Available online at www.sciencedirect.com

ScienceDirect

Transportation Research Procedia 52 (2021) 292–299

Transportation Research Procedia

www.elsevier.com/locate/procedia

23rd EURO Working Group on Transportation Meeting, EWGT 2020, 16-18 September 2020, Paphos, Cyprus

Real-time conflict prediction: a comparative study of machine learning classifiers

Federico Orsini[a,*], Gregorio Gecchele[b], Massimiliano Gastaldi[a,c], Riccardo Rossi[a]

[a]*University of Padua, Department of Civil, Environmental and Architectural Engineering, Via F. Marzolo 9, 35131 Padua, Italy*
[b]*Atraki s.r.l., Via A. Diaz 4, 37015 S. Ambrogio di Valpolicella (Verona), Italy*
[c]*University of Padua, Department of General Psychology, Via Venezia 8, 35131 Padua, Italy*

Abstract

Real-time conflict prediction models (RTConfPM) are an innovative approach to deal with real-time road safety analysis, even in absence of reliable crash data. This paper presents a RTConfPM to predict rear-end conflicts, using Time-To-Collision (TTC) values recorded with radar sensors on multiple motorway cross-sections to define unsafe situations, and traffic conditions recorded on the same sections as inputs to the model. Several classifiers were trained and compared: K-Nearest Neighbors (KNN), Naïve Bayes (NB), Discriminant Analysis (DA), Decision Trees (DT) and Support Vector Machine (SVM). All the proposed models provide better performance than most of crash-based real-time models found in the literature, with KNN and SVM significantly better than the others when considering Recall indicator.

Keywords: Real-time crash prediction; traffic conflicts; road safety; machine learning; rear-end crashes

1. Introduction

Real-time crash prediction models (RTCPM) aim at predicting the occurrence of a crash on a specific road section within a very short time window, using the instantaneous traffic dynamics as input. Several real-time crash prediction models have been developed in recent years: a comprehensive review can be found in Hossain et al. (2019); however, these models cannot be applied if crash data are unavailable or unreliable. In particular, the quality of crash data, in terms of both spatial and temporal precision, must be very high in order to obtain reliable predictions.

An alternative approach consists in using conflicts, instead of crashes, in order to identify unsafe situations (Caleffi et al., 2017; Katrakazas et al., 2018). Conflicts are observable situations in which there is a risk of collisions if the movements of the vehicles involved remain unchanged; they are "precursors of crashes and not alternative outcomes" (Tarko, 2020). Thanks to developments in ITS and sensing technologies, conflicts can be detected with high accuracy and, since they are more frequent than crashes, with shorter observation times.

* Corresponding author. Tel.: +390498275572. *E-mail address:* federico.orsini@dicea.unipd.it

10.1016/j.trpro.2021.01.034

The aim of a real-time conflict prediction model (RTConfPM) is to predict the occurrence of conflicts, instead of crashes, within a short time window, based on current traffic, road and weather conditions. Taking actions in order to prevent the occurrence of conflicts would, in turn, prevent the occurrence of crashes. The link between conflicts and traffic variables can be stronger than the link between crashes and traffic variables, since some crashes happens due to reasons (e.g., driver's distraction), which may be uncorrelated to the traffic/weather conditions; as a consequence, predictive power of a RTConfPM could be better than that of a RTCPM (Orsini et al., 2020c).

Very few works dealt with RTConfPM from an infrastructure-based approach, i.e. collecting data on a cross-section and making prediction on a road segment. Katrakazas et al. (2018) compared some machine learning classifiers using data obtained from a micro-simulation study; Caleffi et al. (2017) applied linear discriminant analysis to predict conflicts on a Brazilian freeway section, using a qualitative approach to define conflicts; Orsini et al. (2020c) compared RTConfPM and RTCPM calibrated on the same real-world dataset, using Support Vector Machine. Other works dealt with this problem from the vehicle point-of-view, i.e. collecting data from a connected vehicle and making prediction on conflicts concerning that vehicle (Formosa et al., 2020; Osman et al., 2019).

To the best of our knowledge, this work is the first comparative study of machine learning classifiers for a RTConfPM based on real-world data. Moreover, there is a lack of such studies, in general, in real-time road safety literature, as underlined by Theofilatos et al. (2019).

