

# A Case Study of AutoML for Supervised Crash Severity Prediction

Juan S. Angarita-Zapata<sup>a</sup> and Gina Maestre-Gongora<sup>b</sup> and Jenny Fajardo Calderín<sup>c</sup>

<sup>a</sup>DeustoTech, Faculty of Engineering, University of Deusto, Bilbao, Spain, js.angarita@deusto.es

<sup>b</sup>Faculty of Engineering, Universidad Cooperativa de Colombia, Medellín, Colombia, gina.maestre@campusucc.edu.co

<sup>c</sup>DeustoTech, Faculty of Engineering, University of Deusto, Bilbao, Spain, fajardo.jenny@deusto.es

## Abstract

Traffic accidents are one of the leading causes of death around the world. One well-established strategy to deal with this public health issue is the design and deployment of road safety systems, which are in charge of predicting traffic crashes to promote safer roads. Increasing data availability has supported Machine learning (ML) to address the prediction of crashes and their severity. Transportation literature reports various methods for such purposes; however, there is no single method that achieves competitive results in all crash prediction problems. In this context, Automated machine learning (AutoML) arises as a suitable approach to automatically address the model selection problem in areas wherein specialized ML knowledge is not always available or affordable, such as road safety. AutoML has been successfully used in other areas; nevertheless, extensive analysis to determine their strengths and weaknesses has not been done in very diverse learning tasks, such as crash severity forecasting. Thus, this paper aims to examine to what extent AutoML can be competitive against ad hoc methods (Gradient Boosting, Gaussian Naive Bayes, k-Nearest Neighbors, Multilayer Perceptron, Random Forest) on crash severity prediction modeled from a supervised learning perspective. We test 3 state-of-the-art AutoML methods (Auto-Sklearn, TPOT, AutoGluon). Results show that AutoML can be considered a powerful approach to support the model selection problem in crash severity prediction.

**Keywords:** Crash severity prediction, Supervised learning, Automated machine learning, Computational intelligence.

## 1 Introduction

Traffic crashes are one of the leading causes of death and injuries worldwide, with approximately a total of 1.35 million deaths each year [5]. To deal with this public issue and its associated social and economic consequences, the 2030 Agenda for Sustainable Development introduces the goal of reducing by half the world number of deaths and injuries resulting from traffic accidents [4]. In this context, a suitable strategy to address traffic accidents is the design, development, and implementation of road safety systems within Intelligent Transportation Systems (ITS) [20]. Their primary purpose is predicting crashes and classify their severity to give policy-makers valuable information, which then can be used to improve traffic flows management and increase road safety.

Traditionally, statistical modeling techniques have been used to predict crashes and classify their severity [16, 21]. However, those approaches usually have limitations dealing with uncertainty and large volumes of data. Currently, the recent emergence of telecommunications technologies integrated into transportation infrastructure generates vast volumes of traffic data. This unprecedented data availability and growing computational capacities have incremented Machine learning (ML) to address the prediction of crashes and their severity. From a ML perspective, crash severity prediction is focused on building a predictive model using historical data to make predictions of accident severity based on new and unseen data [22].

Transportation literature reports a great variety of ML methods that can be used for crash severity prediction [13, 22]. However, despite such availability of methods, there is no unique ML algorithm that can be competitive in all kinds of traffic crash data, such as is stated by the no free lunch theorem [26]. Therefore, deciding the best ML method for the problem at hand is a complex problem that makes expert ML knowledge necessary while maintaining demanding time, human

effort, and high computational capacities.

Automated machine learning (AutoML) arises as a promising approach to reduce the human effort and time cost of ML in research areas wherein specialized ML knowledge is an asset not always available or affordable, such as transportation and ITS. AutoML aims at automatically finding the best combination of pre-processing techniques, ML algorithm and hyperparameters that maximize a performance measure on given data without being specialized in the problem domain where the data comes; the latter is known as general-purpose AutoML [15]. AutoML methods have been successfully used in other areas; however, an extensive analysis to determine their strengths and weaknesses has not been carried out in very diverse learning tasks, such as crash severity forecasting. To the best of our knowledge, only a few papers have applied concepts of AutoML in the transportation domain, specifically in supervised traffic forecasting [6, 7, 8, 24]. Thus, other transportation areas, such as crash severity prediction that poses multiple challenges to the ML paradigm, remains unexplored by AutoML.

