Evaluating Automated Machine Learning on supervised regression Traffic Forecasting Problems

Juan S. Angarita-Zapata, Antonio D. Masegosa and Isaac Triguero

Abstract Traffic Forecasting is a well-known strategy that supports road users and decision-makers to plan their movements on the roads and to improve the management of traffic, respectively. Current data availability and growing computational capacities have increased the use of Machine Learning methods to tackle Traffic Forecasting, which is mostly modelled as a supervised regression problem. Despite the broad range of Machine Learning algorithms, there are no baselines to determine what are the most suitable methods and their hyper-parameters configurations to approach the different Traffic Forecasting regression problems reported in the literature. In Machine Learning, this is known as the Model Selection Problem, and although Automated Machine Learning methods has proved successful dealing with this problem in other areas, it has hardly been explored in Traffic Forecasting. In this work, we go deeply into the benefits of Automated Machine Learning in the aforementioned field. To this end, we use Auto-WEKA, a well-known AutoML method, on a subset of families of Traffic Forecasting regression problems characterised by having loop detectors, as traffic data source, and scales of predictions focused on the point and the road segment levels within freeway and urban environments. The experiments include data from the Caltrans Performance Measurement System and the Madrid City Council. The results show that AutoML methods can provide competitive results for TF with low human intervention.

Juan S. Angarita-Zapata
DeustoTech, Faculty of Engineering, University of Deusto, 48007 Bilbao, Spain.
e-mail: js.angarita@deusto.es

Antonio D. Masegosa
DeustoTech, Faculty of Engineering, University of Deusto, 48007 Bilbao, Spain
IKERBASQUE, Basque Foundation for Science, 48011, Bilbao, Spain.
e-mail: ad.masegosa@deusto.es

Isaac Triguero
Automated Scheduling, Optimisation and Planning (ASAP), School of Computer Science, University of Nottingham, Nottingham NG8 1BB, UK.
e-mail: Isaac.Triguero@nottingham.ac.uk

1 Introduction

Urban development, population growth, and high motorisation rates have increased levels of congestion in cities around the world. One well-established strategy to tackle congestion is the design, development and implementation of Traffic Foecasting (TF) systems. TF can be defined as the prediction of near future traffic conditions (e.g., speed, travel time) for single locations, road segments, or entire networks [33].

The recent emergence of telecommunications technologies integrated to transportation infrastructure generates vast volumes of traffic data. This unprecedented data availability and growing computational capacities have incremented the use of Machine Learning (ML) to address TF. From a data-driven perspective TF can be addressed using different modelling approaches, such as a supervised regression problem [9, 2], as a supervised classification problem [2, 16], or as a clustering-pattern recognition problem [34, 30]. Nevertheless, the supervised regression approach is typically the most widely used modelling perspective in the TF literature. During the last decades, the number of academic publications about TF approached as a supervised regression problem has increased extensively. From a ML perspective [3], a supervised TF regression problem is focused on building a predictive model using historical data to make predictions of continuous traffic measures, based on unseen data.

The transportation literature reports a great number of ML algorithms that can be used for the prediction of traffic, such as, Neural Networks (NNs), Support Vector Machines (SVMs), k-Nearest Neighbors (k-NN) and Random Forest (RF), among others [10]. However, given the broad wide range of ML methods, there are no clear baselines that guide the process of selecting the most appropriate algorithm and its best hyper-parameter setting given the characteristics of the TF problem at hand. In the ML area, this challenge is known as the Model Selection Problem (MSP) [13], and Automated Machine Learning (AutoML) [13, 35] has been one of the most successful approaches to address it so far. AutoML aims at automatically finding the ML algorithm and hyper-parameters configuration pair which maximises a performance measure on given data, using an optimisation strategy that minimizes a predefined loss function.

Although AutoML methods have approached the MSP with high performance in other research areas [37, 17], to the best of authors' knowledge there are only two works that have tackled the MSP in TF [2, 31]. On the one hand, Vlahogianni [31] proposed a AutoML method that handles the prediction of speed in a time horizon of 5 minutes using a supervised regression approach. Contrary to [31], Angarita-Zapata et. al [2] focused on a TF supervised classification problem to predict the Level of Service through multiple time horizons and using a different AutoML method (Auto-WEKA). Notwithstanding, in the spite of the progress achieved by the aforementioned works, AutoML in the transportation area is still in its infancy and there are TF supervised regression problems [1] that still need to be addressed in order to develop more reliable TF systems.

In this book chapter, our objective is to continue deepening into the benefits of AutoML for TF from a supervised regression perspective, following the research line proposed in [2]. Specifically, we use Auto-WEKA, a well-known AutoML method, on a subset of families of TF problems characterised by having loop detectors, as traffic data source, and scales of predictions focused on the point and the road segment levels within freeway and urban environments. We compare the AutoML method versus the general approach in TF, which consists of selecting the best of a set of commonly used ML algorithms. Concretely, we contrast Auto-WEKA results with four state-of-the-art ML algorithms (NN, SVM, k-NN and RF) in the task of forecasting traffic speed, using data taken from the Caltrans Performance Measurement System (PeMS) and the Madrid (Spain) City Council. The main contributions of this work are:

- Exploring, in a more deeply way, the benefits of AutoML for TF supervised regression problems.
- Determining suitable ML algorithms to approach the prediction of traffic at scales of predictions focused on the point and the road segment levels within, freeway and urban environments.