2. Methodology

Similarly to a RTCPM, a RTConfPM is organized according to the following scheme: (i) the training dataset is formed considering unbalanced classification issues; (ii) a classifier is trained and tested; (iii) several performance indicators are used to evaluate the classifier. In this work, Synthetic Minority Oversampling TEcnhique (SMOTE) was used to balance the training dataset; several different basic machine learning classifiers (K-Nearest Neighbors, Naïve Bayes, Discriminant Analysis, Decision Trees and Support Vector Machine), were tested and compared; Recall, Specificity and AUC were used as main indicators to evaluate the models.

2.1. Unbalanced classification and resampling: SMOTE

In RTCPM and RTConfPM the two response classes (safe vs unsafe) are usually strongly unbalanced, with "safe" records outnumbering "unsafe" records by several orders of magnitude; this is also the case of the present case study (see Section 3.3). In unbalanced classification problems, models tend to classify all the samples into the major class, due to their objective functions, which minimize the sum of errors by assigning the same weight to both the major and minor classes (Sun et al., 2007). This issue is generally addressed in RTCPM literature by resampling the dataset, in order to obtain two balanced classes; this can be done with Synthetic Minority Oversampling TEcnhique (SMOTE), proposed by Chawla et al. (2002) and applied in several recent RTCPMs (Basso et al., 2018; Yuan et al., 2019). SMOTE generates synthetic samples of the minority class; these samples are points in the feature space located between neighboring original samples. SMOTE can be used in combination with random under-sampling of the majority class, in order to obtain the desired ratio between the classes (Chawla et al., 2002).

2.2. Machine learning classifiers

2.2.1. K-Nearest Neighbors

K-Nearest Neighbors (KNN) is one of the simplest classification methods in machine learning. The basic idea is that a new sample is classified by looking at the classes of the known samples closest to it. The two main parameters which have to be chosen are: k, i.e. the number of neighbor samples to be considered; the distance function (e.g. Euclidean). KNN can be summarized according to the following procedure:

1. The k-nearest neighbors to the new sample (according to the distance function) are selected;
2. For each class, it is counted how many of these k samples belong to it;
3. The new sample is classified into the class which includes the highest number of the k neighbors.

KNN has been applied in several RTCPM/RTConfPM (Katrakazas et al., 2018; Osman et al., 2019; Theofilatos et al., 2019).

2.2.2. Decision tree

Decision trees (DT) are another simple and commonly used classification method in machine learning. A tree is built by splitting the original full dataset (i.e. the root node) into two branches, then the process is repeated for the two new subsets, and so on, until the all the data of the subset at a node have the same target variable values, or when splitting no longer adds value to the predictions.

Tree growing is performed by recursively partitioning the target variable to maximize "purity" in the child node: indeed, the splits are performed so that each child node is more homogeneous than its parent node. The most commonly used split criterion in DTs is based on Gini's diversity index, calculated as:

$$Gini = 1 - \sum_{1}^{n} \rho_i^2 \tag{1}$$

Where i is the class (e.g., crash/non-crash), n the number of classes, ρ_i the percentage of observations in class i.

DT has been used in several RTCPMs (Hossain and Muromachi, 2011; Theofilatos et al., 2019), and often as a variable-selection method (Yu and Abdel-Aty, 2013).

2.2.3. Naïve Bayes

While KNN and DT/RF do not make any assumptions on the distribution of the underlying data, with Naïve Bayes (NB) technique, it is assumed that data come from a certain distribution, and therefore it is possible to treat data as a statistical sample. In NB classification it is assumed that observations in each response class are samples from probability distributions and, in particular, a separate distribution for each class. Knowing these probability distributions, it is possible to determine the probability of a new observation occurring at a given location in the feature space, under the assumption that it belongs to a specific class. Bayes theorem allows to "reverse" this, and to calculate the probability that the new observation comes from a given class:

$$P(y|x) = \frac{P(x|y)P(y)}{P(x)} \tag{2}$$

Where: $P(y|x)$ is the posterior probability of response class y, given predictor x; $P(x|y)$ the likelihood, i.e. the probability of the predictor given the class; $P(y)$ the prior probability of the class; $P(x)$ the prior probability of the predictor. The "naïve" assumption is that each predictor variable is independent in each response class, and this greatly simplifies the calculation, since performing a prediction only requires determining the probability of the observation and applying (2). Then, since it is assumed that observations are independent in each predictor, the probability is simply the product of the probabilities in each variable. In the end, given a new sample, it is possible to calculate the probability of it belonging to each class, and subsequently to classify it to the most likely.