In this research, our objective is to determine to what extent general-purpose AutoML can be competitive against the ad hoc method on crash severity prediction modeled from a supervised learning perspective. To accomplish this aim, we test 3 state-of-the-art AutoML methods (Auto-Sklearn [10], TPOT [19], Auto-Gluon [9]), whose inner search strategies to automatically find competitive ML methods are based on 1) meta-learning, optimization, and ensemble learning, 2) genetic programming, and 3) stacking and ensemble learning, respectively. We compare the AutoML methods versus the general approach in crash severity prediction, which consists of selecting the best of a set of commonly used ML algorithms. Concretely, we contrast AutoML results with five ML methods (Gradient Boosting, Gaussian Naive Bayes, k-Nearest Neighbors, Multilayer Perceptron, Random Forest) in the task of forecasting the severity of crash accidents. The main contributions of this work are:

- To determine the benefits in terms of performance and human effort of general-purpose AutoML in supervised crash severity prediction.
- To characterize the performance of a set of ML methods against general-purpose AutoML in crash severity prediction.
- To test the performance of AutoML based on 1) meta-learning, optimization, and ensemble learning, 2) genetic programming, and 3) stacking and ensemble learning in supervised crash severity forecasting.

The rest of this paper is structured as follows. Section 2 presents background and related work about ML and AutoML methods in crash severity prediction. Section 3 exposes the methodology followed in this paper to carry out the proposed experimentation. Then, Section 4 analyzes the main results obtained. Finally, conclusions are discussed in Section 5.

## 2 Background and Related Work

This section reviews the background and literature related to ML and AutoML. We start, in Section 2.1, summarizing common data sources and the modeling approaches of ML in crash severity prediction. Then, Section 2.2 introduces the background of general-purpose AutoML and its influence in the transportation area.

### 2.1 Machine learning in crash severity prediction

Crash severity forecasting is designed to aid decision making and planning before accidents occur. Thus, it is possible to minimize the impact on human losses and improve the safety of the roads [29]. The forecasting of accidents can be done using different predictive approaches to analyze and determine the influence of various parameters on accident rates. The latter makes it feasible, on the one hand, to evaluate the effectiveness of the proposed forecasting methods and, on the other hand, to introduce new strategies to increase road safety [17].

Research, innovation, and development efforts aimed at analyzing and forecasting road accidents are increasing due to ITS development and the increased availability of traffic data. Therefore, the evolution of road accident prediction is exposed to a change in the prediction paradigm focusing on data-driven models, especially ML, and leaving aside traditional methods. From a ML perspective, crash severity prediction is approached by building a model from historical data to make predictions on new and unseen data [13, 25]. Different ML approaches can be used depending on the type of input and output (predicted) data. When the traffic measure to be forecasted is continuous, it should be approached as a regression problem; otherwise, a classification perspective can be implemented.

The transportation literature reports a significant number of ML methods that can be used to predict crash severity, such as neural networks, decision trees, and random forest, among others [13]. Using these methods, we can find works that have approached the prediction of crashes based on diverse data sources like GPS traces [27] or open government data portals [18].

Nevertheless, given the broad wide range of ML algorithms, there are no clear baselines that guide the process of selecting the most appropriate algorithm and its best hyperparameter setting for a particular crash severity problem at hand. In ML, this challenge is known as the model selection problem AutoML is one of the most successful approaches to address it so far.

## 2.2 Automated Machine Learning

According to [30], a ML pipeline  $P$  can be defined as a combination of algorithms  $A$  that transforms input data  $X$  into target values  $Y$ . Wherein  $A$  is composed of  $A_{\text{preprocessing}}$ , a subset of preprocessing techniques,  $A_{\text{feature}}$  that is a subset of feature engineering methods, and  $A_{\text{algorithm}}$ , a ML algorithm with a configuration of hyperparameters  $\lambda^i \in \Lambda$ . To build a ML pipeline with this structure, human effort, and high computational capacities are needed because no pipeline can achieve good performance on every learning problem [28]. This is usually done by data scientists who use their specialized knowledge or non-expert users who tackle the problem through a trial and error approach.