The rest of this paper is structured as follows. Section 2 presents background and related work about ML and AutoML methods in the area of TF. Section 3 exposes the methodology followed in this work. Then, Section 4 shows main results obtained by the AutoML method and the baseline algorithms considered. Finally, the main conclusions of the chapter are discussed in Section 5.

2 Background

This section reviews literature related to ML and AutoML in the context of TF. We start presenting a brief history of how TF has evolved in the last years (Section 2.1). Then, Section 2.2 summarises the ML modelling perspectives used to approach the prediction of traffic. Later, Section 2.3 discusses ML methods for TF. Finally, Section 2.4 reviews AutoML methods.

2.1 A brief history of Traffic Forecasting

TF is a relevant research are because of its active role in Intelligent Transportation Systems (ITSs) to address traffic congestion. The main objective of TF is the prediction of near future traffic measures based on past traffic data [33]. Three decades ago, transportation research was focused on predicting traffic at a single location using traffic theory models [12] and classical statistical methods [11]. However, these two approaches are not able to deal, in a efficient way, with uncertainty and big volumes of traffic data.

Recently, the emergence of sensing and telecommunications technologies integrated to ITSs started to generate vast volumes of traffic data, which in turn caused a switch in the modelling paradigm towards a data-driven approach [20]. Since that time, a variety of methods have been proposed placing special emphasis on Computational Intelligence-based approaches, such as NNs [36, 15], Fuzzy logic [4, 5], and Bio-inspired algorithms [24, 16], among others [33].

Currently, although some TF literature still relies on statistical methods, ML methods have attracted the interest of the transportation community and they are present in a wide proportion of contemporary research (see the review published by Ergamun and Levinson [6]). As computational capacities has increased, more complex scenarios with different road settings can be tackled with ML (e.g. network-wide predictions) due to its ability to predict traffic without the need of knowing theoretical traffic mechanisms [10]; in this way, leaving behind traditional approaches in the prediction of traffic.

2.2 Machine Learning modelling approaches for Traffic Forecasting

TF, from a data-driven perspective, is facilitated by sensing infrastructure of ITSs. Some technologies, such as Automatic Vehicle Identification, Electronic Tolls, and GPS, collect individual traffic data related to each vehicle on the road; meanwhile, others collect macroscopic traffic measures (averages of many vehicles) as Vehicle Detection Stations (VDS). Taking as input the data provided by any of the mentioned data sources, when the objective is to predict a continuous traffic variable, the possible ML modelling approaches can be supervised regression or clustering-pattern recognition. In the first case, the focus is on using ML algorithms to learn a functional form based on the input data, without prior models or data distribution assumptions [6]. In this context, the goal is to approximate the learned mapping function in such a way that when the model faces new and unseen traffic data, it is able to make accurate predictions.

In the second case, clustering-pattern recognition, the objective is finding the relationships of different locations by characterising similar traffic measures values from one road to another, and grouping them in clusters that divide the network into correlated groups. Once the clusters have been identified, the next step is to use a supervised regression perspective to predict the traffic conditions, cluster by cluster, based on historical traffic data belonging to each group.

Finally, when the objective is forecasting a discrete traffic measure, the modelling approach is supervised classification that also learns a mapping function based on historical data. For instance, ML methods can forecast the Level of Service (LoS) of a specific road. LoS is a categorical variable that measures the quality of the traffic through letters from A to E in a gradual way, being category A moderate traffic and category E extended delays [26]. It is important to clarify that the forecasting of discrete variables could be also addressed as a supervised regression problem in some occasions, predicting either speed or density (continuous values), and then discretisizing these predictions to obtain the categorical outputs.

Regardless of the aforementioned modelling approaches, ML methods for traffic prediction are based on forecasting traffic based on historical data and their objective is to predict traffic in similar conditions in which this data was observed. In this context, traffic forecasting under severe changes, such as new road infrastructure or traffic control policies, is out of the scope of ML and simulation-based approaches become alternatives that are more suitable [23].

In this work, we centre on ML applied to VDS data from a supervised regression approach because of two reasons. First, VDS data is the most common type of data available and used in transportation literature [20]. Secondly, the supervised regression approach is, by far, the most widely used modelling perspective to predict traffic [1].

2.3 Machine Learning algorithms for TF using VDS data

ML methods applied to TF can be categorised into single or hybrid. The first type corresponds to adaptations of existing ML algorithms that can be classified as parametric and non-parametric [32]. The parametric category assumes the relationship between the explanatory and response variables as known; meanwhile, the non-parametric ones are able to model nonlinear relationships without requiring the mentioned assumptions. Commonly non-parametric algorithms are NNs, SVMs, k-NN, and RF [10, 33].

As mentioned before, the other approach of ML algorithms is hybridisation. Within it, two or more algorithms, from ML or even other areas, are combined to find synergies that improve their isolated performance. Some recent examples are [18], where authors integrate a Boltzmann Machine with Recurrent NNs, and [16], where Genetic Algorithms are integrated with Fuzzy Systems.

Despite the great variety of ML methods, dealing with the MSP in TF is not a trivial task, as mentioned before. The general approach to tackle the MSP in TF consists of testing a set of algorithms with multiple hyper-parameter combinations and select the best one. This requires expert knowledge and a lot of human effort. Nowadays, AutoML has received a lot of attention in ML because of its promising results in dealing with the MSP with low human intervention.