A benefit of this approach is that probabilities give some indications on how "clear" the classification is. Moreover, since predictions are based on the statistical distribution of all the observations, rather than on individual observations, they are robust to noise in the training data and to outliers.

NB has been applied in a number of real-time road safety applications (Osman et al., 2019; Theofilatos et al., 2019).

2.2.4. Discriminant analysis

Similar to NB, in discriminant analysis (DA) it is assumed that observations in each prediction class can be modeled with a normal probability distribution. However, there is no assumption of independence in each predictor, and a multivariate normal (MVN) distribution is fitted to each class.

Training data is used to fit MVN distributions for each response class and, subsequently, to find the location of the boundary between these classes. The boundary between two classes is defined as the locus of points in the feature space where there is equal probability of belonging to either class, and it depends on the parameters of the MVN distributions (means and variance-covariance matrices). If homoscedasticity is assumed, then the boundary is linear; alternatively, it is quadratic. As NB, DA is robust to noise and to outliers.

DA has been applied in few RTCPM/RTConfPM works (Caleffi et al., 2017; Xu et al., 2012).

2.2.5. Support vector machine

The idea of looking for a boundary to separate classes is also part of support vector machine (SVM) approach, which however, contrarily to DA, does not make assumption on probability distributions of training data.

The aim of SVM is to find the hyperplane that separates all data points, maximizing the distance between the classes, i.e. the "margin". This results in a simple optimization problem: choosing the coefficient of the linear boundary to maximize the distance between the boundary and the nearest observations, subject to the constraint that all observations must be on the correct side of the boundary. In the end, the optimal solution is determined only by the observations nearest to the boundary, i.e. the "support vectors".

If, as it often happens, data from the two classes are not linearly separable, SVM can use a soft margin, i.e. a hyperplane that separates many (yet not all) data points; this is done by adding slack variables in the optimization problem, in order to penalize misclassification. Given a set of vectors x_j (j=1,…,n) with dimension d (i.e. the dimension of the feature space) and their respective classes y_j, where y_j can either be equal to 1 or -1 (binary classification problem) and Lagrange multiplier α_j, the optimization problem has the form:

$$\max_{\alpha} \sum_j \alpha_j - \frac{1}{2}\sum_j \sum_k \alpha_j \alpha_k y_j y_k K(x_j, x_k) \qquad s.t. \ \sum_j \alpha_j y_j = 0, \ 0 \le \alpha_j \le C, \ j = 1, \dots, n \qquad (3)$$

In (3), K is a kernel function, which in the linear case is simply $K(x_j, x_k) = x_j^T x_k$. However, some binary classification problems do not have a simple hyperplane as a useful separating criterion, and in that case non-linear kernel functions, such as polynomial or radial basis (RBF), can be employed.

SVM has been applied in many works dealing with real-time road safety (Wang et al., 2019; Yu and Abdel-Aty, 2013), sometimes in combination with SMOTE (Basso et al., 2018; Parsa et al., 2019).

2.3. Performance indicators

Model performance can then be evaluated using several indicators. Recall is the number of correctly predicted unsafe situations divided by the total number of observed unsafe situations. Specificity is the number of correctly predicted safe situations divided by the total number of observed safe situations. While Recall is a very meaningful indicator under a safety point-of-view, with Specificity it is possible to assess operational feasibility of the model, since it quantifies the impact of false positives. A useful tool for an overall evaluation of the model is the ROC curve, obtained by plotting the true positive rate (i.e., Recall) against the false positive rate (i.e. False Alarm Rate, 1 - Specificity) at various threshold settings. From this curve it is possible to derive a widely used performance indicator: AUC, i.e. the area under the ROC curve.

3. Data preparation

In a RTConfPM, input data (Section 3.1) collected during a given time interval are used to predict whether there will be an increased risk of having an unsafe situation (defined according to Section 3.2) in the following time interval.

3.1. Predictor variables: traffic data

Data were collected on a 3-lane Italian motorway for one year, from January 1st to December 31st, 2013 (Orsini et al., 2020a, 2020b). The motorway is about 150km long, and there were 20 cross-sections in each direction, 40 in total, equipped with micro-wave Doppler radars. The following vehicle-by-vehicle information was recorded: speed, time gap, time stamp and vehicle class. These data were then aggregated in 5-minutes intervals, providing the following variables (used as predictors in the classifiers), for each lane of each cross section: traffic volume, percentage of heavy-duty trucks, mean of vehicle speeds, standard deviation of vehicle speeds, therefore resulting in a total of 12 variables.