AutoML is an emerging area that automatically finds the best combination of preprocessing techniques and/or ML algorithm and its hyperparameters. The purpose is to improve a specific performance metric on a given dataset without being specialized in the problem domain wherein this data comes from [15]. Hence, AutoML reduces human bias and improves computational costs by making the construction of ML applications more efficiently.

Current literature [28, 15, 30] reports a variety of AutoML approaches. Within the most representative methods, we can find Auto-Sklearn [10], TPOT [19], and AutoGluon [9]. These are the AutoML methods considered in this paper, and a short description of them is presented below.

- **Auto-sklearn:** An AutoML method uses meta-learning, Bayesian optimization, and ensemble selection to find promising ML pipelines composed of preprocessing methods and one ML classifier. The pipelines are generated from a search space of 15 classifiers, 14 feature preprocessing techniques, and 4 data preprocessing methods. The interested reader is referred to [10] for further details.
- **TPOT:** It is an AutoML method that uses genetic programming to design and optimize ML pipelines, which maximize the classification accuracy for a given supervised learning dataset. At its core, TPOT has a search space of 6 classifiers,

5 feature preprocessing methods, and 7 data preprocessing methods. The interested reader is referred to [19] for further details.

- **AutoGluon:** It is an AutoML based whose search strategy of ML pipelines is based on bagging, stacking, and ensemble learning. Its search space is composed of 6 classifiers and 2 data preprocessing approaches. The interested reader is referred to [9] for further details.

In the transportation area, to the best authors' knowledge, only a few papers have used AutoML methods in this knowledge domain [24, 8, 6]. The first research carried out by Vlahogianni et al. [24] proposed a surrogate modeling technique that optimizes both the algorithm selection and the hyperparameter setting. The AutoML task is performed from an algorithms base of 3 ML methods to forecast average traffic speed.

After that, Angarita et al. in [7, 6, 8] tested Auto-Sklearn and AutoWEKA [23] in the prediction of traffic modeled as a supervised learning problem. Those papers compared the performance of AutoML w.r.t. the general approach in traffic forecasting, which consists of selecting by trial and error the best of a set of algorithms. The authors found that Auto-Sklearn and AutoWEKA are competitive approaches and have difficulties with the time horizon of predictions. As the horizon increases, the performance of the methods tends to fall. The latter is a crucial issue within AutoML that needs to be improved because, in the transportation domain, policy-makers require information about the evolution of traffic over the short- and long-term to improve traffic management.

The main differences between this research and the previous papers lay in the transportation problem approached and the AutoML methods considered. In this work, we are dealing with crash severity prediction modeled as a supervised classification problem (see Section 3.2). Thus, this would be the first time that the performance of AutoML is studied in road safety systems. The latter could offer clues about how useful it could be AutoML in a research area wherein expert ML knowledge is an asset not always available or affordable. We are also applying AutoML methods based on genetic programming and bagging-stacking-ensemble learning, which are AutoML approaches not previously considered in the transportation area.

## 3 Methodology

This paper seeks to answer whether general-purpose AutoML can be competitive against the ad hoc method on crash severity prediction modeled from a supervised

learning perspective. To accomplish such purpose, we compare to what extent the results of AutoML differ from the general approach in crash severity prediction in which a set of baseline algorithms is tested over the forecasting problem at hand, and the one with the best performance metric is chosen.

The following parts of this section are devoted to present the case study and its raw data in Section 3.1. Next, Section 3.2 introduces the datasets considered for this paper that were generated from the raw data. Then Section 3.3 shows the experimental set-up proposed, which includes the performance metrics, the methods for the experimentation, and the statistical tests selected to compare the methods.

### 3.1 Case Study and Raw Data

The 2019 report of the Colombian National Observatory of Road Safety points out 180.373 records of accidents. The latter indicates that, out of every 10 people involved in crash events, 4 end up injured and 1 dies [2]. In Colombia, the departments of Antioquia and Cundinamarca in their capital cities (Medellín and Bogotá) are reported as the places with the highest accident rate in the country. Within this research, we focus on the case of Medellín.