2.4 AutoML in Traffic Forecasting

As stated above, AutoML deals with MSP as an optimisation problem whose objective consists of finding the ML algorithm, from a pre-defined base of algorithms, and its hyper-parameter configuration that maximises an accuracy measure on a given ML problem. In this sense, AutoML aims to improve the current way of building ML applications by automating the application of ML algorithms to data-sets, in such a way that enables to human users avoiding tedious tasks (e.g.,hyper-parameter optimisation). Although current AutoML methods have already produced impressive results, the field is still far from being mature.

The first AutoML method in tackling simultaneously the selection of algorithm and hyper-parameters was Auto-WEKA [29]. It uses Bayesian optimisation to search for the best pair (algorithm, hyper-parameter setting), considering a base of 39 algorithms implemented in WEKA (a well-known open-source ML software that contains algorithms for data analysis and predictive modelling). Subsequently, Komer et al. [14] and Feurer et al. [7] developed Hyperopt-sklearn and Auto-sklearn, respectively. These two frameworks automatically select ML algorithms and hyper-parameter values from scikit-learn[1]. In the case of [14], the AutoML method uses Hyperopt Python library for the optimisation process, concretely a Bayesian optimization method as Auto-WEKA. Meanwhile, Auto-sklearn stores the best combination of ML algorithm and hyper-parameters that have been found for each previous ML problem, and using meta-learning it chooses a starting point for a sequential optimisation process.

More recently, Sparks et al. [27] proposed a method that supports distributed computing for AutoML, and Sabharwal et al. [25] developed a cost-sensitive training data allocation method that assesses a pair (algorithm, hyper-parameters setting) on a small random sample of the data-set, and gradually expands it over time to re-evaluate it when one combination is promising. Then, Olson and Moore [21] designed a framework for building and tuning classification and regression ML pipelines. It uses genetic programming to construct flexible pipelines and to select an algorithm in each pipeline stage. However, TPOT does not exhaustively test all different combinations of hyper-parameters which in turn causes that some promising configuration may be ignored.

Lately, Swearingen et al.[28] built ATM, which is a collaborative service to build optimised ML pipelines. This framework has a strong emphasis on parallelisation enabling the distribution of a single combination (algorithm, hyper-parameter setting) in a cluster to process it in a more efficient way. Currently, ATM uses the same base of algorithms from scikit-learn, and it finishes the optimisation process after either a fixed number of iterations or after expending a time budget defined by the human user. One year later, Mohr et al. [19] developed ML-Plan, a framework for building ML pipelines based on hierarchical task networks. ML-Plan is initialised with a fixed set of pre-processing algorithms, classification algorithms, and their respective potential hyper-parameters. Nevertheless, ML-Plan only considers a supervised classification approach, ignoring the supervised regression perspective that, as it was stated before, is the most common approach in TF.

For this research, we select Auto-WEKA because of a twofold reason. First, its wider variety base of regression algorithms in comparison with the others approaches reviewed. Second, unlike the aforementioned methods that only consider a predefined set of hyper-parameters combinations, Auto-WEKA has no limitations in the hyper-parameter space to be explored.

Moving from general-purpose AutoML methods to the transportation area, to the best authors' knowledge, only two works have tackled the MSP in TF [2, 31]. Angarita et. al [2] used Auto-WEKA and compared it to the general approach (which consists of selecting the best of a set of algorithms) over a multi-class

[1] Scikit-learn is a Python library of ML algorithms: http://scikit-learn.org

imbalanced classification TF problem, predicting traffic Level of Service at a fixed location through multiple time horizons. In the case of Vlahogianni [31], the author proposed a meta-modelling technique that, based on surrogate modelling and a genetic algorithm with an island model, optimises both the algorithm selection and the hyper-parameter setting. The AutoML task is performed from an algorithms base of three ML methods (NN, SVM and Radial Base Function) that forecast average speed in a time horizon of 5 minutes, using a regression approach.

The main differences between this research and the two aforementioned works lay on the addressed TF problems and ML modelling approach used. Regarding the problems, we predict traffic speed for TF problems characterised by having scales of predictions at the point and the road segment level, within freeway and urban environments. This means that we consider problems that take into account the temporal dimension of traffic, on the one hand, and the temporal-spatial component of traffic data on the other hand. Lastly, with respect to the modelling approach, we use an AutoML method for a supervised regression approach that considers a much broader base of algorithms that the one used by Vlahogianni [31]. In the case of the work of Angarita et. al [2], the same AutoML method is considered but for a supervised classification approach whereas in this work we are considering a regression approach.

3 Methodology

This research seeks to keep exploring the benefits that AutoML can bring to TF. To accomplish such purpose, we compare to what extent the results of AutoML differ from the general approach in TF, in which a set of Baseline Algorithms (BAs) is tested over the forecasting problem at hand, and the one with best performance metrics is chosen. We select Auto-WEKA, as AutoML method, and NN, SVM, $k-NN$, and RF, as the BAs that represent the general approach. The following parts of this sections are devoted to give more details about how Auto-WEKA finds, in a iterative way, the best combination of ML algorithm and hyper-parameters settings (Section 3.1); the raw data stored to represent the prediction of traffic speed in freeway and urban environments (Section 3.2); and the data-sets generated as well as the experimental set-up of this work (Section 3.3).