A variable selection procedure was applied on these data (the details are reported in Orsini et al. (2020c)), showing that all of them are relevant: removing one or more of them does not improve model performance; in fact, it may affect results quality.

3.2. Response classes: safe and unsafe time intervals

With the same disaggregated data, it was possible to calculate Time-To-Collision (TTC) between each pair of vehicles:

$$TTC = \frac{R}{RR} = \frac{v_L * t_{GAP}}{v_F - v_L} \quad (4)$$

where R (range) is the separation between the leading and the following vehicle, computed assuming constant speed of the vehicles, and RR (range rate) is the speed difference between the following and the leading vehicle. In (4), v_L and v_F are the speeds of the leading and the following vehicles, respectively; t_{GAP} is the time gap between them, recorded at the cross-section.

Therefore, for each 5-minute interval, a TTC value for each couple of consecutive vehicles on each lane of the cross-section is available, and, consequently, it is possible to determine whether one or more conflicts have happened or not in that interval. With this information, it is then possible to determine whether the 5-minute interval should be considered "safe" or "unsafe". In order to do so:

1. Conflicts should be identified, by setting an appropriate TTC threshold to separate normal vehicle interactions from real traffic conflicts.
2. A minimum number of conflicts within the 5-minute interval should be set.

The need to set a minimum number of conflicts to define a time interval as "unsafe" arises from the fact that conflicts, like crashes, may be caused not just by traffic characteristics (which are input variables to the RTConfPM), but also for example by human error, which is not and cannot be an input variable in the RTConfPM presented in this work. If a 5-minute interval is considered unsafe because a single conflict happened, we might include in the unsafe set situations which are not strictly unsafe under a traffic variables point of view. To overcome this issue and to consider as unsafe situations only those in which conflicts are caused, at least in part, by traffic conditions, a possible solution is to define as unsafe a 5-minute interval only if multiple conflicts happened.

In this case study, a 0.7 seconds TTC threshold and a minimum number of 3 conflicts were selected; in other words, in the RTConfPM a 5-minute interval was considered unsafe if at least 3 TTC values under 0.7 seconds were recorded during it. These values were chosen with a sensitivity analysis; for more details, the reader is referred to Orsini et al. (2020c).

3.3. Training and test datasets

Following the procedure described in Sections 3.1 and 3.2, unsafe situations were detected and matched with the traffic variables recorded on the same cross-section in the previous 5-minute interval; as a result, a dataset containing almost 3.5 million records was formed. Each record contained: (i) a unique ID, formed by the day of the year (1 to 365), the time interval of the day (1 to 288) and the cross-section (1 to 40); (ii) the corresponding values of the predictor variables; (iii) the response class, i.e. safe or unsafe. In accordance with the definition of "unsafe" situation presented in Section 3.2, 1,258 records were marked as unsafe, producing a strongly unbalanced dataset, with a 1:2,767 ratio between the two classes. In order to obtain a balanced dataset for model training, the following procedure (depicted in Figure 1) was followed:

1. The original full dataset was divided into a full training dataset and full test dataset (20% hold-out), keeping the 1:2,767 ratio between the classes.
2. The full training dataset was separated into two datasets containing exclusively unsafe/safe situations.
3. SMOTE (k=5) was applied to the unsafe situations dataset, producing a SMOTEd unsafe situations dataset with 1,258*5=6,290 records.
4. Another 6,290 records were randomly sampled from the safe situation dataset, and the remaining records were discarded.
5. SMOTEd unsafe situations and randomly sampled safe situations were joined in the SMOTEd dataset, which had a 1:1 ratio between the two classes.

6. The SMOTEd dataset was divided into a SMOTEd training dataset and a SMOTEd test dataset (20% hold-out), keeping the 1:1 ratio between the classes.

The SMOTEd training dataset was used for training the classifiers, whose performance was subsequently evaluated on a balanced test dataset (SMOTEd test dataset) and an unbalanced test dataset (full test dataset).