Medellín is the second most populated city of Colombia after Bogotá. It has a particular topography located at the Aburrá Valley in the central mountain range of the Andes. In 2019, the city had a population of 2.490.164 inhabitants. The Single National Registry of Traffic [3] reports that between 2016 and 2019, the number of cars in Medellín has been increasing steadily, equivalent to a 14.3% annual increase, which could be a potential factor to increase road accidents.

The raw data to be analyzed in this study is obtained from Medellín's open data portal: MeDATA ([www.medata.gov.co](http://www.medata.gov.co)). The data corresponds to traffic incidents registered by the Mobility Secretary of the Mayor's Office of Medellín between 2014 and 2018 (see Figure 1). For the particular case of this data, a traffic incident is understood as: "an event, generally involuntary, generated by at least one vehicle in motion, which causes damage to people and goods involved in it, and also affects the normal circulation of vehicles moving along the road or roads in the place or within the area of influence of the event" [1].

### 3.2 Datasets

From the raw data presented above, we generate 5 datasets wherein each of them corresponds to 1 year of historical data. Crash severity is the variable to be predicted, and it has two possible labels that character-

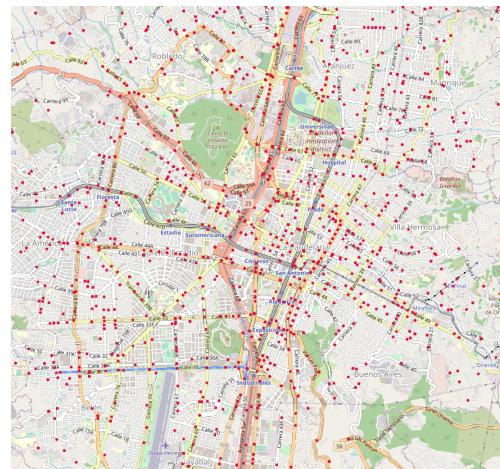


Figure 1: Sample of car accidents in Medellin between 2014 and 2018

Table 1: Datasets description

Dataset	Instances	Distribution of Classes	
		People injured	Only material damages
Dataset 2014	41.776	23.198	18.578
Dataset 2015	42.427	23.550	18.877
Dataset 2016	46.838	26.594	20.244
Dataset 2017	42.443	22.917	19.526
Dataset 2018	46.655	24.247	22.408

ize the severity of an accident: *people injured* and *only material damages*. Therefore, we model the objective variable as a supervised binary classification problem.

Each dataset contains the same 9 attributes composed of traffic-related and calendar features. Traffic attributes are *Type of accident*, *GPS coordinates of the accident*, and *Type of road wherein the accident occurs*. Whereas calendar features are *Minute*, *Hour*, *Day of the week*, *Day of the year*, and *Month*. Table 1 presents a summary of binary datasets used in the experimentation.

### 3.3 Experimental set-up

In this section, we present the metric used to measure performance, the methods chosen, the statistical test to compare the performance of the methods, and the hardware choice to carry out the experimentation.

- **Performance metrics:** For this paper's results, we follow the same experimental set-up proposed in the AutoML benchmark of [12]. It is the standard to make fair comparisons between AutoML approaches and baseline algorithms. In particular, the area under the receiver operator curve (ROC\_AUC) is used for the binary classification problems considered in this study. Besides, the final score achieved by every method is the average

of a ten-fold cross-validation process.

- **AutoML methods and Baseline:** AutoML competitors are Auto-Sklearn (*As*), TPOT (*Tp*), and AutoGluon (*Ag*) with its default hyperparameter values using three execution times: 15, 60, and 150 minutes. The latter corresponds to the time that the methods take to find the best ML algorithm and its hyperparameter configuration for a given dataset. The assumption is that longer time budgets lead to better results; therefore, such a progressive increase from 15 to 150 minutes should exemplify the expected behavior [14]. Additionally, every execution time assigned to a particular AutoML method is considered as an individual AutoML competitor.

As baseline methods, we use Gradient Boosting (*GrB*), Gaussian Naive Bayes (*GnB*), k-Nearest Neighbors (*kNN*), Multilayer Perceptron (*MLP*), and Random Forest (*RF*)<sup>1</sup>. Finally, it is relevant to highlight that we have not performed any optimization or extra-adjustment of the AutoML methods' hyperparameters neither the baseline. The latter is justified because our aim is to compare the performance of AutoML versus the baseline using the same human effort for both of them to carry out a fairer comparison.