3.1 Auto-WEKA

Auto-WEKA approaches the algorithm selection problem through a Bayesian optimisation method. It considers the space of WEKA's ML algorithms $\boldsymbol{X} = \left\{X^{(1)}, \ldots, X^{(k)}\right\}$ and their hyperparameter spaces $\boldsymbol{A} = \left\{A^{(1)}, \ldots, A^{(k)}\right\}$ to identify the combination of algorithm $X^{(i)} \in X$ and hyper-parameters $A^{(i)} \in A$, which minimises cross-validation loss (Equation 1), where $\gamma\left(X_A^{(i)}, D_{train}^{(i)}, D_{test}^{(i)}\right)$ denotes the loss achieved by algorithm $X^{(i)}$ with hyper-parameters $A^{(i)}$ when trained on training data-set $D_{train}^{(i)}$ and evaluated on test data-set $D_{test}^{(i)}$.

$$X_A^* * = \underset{X^{(i)} \in X, A^{(i)} \in A}{argmin} \frac{1}{K} \sum_{i=1}^{k} \gamma\left(X_A^{(i)}, D_{train}^{(i)}, D_{test}^{(i)}\right) \tag{1}$$

Thornton et. al [29] call this the combined algorithm selection and hyperparameter optimisation (CASH) problem: determining $argmin_{\theta \in \Theta} f(\Theta)$ wherein each configuration $\theta \in \Theta$ contains the choice of algorithm $X^{(i)} \in X$ and its hyper-parameters setting $A^{(i)} \in A$. With this problem definition, the Bayesian optimisation strategy fits a probabilistic model to capture the relationship between different hyperparameter configurations and their performance; it then uses this model to select the most promising hyperparameter setting, assesses it, updates the model with the result of configuration chose, and iterates until a predefined time budget is reached.

One drawback of the Bayesian optimisation approach is its high computational cost at the moment of initialising the search for the most promising hyperparameter setting. Besides, as the space of algorithms and hyperameters increases, the computational cost of evaluating them also increments. To overcome this issue, Feurer et al . [7] proposed a method to warm-start the Bayesian search. Concretely, the authors use a meta-learning approach that quickly suggests some instantiations of ML algorithms with their hyperparameter settings that are likely to perform well. Meta-learning performs a pre-selection of promising configurations that are fed into the optimisation procedure, which ultimately is in charge of doing a fine-grained optimisation of them. Thus, it is possible to decrease in an efficient way the computational costs associated with broad spaces of algorithms and hyperparameters.

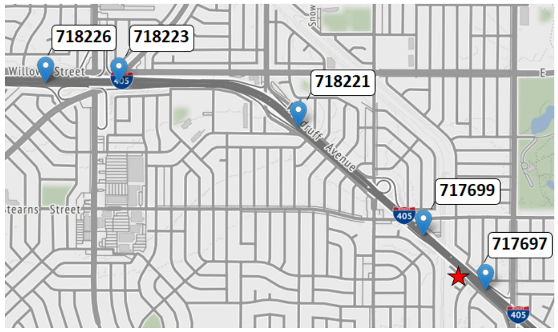
Taking into account that Auto-WEKA does not incorporate any mechanism to deal with the aforementioned issue, in this work, we consider the full range of ML algorithms that the AutoML includes.

3.2 Raw data

Freeway data used in this work is provided by the Caltrans Performance Measurement System [2] whose information is collected, in real time every 30 seconds, from nearly 40,000 individual detectors spanning the freeway system across the metropolitan area of California (USA). According to recent literature, this data source is highly used in the area of TF because of its high quality data, availability of various traffic measures and its public accessibility.

[2] http://pems.dot.ca.gov

Fig. 1 Location of 5 freeway sensors in California State (USA). The detector marked with a ⋆ symbol represents the forecast target location.

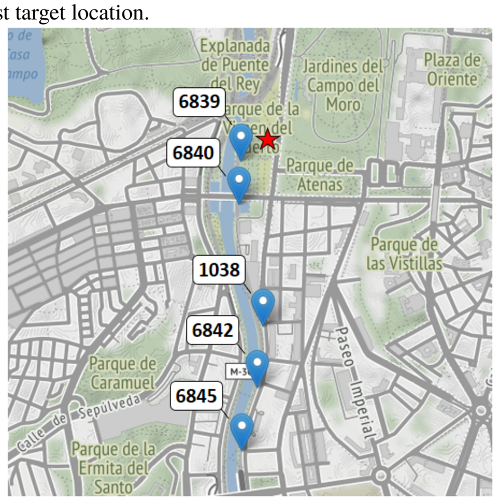


The route selected for our experiments is the California Interstate I405-S. It is a heavily trafficked freeway by commuters along its entire length [22]. Particularly, we focus on the loop detectors shown in Figure 1, where the detector marked with a ⋆ symbol represents the forecast target location. The traffic measure collected from the detectors is speed in an aggregation time of 5 minutes within the time window from March 1, 2019 to April 7, 2019 (38 days of data).

Contrarily, the urban data included in this research is the one obtained from the Madrid Open Data Portal [3]. The Madrid City Council provides through this website access to traffic data around the whole city, publishing 15-minute aggregates and live 5-minute aggregates of flow, occupancy and speed data in more than 3600 measuring stations (loops).

[3] https://datos.madrid.es/portal/site/egob/

Fig. 2 Location of 5 urban sensors in Madrid city (Spain). The detector marked with a ⋆ symbol represents the forecast target location.



We chose the M-30 motorway that circles the central districts of Madrid and that is considered the busiest Spanish road because of its traffic jams. On this route, we focus on the loop detectors depicted in Figure 2 where again the ⋆ symbol represents the forecast target location. From them, we extract traffic speed data, in an aggregation time of 15 minutes, for the period slapsed between February 2, 2019 and February 28, 2019 (27 days of data).