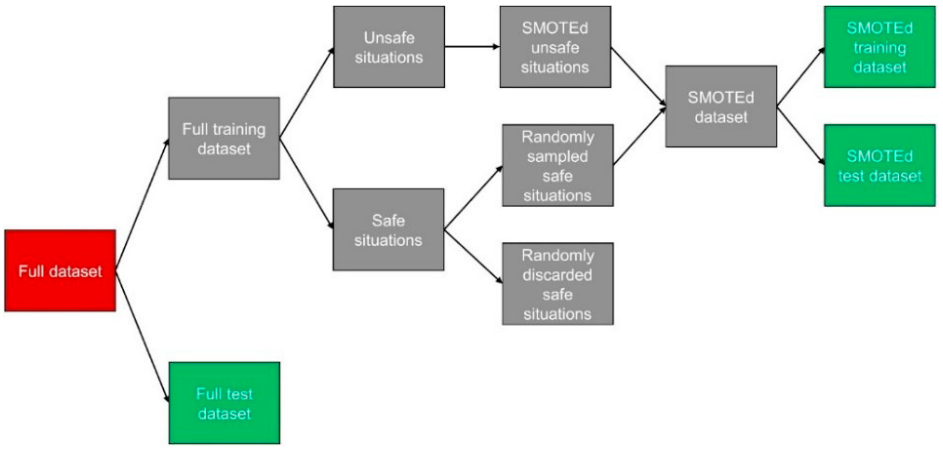

Figure 1. Overview of training, test and full-test dataset formation.

4. Results

The SMOTEd training dataset, obtained after data preparation procedures, was used to train several different classifiers: KNN, DT, DA, NB and SVM. As mentioned in Section 2, the training of each of these classifiers depends on some specific parameters of the classifier itself; therefore, each classifier was trained several times, using different combination of its parameters. For the sake of brevity, it was not feasible to report the results of all parameter combinations of each classifier; in this paper, only the results of the best performing combination are presented.

The following list contains the values of the parameters of the best combination, for each classifier.

- KNN: k=5; Euclidean distance function.
- DT: Gini split criterion.
- NB: normal distribution.
- DA: quadratic discriminant analysis.
- SVM: Linear kernel function, cost=1, gamma=1.

For robustness, Monte Carlo cross-validation with 100 repetitions was adopted. In practice, the 80%-20% split of SMOTEd dataset (see Section 3.3 and Figure 1) was repeated 100 times and each time classifiers were re-trained, and their performance evaluated. The values of Recall, Specificity and AUC presented in Tables 1 are the averages across the 100 repetitions.

The results reported in Table 1, show generally very high performance, compared to many RTCPM in the literature (Theofilatos et al., 2019; Yuan et al., 2019). Train and Test indicators are almost identical for all classifiers, except for DT, which suffers from a decrease (around 0.03-0.04) in all indicators, suggesting that there may be some slight overfitting problem.

The most relevant indicators, in order to evaluate the performance of the classifiers are those related to the Full-test dataset, i.e. the real-world unbalanced dataset. Here DT approach shows another decrease in all indicators, proving to be the least reliable of the five classifiers compared in this work. Also the NB approach is not completely satisfying, presenting the lowest Specificity of all classifiers, whereas DA overperforms it in all indicators. KNN and SVM results are in general very good, with excellent Recall and Specificity and AUC only lightly worse than DA.

Overall, considering that Recall is the most important indicator under a safety point of view, KNN and SVM appear to be the best classifiers, since they both are able to correctly predict more than 98% of the real-world unsafe situations;

on the other hand, DA, despite having slightly better Specificity and AUC, fails to predict more than 8% of unsafe situations (6% more than KNN and SVM).

In view of practical applications, it is important to have low False Alarm Rate (1-Specificity), because a system which triggers false alarms too often would be unfeasible to apply and/or would compromise the effectiveness of intervention strategies. Under this point of view, KNN and SVM provide good results, with rates lower than 7%; therefore, a false alarm is triggered only once out of 18~19 5-minutes time intervals.

Another important aspect to consider is consistency, i.e. how much the predicting performance depends on the specific partition of training/test datasets. In order to evaluate consistency, we analyzed the 95% confidence intervals (across the 100 repetition of the Monte Carlo cross-validation) of the indicators in the full-test dataset (see Figure 2). In particular, considering Recall indicator, it is possible to appreciate that both KNN and SVM not only have the best average value, but also the narrowest confidence interval, underlining once more that they are the best performing classifiers in this case study.

Table 1. Comparison of classifiers performance.