- **Statistical tests:** We made use of non-parametric statistical tests to assess the differences in performance of the methods. Two statistical tests are used following the guidelines proposed in [11]. First, Friedman's test for multiple comparisons is applied to check whether there are differences among the methods. Then, the Holm's test is used to check whether the differences of the Friedman ranking are statistically significant or not.
- **Hardware choice:** We opted to use a cluster based on Centos 7.6 with kernel 3.10.0, and using QuadCore Intel Xeon processors of 16 GB of RAM.

## 4 Results

The objective of the experimentation done in this paper is two-fold:

- To compare the competitiveness and the significance of general-purpose AutoML versus the 5 baseline algorithms in forecasting crash severity.
- To assess the performance of general-purpose AutoML based on 1) meta-learning, optimization,

<sup>1</sup>All baseline methods correspond to the default implementations provided by Scikit-learn 0.20 library.

and ensemble learning, 2) genetic programming, and 3) bagging, stacking, and ensemble learning in supervised accident severity forecasting.

To make the aforementioned analysis, we show the results obtained by the 3 AutoML methods (*As*, *Ag*, *Tp*), considering 3 execution times for each of them (15, 60, 150 minutes), and the 5 baseline algorithms (*GrB*, *GnB*, *kNN*, *MLP*, *RF*) in Figures 2 and 3.

Figure 2 exposes a barplot with the results obtained by every method along the 5 datasets considered in this experimentation. The Y-axis represents the ROC\_AUC value of each method after the 10 cross-validation process. In contrast, X-axis has the datasets in which the methods were evaluated to obtain the reported ROC\_AUC values. Complementarily, Figure 3 shows a boxplot with information about the distribution of the results shown in Figure 2, but in this case, they are depicted in an aggregated form. The latter means that the Y-axis shows the averaged ROC\_AUC values of each method in the 5 datasets, while the X-axis represents the evaluated methods.

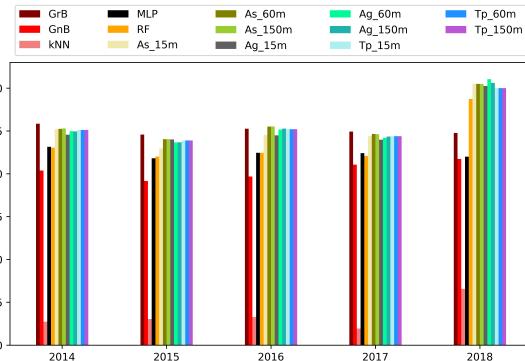


Figure 2: ROC\_AUC results of every method along the 5 datasets considered.

Beginning with the results shown in Figure 2, we can point out that all AutoML approaches and the baseline algorithms, except *kNN*, have acceptable performance in all the datasets. As a general overview, almost all the methods achieve at least 80% of accuracy in the 5 datasets. In the AutoML methods, regardless of each of them's inner search strategy, their performance is homogeneous across 4 datasets, with a special high performance in dataset 2018. Regarding the algorithms of the baseline, *GrB*, *MLP*, and *RF* are the best methods. For the latter case, it is relevant to highlight the performance of *GrB*, which is the baseline method able to have very similar results with respect to AutoML methods. The latter is interesting because it has not been considered baseline in the open AutoML benchmark [12], and it could be included in future AutoML works to make a comparison against ML algorithms.

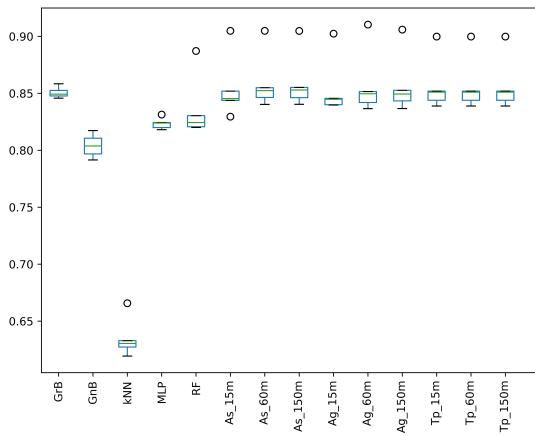


Figure 3: Aggregated ROC\_AUC results of each evaluated method.