3.3 Data-sets and Experimental set-up

In this work, we approach two types of TF regression problems with different instances of them that are described below. The first type corresponds to the prediction of traffic at a target location, in a freeway environment; on the one hand, using only past traffic data of this location (temporal data, T), and then considering historical traffic data coming from the target location and from four downstream positions (temporal and spatial data, TS). Besides, in both instances, the input is enriched with calendar data (CD).

The second kind of TF problems is focused on forecasting traffic speed within an urban context. Repeatedly, the predictions are done for a single target location considering exclusively historical data of this spot; and on the other hand, taking into account past traffic data of the objective location together with other four downstream positions. Again, the input data in both instances is complemented with calendar data.

For the two families of TF problems described, we generate 18 data-sets in which speed is the traffic measure to be predicted. In the freeway case, time horizons wherein speed is predicted are 5, 15, 30, 45, and 60 minutes using data granularity of 5 minutes (granularity means how often the traffic measure is aggregated). Differently, for the urban TF problems, the forecasting time steps are 15, 30, 45, and 60 minutes with data granularity of 15 minutes. To better identify the data-sets, they are named following the next structure: *Context_InputData_Granularity_TimeHorizon*.

Attributes of freeway data-sets where the input is composed of only traffic data from the target location together with calendar data are: *Day of the week; Minute of the day; Traffic speed of the objective spot at past 5, 10, 15, 20, 25, 30, 35, 40, and 45 minutes; and Current traffic speed in such point*. In the case of freeway data-sets where the input consists of historical speed taken from the target location and from four downstream detectors, the attributes are: *Day of the week; Minute of the day; Traffic speed of the target position and four downstream locations at past 5, 10, 15, and 20 minutes; and Current speed of these five spots*.

Attributes of urban data-sets in which the input comprises traffic data of the target spot and calendar information are: *Day of the week; Minute of the day; Traffic speed of the objective spot at past 15, 30, 45, 60, 75, 90, 105, 120, and 135 minutes; and Current traffic speed in this point of interest*. Contrarily, urban data-sets, wherein the input is past traffic speed stored from the target location and from four downstream positions in addition to calendar, have the following attributes: *Day of the week; Minute of the day; Traffic speed of the target position and four downstream locations at 15, 30, 45, and 60 minutes in the past; and Current speed of these five positions*.

Lastly, for the experimentation with Auto-WEKA, three execution times (ET) were considered: 15, 150, and 300 minutes. These correspond to the time that the method takes to find the best ML algorithm and its hyper-parameter configuration for a given data-set. Furthermore, five repetitions with different initial seeds were carried out for each execution time. In the case of the BAs, we test them using WEKA. The process of evaluating every BA over a data-set was done with 5 repetitions with different initial seeds, and using the default hyper-parameter setting offered by WEKA. We have not performed any optimisation or extra-adjustment of the BAs' hyper-parameters because our aim is to compare the performance of AutoML versus BAs using the same human effort for both of them in order to make a fairer comparison.

4 Results

This section presents the results obtained with the experimental set-up proposed in the previous section. We evaluated the performance of the AutoML method and the BAs using the metric *Root-Mean-Square Error (RMSE)*, which is applied for regression problems to measure the average magnitude of the error between the predictions of a learning model and the actual values extracted from the raw data. Its calculation is expressed as $RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \overline{y_i})^2}$ wherein n corresponds to the number of samples in the data-set.

Table 1 shows the mean and standard deviation (between brackets) of the $RMSE$ values obtained by both Auto-WEKA and the BAs over all repetitions for each data-

Table 1 Mean $RMSE$ values and their standard deviations (in brackets) obtained, for freeway (Fw) and urban (Ub) data-sets, by the AutoML method and the BAs.