Dataset	Train			Test			Full-test		
Indicator	Recall	Specificity	AUC	Recall	Specificity	AUC	Recall	Specificity	AUC
KNN	0.994	0.945	0.999	0.993	0.936	0.970	0.983	0.936	0.962
DT	0.990	0.977	0.997	0.955	0.940	0.959	0.861	0.940	0.915
NB	0.930	0.909	0.968	0.929	0.909	0.968	0.903	0.909	0.961
DA	0.955	0.957	0.974	0.953	0.956	0.973	0.919	0.956	0.964
SVM	0.987	0.934	0.952	0.987	0.933	0.950	0.982	0.933	0.951

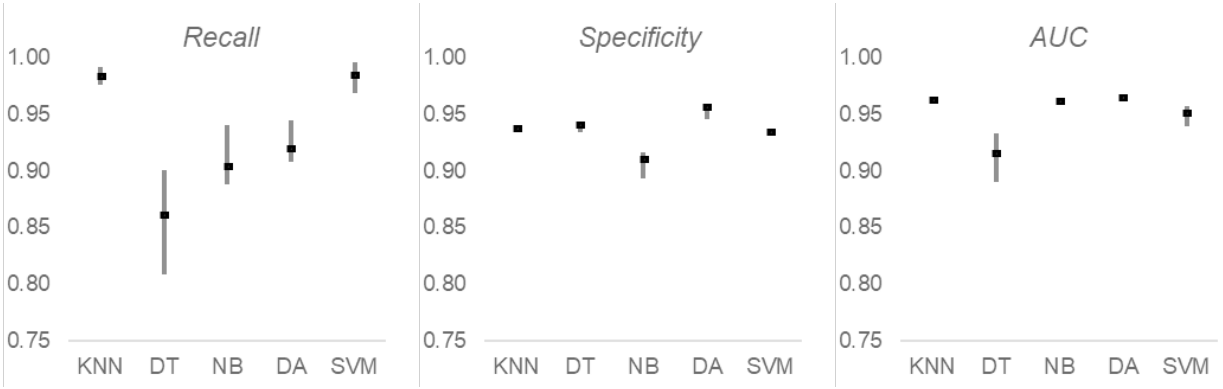

Figure 2. Full-test dataset. Average values (black square) and 95% confidence intervals (grey bars), for each classifier, across 100 repetitions of Monte Carlo cross-validation for: (a) Recall; (b) Specificity; (c) AUC.

5. Discussion and conclusion

This work compared several machine learning classifiers for real-time conflict prediction, by applying them for predicting rear-end unsafe situations in a real-world motorway case study. According to our findings, KNN and SVM appear to be the most effective, providing by far the best and most consistent values of Recall indicator, which is the most relevant under a safety point of view. Moreover, having a high Specificity (and therefore low False Alarm Rate) they show good potential for practical applications.

A possible explanation of why they are the best classifiers in this application is that they do not make assumptions on the probability distribution of training data, contrarily to NB and DA. In NB, it is assumed that each class (safe vs unsafe) has a different distribution and that each predictor variable is independent in each response class. DA, which relaxes the second assumption, provides better results, hinting that indeed their lower performance could be attributed

to these assumptions. DT, like KNN and SVM does not make assumption on the distribution of data; however, the DT model implemented in this case study appears to suffer from overfitting, which indeed is something that often affects this approach. A possible solution could be to apply an ensemble learning approach and consider multiple decision trees: one of the most popular of such approaches is Random Forest.

In the future, other types of approaches will be investigated: (i) statistical models, such as classic and Bayesian logistics regressions, which, in addition of providing yes-no predictions, could offer more insights on the effects of the various input variables; (ii) ensemble learning and deep learning approaches, which may be able to improve performance by exploiting more complex relations between data.

Acknowledgements

This work was supported by the University of Padua (“Advances in proactive and reactive approaches to road safety: development of innovative crash risk prediction and automatic incident detection systems"- BIRD177377/17).