Regarding the aggregated performance of the methods displayed in Figure 3, we can state that AutoML results are quite similar under the 3 execution times (15, 60, and 150 minutes) evaluated. There is no AutoML method that consistently outperforms all AutoML competitors. The latter is consistent with other results of AutoML in transportation problems wherein longer run times do not necessarily lead to drastically better results [6, 7]. For the case of *As* (based on meta-learning, optimization, and ensemble learning) and *Ag* (based bagging, stacking, and ensemble learning), they get slight score improvements as their allocated time budget increases. On the other hand, *TPOT* (based on genetic programming) cannot achieve gains in its performance regardless of the execution time considered.

In comparison to the baseline, AutoML methods perform better than *GnB*, *kNN*, *MLP*, and *RF*; however, they do not outperform *GrB*, which is the best method of the baseline. These results are interesting because to get the conclusion that *GrB* is the best method in the baseline and a strong competitor of AutoML, the human user should run all the algorithms in the baseline overall datasets and compare their performance among them, which is a time-consuming task. However, running any of the AutoML methods with at least a short execution time and employing less human effort, the user can achieve similar or better results than those obtained with *GrB*.

Finally, to assess whether the differences in performance observed in Figures 2 and 3 are significant or not, we made use of non-parametric statistical tests. Two statistical tests have been applied following the guidelines proposed in [11]. First, Friedman's test for multiple comparisons has been applied to check

whether there are significant differences between the AutoML methods and the baseline. Then, the Holm post-hoc test has also been applied to assess the significance of the differences in performance.

Table 2: Friedman's average ranking and *p*-values obtained through Holm post-hoc test using *As\_150m* as control method.

Method	Av. Ranking	<i>p</i> -values
As_150m	2.6	-
As_60m	2.7	0.97
GrB	3.6	0.71
Ag_150m	6	0.20
Tp_15m	6.2	0.17
Tp_60m	6.2	0.17
Tp_150m	6.2	0.17
As_15m	6.7	0.12
Ag_60m	7	0.10
Ag_15m	8	<b>0.04</b>
RF	11.3	<b>0.001</b>
MLP	11.5	<b>0.0007</b>
GnB	13	<b>0.00008</b>
kNN	14	<b>0.00001</b>

Considering that the *p*-value returned by these tests was 0, the null hypothesis can be rejected. Thus, the mean ranking returned by the test is displayed in Table 2, confirming the better global results of AutoML methods and *GrB*. Besides, Holm post-hoc test has used *As\_150m* as control algorithm (because it is the method that achieved the best overall performance) to assess the significance of the differences in performance with respect to the other algorithms. Table 2 presents the adjusted *p*-values returned by this test. In order to highlight significant differences, those *p*-values lower than 0.05 are shown in bold. Looking at Table 2, there are important differences in the test's outcomes. It can be said that *As\_150m* is statistically better than *Ag\_15m*, *RF*, *MLP*, *GnB*, and *kNN*, but not to *Ag*, *Tp*, and *GrB*.

## 5 Conclusions

In this work, we aimed at deepening into the benefits of AutoML for supervised crash severity prediction. Concretely, we have presented and compared to what extent the results of AutoML differ from the general approach in crash severity forecasting, which is based on choosing the best method from a set of predefined algorithms. We used Auto-Sklearn, TPOT, and Auto-Gluon as AutoML methods; whereas, Gradient Boosting, Gaussian Naive Bayes, k-Nearest Neighbors, Multilayer Perceptron, and Random Forest were the baseline methods. The comparisons were made on the prediction of crash severity modeled as a supervised classification problem. The performance of the methods was measured using the ROC\_AUC metric, and they

were tested on a real case study consisting of traffic accident data from the city of Medellín, Colombia.

The analysis of the obtained results showed that, in the case of AutoML, Auto-Sklearn was the approach with better performance than the other AutoML competitors. Although with no significant differences among them, Auto-Sklearn with 150 minutes of execution time was the method located in the first position of Friedman's average ranking. Regarding the baseline methods, Gradient Boosting was the best algorithm of the baseline and had comparable performance with AutoML methods. From a general view, AutoML is statistically better than *RF*, *MLP*, *GnB*, and *kNN*, and the transportation user can expect similar or even better results than the best baseline methods with a lower human effort. We have shown that Auto-Sklearn and AutoGluon can be considered powerful approaches that can support the model selection problem and improve the accuracy of the predictions in the domain of crash severity forecasting.