Data-sets	**Auto-WEKA**			**Baseline Algorithms**			
	15mET	**150mET**	**300mET**	**k-NN**	**NN**	**RF**	**SVM**
Fw_T+CD_5m_5	2.87 (0.08)	2.87 (0.08)	2.91 (0.06)	4.25 (0.14)	2.93 (0.20)	**2.86 (0.06)**	2.90 (0.09)
Fw_T+CD_5m_15	5.81 (0.33)	5.80 (0.28)	5.82 (0.34)	6.66 (0.22)	5.90 (0.45)	**5.16 (0.19)**	5.68 (0.18)
Fw_T+CD_5m_30	7.35 (0.85)	**6.76 (0.41)**	6.99 (0.68)	8.30 (0.39)	9.05 (1.59)	7.06 (0.13)	8.19 (0.23)
Fw_T+CD_5m_45	8.30 (1.09)	7.83 (1.12)	8.53 (0.30)	8.72 (0.20)	10.26 (1.15)	**7.70 (0.25)**	9.65 (0.17)
Fw_T+CD_5m_60	9.12 (1.87)	9.01 (1.67)	9.61 (1.70)	9.01 (0.26)	10.90 (0.74)	**7.99 (0.08)**	10.56 (0.09)
Fw_TS+CD_5m_5	1.19 (0.05)	1.16 (0.01)	1.17 (0.03)	1.46 (0.03)	1.44 (0.29)	**1.13 (0.05)**	1.11 (0.03)
Fw_TS+CD_5m_15	1.92 (0.00)	2.00 (0.47)	2.01 (0.55)	1.78 (0.06)	2.16 (0.24)	**1.64 (0.03)**	1.86 (0.05)
Fw_TS+CD_5m_30	2.12 (0.37)	2.37 (0.47)	**1.90 (0.41)**	1.95 (0.13)	2.60 (0.26)	1.91 (0.08)	2.43 (0.05)
Fw_TS+CD_5m_45	2.50 (0.48)	2.33 (0.49)	2.14 (0.49)	**2.05 (0.09)**	2.92 (0.24)	2.06 (0.07)	2.82 (0.05)
Fw_TS+CD_5m_60	3.17 (0.63)	2.82 (0.69)	2.26 (0.49)	**2.16 (0.09)**	2.89 (0.15)	**2.16 (0.12)**	3.10 (0.11)
Ub_T+CD_15m_15	**5.62 (0.15)**	5.76 (0.26)	5.71 (0.36)	7.74 (0.40)	7.68 (1.27)	5.77 (0.03)	6.05 (0.11)
Ub_T+CD_15m_30	**5.71 (0.29)**	5.97 (0.57)	5.74 (0.35)	8.20 (0.37)	8.02 (1.03)	5.80 (0.23)	6.33 (0.25)
Ub_T+CD_15m_45	5.68 (0.14)	5.73 (0.15)	**5.65 (0.03)**	8.45 (0.20)	8.25 (1.88)	6.16 (0.26)	6.80 (0.37)
Ub_T+CD_15m_60	5.91 (0.12)	**5.85 (0.13)**	5.88 (0.25)	8.52 (0.60)	7.25 (0.70)	5.98 (0.42)	7.05 (0.42)
Ub_TS+CD_15m_15	8.97 (0.46)	8.84 (0.38)	8.83 (0.18)	10.42 (0.72)	14.81 (0.93)	**7.92 (0.30)**	8.45 (0.35)
Ub_TS+CD_15m_30	7.91 (0.23)	7.80 (0.17)	**7.61 (0.23)**	12.95 (0.80)	17.18 (1.82)	9.34 (0.53)	10.75 (0.66)
Ub_TS+CD_15m_45	9.89 (0.18)	9.56 (0.23)	**9.54 (0.24)**	13.96 (0.79)	19.02 (2.94)	9.74 (0.51)	11.53 (0.41)
Ub_TS+CD_15m_60	9.25 (0.09)	9.07 (0.24)	**8.94 (0.11)**	13.07 (0.52)	17.09 (0.96)	9.77 (0.91)	11.94 (0.84)

set. $RMSE$ values in bold indicate the best result in every data-set achieved from either any of the BAs or any of the Auto-WEKA's execution times.

As it can be seen in Table 1, the AutoML method performs better than the BAs along eight of the data-sets. In all the other cases, RF or NN obtain better results than Auto-WEKA although with small improvements ranging from 0.01 to 1.31 in the $RMSE$ values. These results are interesting because in order to get the conclusion that RF and NN are the best BAs in those cases, the human user should run all BAs over all data-sets and compare their performance among them, which is a time consuming task. However, running Auto-WEKA only once, and therefore employing less human effort, the user can achieve similar or better results than those obtained with the best BAs.

Regarding data-sets characteristics, we can see that they do influence the differences between results of Auto-WEKA and BAs. Concretely, for all urban data-sets with a granularity of 15 minutes (with the exception of the data-set $Ub\_TS+CD\_15m\_15$), the AutoML method obtains the best $RMSE$ values. On the other hand, RA works specially well on freeway data-sets with the shortest and longest time horizons to be predicted, excluding both $Fw\_T+CD\_5m\_30$ and $Fw\_TS+CD\_5m\_30$ data-sets in which the AutoML get the best $RMSE$ performance.

Another interesting aspect is the relation between the execution time and the performance of the models provided by Auto-WEKA. Longer execution times contribute to obtaining better results, particularly, in the urban data-sets with longer time horizons. In the case of freeway data-sets where RF and NN algorithms are the best ones, the results improve when the Auto-WEKA's execution time increases from 15 to 150 minutes, but they are worse when we pass from 150 to 300 minutes. Similar

Table 2 Friedman's average ranking for the Auto-WEKA execution times

Auto-WEKA mET	Avg. Ranking
300mET	1.8333
150mET	1.8889
15mET	2.2778

to what happened in [2], we observed that this worsening is due to the over-fitting produced by the hyper-parameters selected by Auto-WEKA. This result indicates that it is necessary to introduce mechanisms in the AutoML method to deal with over-fitting, especially when execution times are high.

To assess whether the differences in performance observed in Table 1 are significant or not, we made use of non-parametric statistical tests. Two statistical tests have been applied, following the guidelines proposed in [8]. First, the Friedman's test for multiple comparisons has been applied to check whether there are significant differences among the three execution times of Auto-WEKA. Given that the p-value returned by this test is 0.35, the null hypothesis cannot be rejected in any of the cases. According to the Friedman's average ranking, shown in Table 2, $300meT$ is the best execution time of the AutoML method confirming that, for this type of methods, the longer the execution time, the better.

In order to assess if the differences observed between the best AutoML method ($300mET$), and the BAs are significant or not, we also used Friedman's non-parametric test. Considering that the p-value returned by these tests was 0, the null hypothesis could be rejected. The mean ranking returned by the test is displayed in Table 3, confirming the better global results of RF against the others BAs and Auto-WEKA $300mET$. At the same time, it also shows the better global results of the AutoML method versus $k-NN$, NN and SVM.

