (2001). *Decreto Ministeriale 5 novembre 2001, n. 6792 (S.O. n.5 alla G.U. n.3. del 4.1.02) Norme funzionali e geometriche per la costruzione delle strade*.

Ministry of Infrastructure and Transport. (2012). *Decreto Ministeriale numero 137 del 02/05/2012, Linee guida per la gestione della sicurezza delle infrastrutture stradali ai sensi dell'art. 8 del decreto legislativo 15 marzo 2011, n. 35*.

Mokhtarimousavi, S., Anderson, J. C., Azizinamini, A., & Hadi, M. (2019). Improved support vector machine models for work zone crash injury severity prediction and analysis. *Transportation Research Record: Journal of the Transportation Research Board*, *2673*(11), 680–692. https://doi.org/10.1177/0361198119845899

Rahim, M. A., & Hassan, H. M. (2021). A deep learning based traffic crash severity prediction framework. *Accident; Analysis and Prevention*, *154*(September 2020), 106090. https://doi.org/10.1016/j.aap.2021.106090

Rasmussen, C. E., & Williams, C. K. I. (2006). Gaussian processes for machine learning. *International Journal of Neural Systems*, *103*(14), 429. https://doi.org/10.7551/mitpress/3206.001.0001

Schapire, R. E. (1999). A brief introduction to boosting. In *Ijcai* (pp. 1401–1406).

Schapire, R. E. (2002). The boosting approach to machine learning an overview. In *MSRI Workshop on Nonlinear Estimation and Classification*.

Schlögl, M., Stütz, R., Laaha, G., & Melcher, M. (2019). A comparison of statistical learning methods for deriving determining factors of accident occurrence from an imbalanced high resolution dataset. *Accident; Analysis and Prevention*, *127*, 134–149. https://doi.org/10.1016/j.aap.2019.02.008

Singh, G., Pal, M., Yadav, Y., & Singla, T. (2020). Deep neural network-based predictive modeling of road accidents. *Neural Computing and Applications*, *32*(16), 12417–12426. https://doi.org/10.1007/s00521-019-04695-8

Singh, G., Sachdeva, S. N., & Pal, M. (2016). M5 model tree based predictive modeling of road accidents on non-urban sections of highways in India. *Accident; Analysis and Prevention*, *96*, 108–117. https://doi.org/10.1016/j.aap.2016.08.004

Singh, G., Sachdeva, S. N., & Pal, M. (2018). Support vector machine model for prediction of accidents on non-urban sections of highways. *Proceedings of the Institution of Civil Engineers - Transport*, *171*(5), 253–263. https://doi.org/10.1680/jtran.17.00019

Smola, A. J., Schölkopf, B., & Schölkopf, S. (2004). A tutorial on support vector regression. *Statistics and computing 14*, 199–222. https://doi.org/10.1023/B:STCO.0000035301.49549.88

Smola, A. (1996). *Regression estimation with support vector learning machines* [Master's thesis]. Technische Universit at Munchen.

Snoek, J., Larochelle, H., & Adams, R. P. (2012). Practical Bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems*.

The MathWorks Inc. (2020). MATLAB, Version R2020b. Natick, Massachusetts; 2010.

Theofilatos, A., Chen, C., & Antoniou, C. (2019). Comparing machine learning and deep learning methods for real-time crash prediction. *Transportation Research Record: Journal of the Transportation Research Board*, *2673*(8), 169–178. https://doi.org/10.1177/0361198119841571

Vapnik, V., Golowich, S. E., & Smola, A. (1997). Support vector method for function approximation, regression estimation, and signal processing. In D. Touretzky, M. Mozer, &. M. Hasselmo (Eds.), *Advances in neural information processing systems* (pp. 281–287). MIT Press.

Vapnik, V., & Lerner, A. (1963). Pattern recognition using generalized portrait method. *Automation and Remote Control*, *24*, 774–780.

Wahab, L., & Jiang, H. (2020). Severity prediction of motorcycle crashes with machine learning methods. *International Journal of Crashworthiness*, *25*(5), 485–492. https://doi.org/10.1080/13588265.2019.1616885

Wang, J., Kong, Y., & Fu, T. (2019). Expressway crash risk prediction using back propagation neural network: A brief investigation on safety resilience. *Accident; Analysis and Prevention*, *124*, 180–192. https://doi.org/10.1016/j.aap.2019.01.007

World Health Organization. (2018). *Global status report on road safety 2018*.

Wu, J., Chen, X. Y., Zhang, H., Xiong, L. D., Lei, H., & Deng, S. H. (2019). Hyperparameter optimization for machine learning models based on Bayesian optimization. *Journal of Electronic Science and Technology*, *17*(1), 26–40. https://doi.org/10.11989/JEST.1674-862X.80904120

Xiao, J. (2019). SVM and KNN ensemble learning for traffic incident detection. *Physica A: Statistical Mechanics and Its Applications*, *517*, 29–35. https://doi.org/10.1016/j.physa.2018.10.060

Xiao, J., & Liu, Y. (2012). *Traffic incident detection by multiple kernel support vector machine ensemble* [Paper presentation]. IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC (pp. 1669–1673). https://doi.org/10.1109/ITSC.2012.6338751

You, J., Wang, J., & Guo, J. (2017). Real-time crash prediction on freeways using data mining and emerging techniques. *Journal of Modern Transportation*, *25*(2), 116–123. https://doi.org/10.1007/s40534-017-0129-7

Yuan, J., Abdel-Aty, M. A., Gong, Y., & Cai, Q. (2019). Real-time crash risk prediction using long short-term memory recurrent neural network. *Transportation Research Record: Journal of the Transportation Research Board*, *2673*(4), 314–326. https://doi.org/10.1177/0361198119840611

Zhang, J., Li, Z., Pu, Z., & Xu, C. (2018). Comparing prediction performance for crash injury severity among various machine learning and statistical methods. *IEEE Access*, *6*, 60079–60087. https://doi.org/10.1109/ACCESS.2018.2874979

Zhang, J., & Yang, J. (1997). Instance-based learning for highway accident frequency prediction. *Computer-Aided Civil and Infrastructure Engineering*, *12*(4), 287–294. https://doi.org/10.1111/0885-9507.00064