Further research lines that we aim to explore are: comparing the performance of AutoML in different case studies to corroborate whether its competitiveness is consistent through diverse crash accident settings; and carrying out a feature importance study to determine what are the attributes of crash severity data that more influence the performance of AutoML.

### Acknowledgement

J. S. Angarita-Zapata acknowledges the support received from the Spanish Ministry of Science and Innovation under the research project PID 2019-109393RA-I00.

G. Maestre-Gongora acknowledges the support received from Universidad Cooperativa de Colombia and Ministry of Science, Technology and Innovation for Scholarship "National Doctoral Studies- Call 567".

J. Fajardo acknowledges the support received from the European Union's Horizon 2020 research and innovation program under grant agreement No. 769142 and No. 815069.

### References

- [1] Law 769 of 2002 - National Traffic Code, [www.secretariosenado.gov.co/senado/basedoc/ley\\_0769\\_2002.html](http://www.secretariosenado.gov.co/senado/basedoc/ley_0769_2002.html), accessed: 2021-01-20.
- [2] National Observatory of Road Safety of Colombia, [ansv.gov.co/observatorio/indexf2019n.html](http://ansv.gov.co/observatorio/indexf2019n.html), accessed: 2020-12-28.
- [3] Single National Registry of Traffic of Colombia national report, [www.runt.com.co/](http://www.runt.com.co/), accessed: 2021-12-21.
- [4] United Nations road safety considerations in support of the 2030 agenda for sustainable development, [www.who.int/news-room/fact-sheets/detail/road-traffic-injuries](http://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries), accessed: 2021-01-21.
- [5] World Health Organization road traffic injuries, [www.who.int/news-room/fact-sheets/detail/road-traffic-injuries](http://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries), accessed: 2021-01-21.
- [6] J. S. Angarita-Zapata, A. D. Masegosa, I. Triguero, Evaluating Automated Machine Learning on Supervised Regression Traffic Forecasting Problems, Springer International Publishing, Cham, 2020, pp. 187–204.
- [7] J. S. Angarita-Zapata, A. D. Masegosa, I. Triguero, General-purpose automated machine learning for transportation: A case study of auto-sklearn for traffic forecasting, in: M.-J. Lesot, S. Vieira, M. Z. Reformat, J. P. Carvalho, A. Wilbik, B. Bouchon-Meunier, R. R. Yager (Eds.), Information Processing and Management of Uncertainty in Knowledge-Based Systems, Springer International Publishing, Cham, 2020, pp. 728–744.
- [8] J. S. Angarita-Zapata, I. Triguero, A. D. Masegosa, A Preliminary Study on Automatic Algorithm Selection for Short-Term Traffic Forecasting, in: J. Del Ser, E. Osaba, M. N. Bilbao, J. J. Sanchez-Medina, M. Vecchio, X.-S. Yang (Eds.), Intelligent Distributed Computing XII, Springer International Publishing, 2018, pp. 204–214.
- [9] N. Erickson, J. Mueller, A. Shirkov, H. Zhang, P. Larroy, M. Li, A. Smola, Autogluon-tabular: Robust and accurate automl for structured data, arXiv preprint arXiv:2003.06505.
- [10] M. Feurer, A. Klein, K. Eggensperger, J. Springenberg, M. Blum, F. Hutter, Efficient and Robust Automated Machine Learning, in: C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, R. Garnett (Eds.), Advances in Neural Information Processing Systems, Curran Associates, Inc., 2015, pp. 2962–2970.
- [11] S. Garcia, A. Fernandez, J. Luengo, F. Herrera, Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental

- analysis of power, *Information Sciences* 180 (10) (2010) 2044 – 2064.
- [12] P. Gijsbers, E. LeDell, S. Poirier, J. Thomas, B. Bischl, J. Vanschoren, An open source automl benchmark, CoRRWork presented at AutoML Workshop at International Conference on Machine Learning 2019.
- [13] C. Gutierrez-Osorio, C. Pedraza, Modern data sources and techniques for analysis and forecast of road accidents: A review, *Journal of Traffic and Transportation Engineering (English Edition)* 7 (4) (2020) 432 – 446.
- [14] I. Guyon, I. Chaabane, H. J. Escalante, S. Escalera, D. Jajetic, J. R. Lloyd, N. Macià, B. Ray, L. Romaszko, M. Sebag, A. R. Statnikov, S. Treguer, E. Viegas, A brief review of the chalearn automl challenge: Any-time any-dataset learning without human intervention, in: Proceedings of the Workshop on Automatic Machine Learning, New York, USA, 2016, pp. 21–30.
- [15] F. Hutter, L. Kotthoff, J. Vanschoren (Eds.), *Automated Machine Learning: Methods, Systems, Challenges*, Springer, 2018.
- [16] E. Kidando, R. Moses, E. E. Ozguven, T. Sando, Incorporating travel time reliability in predicting the likelihood of severe crashes on arterial highways using non-parametric random-effect regression, *Journal of Traffic and Transportation Engineering* 6 (5) (2019) 470 – 481.
- [17] E. Kurakina, S. Evtukov, J. Rajczyk, Forecasting of road accident in the dvre system, *Transportation research procedia* 36 (2018) 380–385.
- [18] K. Moriya, S. Matsushima, K. Yamanishi, Traffic risk mining from heterogeneous road statistics, *IEEE Transactions on Intelligent Transportation Systems* 19 (11) (2018) 3662–3675.
- [19] R. S. Olson, N. Bartley, R. J. Urbanowicz, J. H. Moore, Evaluation of a tree-based pipeline optimization tool for automating data science, in: Proceedings of the Genetic and Evolutionary Computation Conference 2016, ACM, New York, NY, USA, 2016, pp. 485–492.
- [20] A. Perallos, U. Hernandez-Jayo, E. Onieva, I. J. García-Zuazola, *Intelligent Transport Systems: Technologies and Applications*, 1st Edition, Wiley Publishing, 2015.
- [21] P. T. Savolainen, F. L. Manner, D. Lord, M. A. Quddus, The statistical analysis of highway crash-injury severities: A review and assessment of methodological alternatives, *Accident Analysis Prevention* 43 (5) (2011) 1666 – 1676.
- [22] P. B. Silva, M. Andrade, S. Ferreira, Machine learning applied to road safety modeling: A systematic literature review, *Journal of Traffic and Transportation Engineering* 7 (6) (2020) 775 – 790.
- [23] C. Thornton, F. Hutter, H. H. Hoos, K. Leyton-Brown, Auto-WEKA: Combined selection and hyperparameter optimization of classification algorithms, in: Proc. of KDD-2013, 2013, pp. 847–855.
- [24] E. I. Vlahogianni, Optimization of traffic forecasting: Intelligent surrogate modeling, *Transportation Research Part C: Emerging Technologies* 55 (2015) 14 – 23.
- [25] E. I. Vlahogianni, M. G. Karlaftis, J. C. Golias, Short-term traffic forecasting: Where we are and where we're going, *Transportation Research Part C: Emerging Technologies* 43 (2014) 3–19.
- [26] D. H. Wolpert, W. G. Macready, No free lunch theorems for optimization, *IEEE Transactions on Evolutionary Computation* 1 (1) (1997) 67–82.
- [27] X. Xiong, L. Chen, J. Liang, A new framework of vehicle collision prediction by combining svm and hmm, *IEEE Transactions on Intelligent Transportation Systems* 19 (3) (2018) 699–710.
- [28] Q. Yao, M. Wang, Y. Chen, W. Dai, H. Yi-Qi, L. Yu-Feng, T. Wei-Wei, Y. Qiang, Y. Yang, Taking Human out of Learning Applications: A Survey on Automated Machine Learning, CoRR.
- [29] X. Zheng, M. Liu, An overview of accident forecasting methodologies, *Journal of Loss Prevention in the process Industries* 22 (4) (2009) 484–491.
- [30] M.-A. Zöller, M. F. Huber, Survey on Automated Machine Learning, arXiv preprint arXiv:1904.12054.