Table 3 Friedman's average ranking and Adjusted p-Values obtained through Holm post-hoc test using RF as control algorithm

Algorithms	Avg. Ranking	Adj. p-values
RF	1.6111	-
Auto-WEKA (300mET)	2	4.60597 e-1
SVM	3.0556	**1.2264 e-2**
k-NN	3.7778	**1.18 e-4**
NN	4.5556	**0**

Holm post-hoc test has also been applied using RF 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 3 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 3, there are important differences in the test's outcomes. It can be said that RF results improve significantly the rest of BAs, but not the $300mET$ of Auto-WEKA.

Table 4 ML methods selected by Auto-WEKA and absolute frequency in which they were suggested for freeway and urban data-sets

ML methods	*Fw_T+CD_5m_5*	*Fw_T+CD_5m_15*	*Fw_T+CD_5m_30*	*Fw_T+CD_5m_45*	*Fw_T+CD_5m_60*	*Fw_TS+CD_5m_5*	*Fw_TS+CD_5m_15*	*Fw_TS+CD_5m_30*	*Fw_TS+CD_5m_45*	*Fw_TS+CD_5m_60*	**Subtotal Fw**	*Ub_T+CD_15m_15*	*Ub_T+CD_15m_30*	*Ub_T+CD_15m_45*	*Ub_T+CD_15m_60*	*Ub_TS+CD_15m_15*	*Ub_TS+CD_15m_30*	*Ub_TS+CD_15m_45*	*Ub_TS+CD_15m_60*	**Subtotal Ub**	**Total Fw + Ub**
M5	8	0	0	0	0	4	0	0	0	0	12	1	0	0	0	1	5	12	6	25	37
Linear Regression	1	0	0	0	0	7	0	0	0	0	8	0	0	0	0	0	0	0	0	0	8
Random Committee	2	0	0	0	0	0	0	3	0	0	5	0	0	0	0	0	0	0	0	0	5
Bagging	3	0	0	1	0	0	0	1	1	0	6	2	4	4	2	3	4	2	6	27	33
Additive Regression	1	0	1	0	2	0	0	0	1	1	6	2	1	2	1	2	2	0	0	10	16
Random Forest	0	7	4	2	5	2	5	2	2	1	30	7	3	4	5	8	1	0	1	29	59
Random Committee	0	6	2	1	3	0	0	0	0	2	14	2	5	3	3	1	1	0	0	15	29
Vote	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1
LWL	0	1	0	0	0	1	0	0	2	1	5	1	0	1	1	0	0	0	0	3	8
IBk	0	0	6	10	5	0	6	5	7	6	45	0	1	1	0	0	0	0	0	2	47
KStar	0	0	1	0	0	0	3	3	1	3	11	0	0	0	0	0	0	0	0	0	11
SMOreg	0	0	1	1	0	0	1	0	0	1	4	0	0	0	0	0	0	0	0	0	4
Random Subspace	0	0	0	0	0	1	0	1	1	0	3	0	0	0	1	0	0	0	0	1	4
Gaussian Processes	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	2	2
J48	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	1	1	2	6	6

To finalize with this section, we analyse the ML methods selected by Auto-WEKA over all data-sets. Table 4 summarises how many times an algorithm is selected to forecast traffic speed along the data-sets. It is important to clarify that Auto-WEKA has a base of 39 algorithms and the ones that were not suggested for the data-sets evaluated are not included in Table 4. As each data-set was evaluated with three Auto-WEKA's running times along five repetitions in each of them, one algorithm can be chosen at most 15 times per data-set.

According to the results of Table 4, for freeway data-sets, IBk (k-NN) is the most selected method with the exception of data-sets $Fw\_T+CD\_5m\_5$, $Fw\_T+CD\_5m\_15$ and $Fw\_TS+CD\_5m\_5$ wherein two Tree-based algorithms ($M5$ and RF) and one regression algorithm are the most chosen. On the other hand, for urban data-sets, Tree-based algorithms (RF and $M5$) are the most chosen algorithms, excluding the data-set $Ub\_T+CD\_15m\_30$ in which the method with the highest frequency is $Bagging$. In the cases of data-sets $Ub\_T+CD\_15m\_45$ and $Ub\_TS+CD\_15m\_60$, Tree-based algorithms got the selection frequency with Ensemble methods ($RandomCommittee$ and $Bagging$).

In general, the three most chosen algorithms (RF, IBk, $M5$) along all data-sets belong to Tree-based and Lazy families of methods. This is in concordance with the results obtained by Angarita et. al [2] wherein RF was the most selected ML algorithm to address the TF classification problem approached by the authors. Furthermore, the relevance of hyper-parameter tuning can be appreciated through the case of IBk algorithm. Concretely, the instance of this method within the BAs $(k-NN)$ did not achieve competitive results; however, in the case of Auto-WEKA, its performance was improved, without the need of human effort. The better adjustment of hyper-parameters done by the AutoML method, make the IBk algorithm be among the three most selected methods.

5 Conclusions

In this paper, we have focused on deepening into the benefits of AutoML for supervised regression in the field of TF. To accomplish such purpose, we have compared to what extent the results of AutoML differ from the general approach in TF. We used Auto-WEKA as AutoML method and *NN*, *SVM*, *k-NN* and *RF* as BAs. Concretely, our comparisons were made based on predicting traffic speed, over multiple time horizons ahead, for two different scenarios for the TF regression problems with different instances of them. The first type corresponds to the prediction of traffic at a target location, in a freeway environment; on the one hand, using only past traffic data of the target location, and, on the other hand, considering historical traffic data coming from the target location and from four downstream positions. The second type of TF problems is focused on forecasting traffic speed within an urban context, using the same variants as for the freeway scenario described above.