References

Basso, F., Basso, L.J., Bravo, F., Pezoa, R., 2018. Real-time crash prediction in an urban expressway using disaggregated data. Transp. Res. Part C Emerg. Technol. 86, 202–219. https://doi.org/10.1016/j.trc.2017.11.014

Caleffi, F., Anzanello, M.J., Cybis, H.B.B., 2017. A multivariate-based conflict prediction model for a Brazilian freeway. Accid. Anal. Prev. 98, 295–302. https://doi.org/10.1016/j.aap.2016.10.025

Chawla, N. V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P., 2002. SMOTE: Synthetic minority over-sampling technique. J. Artif. Intell. Res. 16, 321–357. https://doi.org/10.1613/jair.953

Formosa, N., Quddus, M., Ison, S., Abdel-Aty, M., Yuan, J., 2020. Predicting real-time traffic conflicts using deep learning. Accid. Anal. Prev. 136, 105429. https://doi.org/10.1016/j.aap.2019.105429

Hossain, M., Abdel-Aty, M., Quddus, M.A., Muromachi, Y., Sadeek, S.N., 2019. Real-time crash prediction models: State-of-the-art, design pathways and ubiquitous requirements. Accid. Anal. Prev. 124, 66–84. https://doi.org/10.1016/j.aap.2018.12.022

Hossain, M., Muromachi, Y., 2011. Understanding Crash Mechanisms and Selecting Interventions to Mitigate Real-Time Hazards on Urban Expressways. Transp. Res. Rec. J. Transp. Res. Board 2213, 53–62. https://doi.org/10.3141/2213-08

Katrakazas, C., Quddus, M., Chen, W.H., 2018. A Simulation Study of Predicting Real-Time Conflict-Prone Traffic Conditions. IEEE Trans. Intell. Transp. Syst. 19, 3196–3207. https://doi.org/10.1109/TITS.2017.2769158

Orsini, F., Gecchele, G., Gastaldi, M., Rossi, R., 2020a. Large-scale road safety evaluation using extreme value theory. IET Intell. Transp. Syst. https://doi.org/10.1049/iet-its.2019.0633

Orsini, F., Gecchele, G., Gastaldi, M., Rossi, R., 2020b. Transferability and seasonality in extreme value theory applications to road safety: a case study in an Italian motorway. Adv. Transp. Stud. 2, 33–46.

Orsini, F., Gecchele, G., Rossi, R., Gastaldi, M., 2020c. A conflict-based approach for real-time road safety analysis. Under Rev.

Osman, O.A., Hajij, M., Bakhit, P.R., Ishak, S., 2019. Prediction of Near-Crashes from Observed Vehicle Kinematics using Machine Learning. Transp. Res. Rec. 2673, 463–473. https://doi.org/10.1177/0361198119862629

Parsa, A.B., Taghipour, H., Derrible, S., Mohammadian, A.K., 2019. Real-time accident detection: Coping with imbalanced data. Accid. Anal. Prev. 129, 202–210. https://doi.org/10.1016/j.aap.2019.05.014

Sun, Y., Kamel, M.S., Wong, A.K.C., Wang, Y., 2007. Cost-sensitive boosting for classification of imbalanced data. Pattern Recognit. 40, 3358–3378. https://doi.org/10.1016/j.patcog.2007.04.009

Tarko, A.P., 2020. Chapter 3 - Traffic conflicts as crash surrogates, in: Tarko, A.P. (Ed.), Measuring Road Safety Using Surrogate Events. Elsevier, pp. 31–45. https://doi.org/https://doi.org/10.1016/B978-0-12-810504-7.00003-3

Theofilatos, A., Chen, C., Antoniou, C., 2019. Comparing Machine Learning and Deep Learning Methods for Real-Time Crash Prediction. Transp. Res. Rec. 2673, 169–178.

Wang, L., Abdel-Aty, M., Lee, J., Shi, Q., 2019. Analysis of real-time crash risk for expressway ramps using traffic, geometric, trip generation, and socio-demographic predictors. Accid. Anal. Prev. 122, 378–384. https://doi.org/10.1016/j.aap.2017.06.003

Xu, C., Liu, P., Wang, W., Li, Z., 2012. Evaluation of the impacts of traffic states on crash risks on freeways. Accid. Anal. Prev. 47, 162–171. https://doi.org/10.1016/j.aap.2012.01.020

Yu, R., Abdel-Aty, M., 2013. Utilizing support vector machine in real-time crash risk evaluation. Accid. Anal. Prev. 51, 252–259. https://doi.org/10.1016/j.aap.2012.11.027

Yuan, J., Abdel-Aty, M., Gong, Y., Cai, Q., 2019. Real-Time Crash Risk Prediction using Long Short-Term Memory Recurrent Neural Network. Transp. Res. Rec. 2673, 314–326. https://doi.org/10.1177/0361198119840611