From the results we drawn interesting conclusions. From a Computer Science perspective, the AutoML method improves three out of the four BAs, and obtains similar results to RF (the best BA) without statistically significant differences. With a lower human effort, the user can expect similar o even better results (in the case of urban data-sets) than the best BA. Besides, another interesting conclusion is that higher execution times for Auto-WEKA not always leads to better results as we can expect. With some preliminary tests we detected that is was due to over-fitting issues.

From a transportation approach, Tree-based algorithms and IBk are suitable methods to make predictions, in freeway contexts, at a target location using either traffic data obtained only from that location or data also provided by other locations surrounding the target position. On the other hand, Tree-based and Ensembles algorithms seem to be the best approach to forecast traffic speed for urban environments.

Further research lines that we aim to explore in the future are: I) comparing optimisation and meta-learning strategies to find the best pair (algorithm and hyper-parameter setting); II) integrating data pre-processing techniques to the AutoML process; and III) exploring the benefits and recommendations of algorithms done by AutoML methods at the moment of approaching more complex families of TF regression problems.

Acknowledgements This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 636220 and the Marie Sklodoska-Curie grant agreement No. 665959.

References

1. Angarita-Zapata, J.S., Masegosa, A.D., Triguero, I.: A taxonomy of traffic forecasting regression problems from a supervised learning perspective. IEEE Access pp. 1–1 (2019). DOI 10.1109/ACCESS.2019.2917228
2. Angarita-Zapata, J.S., Triguero, I., Masegosa, A.D.: 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, pp. 204–214. Springer International Publishing (2018)
3. Charte, D., Charte, F., García, S., Herrera, F.: A snapshot on nonstandard supervised learning problems: taxonomy, relationships, problem transformations and algorithm adaptations. Progress in Artificial Intelligence pp. 1–14 (2018)
4. Chen, J.F., Lo, S.K., Do, Q.H.: Forecasting Short-Term Traffic Flow by Fuzzy Wavelet Neural Network with Parameters Optimized by Biogeography-Based Optimization Algorithm. In: Computational Intelligence and Neuroscience, pp. 1–13 (2018)
5. Chen, W., An, J., Li, R., Fu, L., Xie, G., Bhuiyan, M.Z.A., Li, K.: A novel fuzzy deep-learning approach to traffic flow prediction with uncertain spatial-temporal data features. Future Generation Computer Systems **89**, 78 – 88 (2018)
6. Ermagun, A., Levinson, D.: Spatiotemporal traffic forecasting: review and proposed directions. Transport Reviews **38**(6), 786–814 (2018)
7. Feurer, M., Klein, A., Eggensperger, K., Springenberg, J., Blum, M., Hutter, F.: Efficient and Robust Automated Machine Learning. In: Advances in Neural Information Processing Systems 28, pp. 2962–2970 (2015)
8. Garcia, S., Fernandez, A., Luengo, J., Herrera, F.: 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), 2044 – 2064 (2010)
9. Howell, S.: Meta-analysis of machine learning approaches to short-term urban traffic prediction. In: Scottish Transport Applications and Research Conference, pp. 1–15. STAR (2018)
10. I. Laña and J. Del Ser and M. Velez and E. I. Vlahogianni: Road Traffic Forecasting: Recent Advances and New Challenges. IEEE Intelligent Transportation Systems Magazine **10**(2), 93–109 (2018)
11. Karlaftis, M., Vlahogianni, E.: Statistical methods versus neural networks in transportation research: Differences, similarities and some insights. Transportation Research Part C: Emerging Technologies **19**(3), 387 – 399 (2011)
12. Kerner, B.S.: The physics of traffic. Physics World **12**(8), 25–30 (1999)
13. Kerschke, P., Hoos, H., Neumann, F., Trautmann, H.: Automated Algorithm Selection: Survey and Perspectives. Evolutionary Computation **27**, 1–47 (2018)
14. Komer, B., Bergstra, J., Eliasmith, C.: Hyperopt-sklearn: automatic hyperparameter configuration for scikit-learn. In: Proceedings of SciPy, p. 33–39 (2014)
15. Liu, Z., Guo, J., Cao, J., Wei, Y., Huang, W.: A Hybrid Short-term Traffic Flow Forecasting Method Based on Neural Networks Combined with K-Nearest Neighbor. Journal on Traffic and Transportation Technology **30**(4), 445–456 (2018)
16. Lopez-Garcia, P., Onieva, E., Osaba, E., Masegosa, A.D., Perallos, A.: A Hybrid Method for Short-Term Traffic Congestion Forecasting Using Genetic Algorithms and Cross Entropy. IEEE Transactions on Intelligent Transportation Systems **17**(2), 557–569 (2016)
17. Luo, G.: A review of automatic selection methods for machine learning algorithms and hyper-parameter values. Network Modeling Analysis in Health Informatics and Bioinformatics **5**(1), 5–18 (2016)

18. Ma, X., Yu, H., Wang, Y., Wang, Y.: Large-scale transportation network congestion evolution prediction using deep learning theory. PloS one **10**(3), 1–17 (2015)
19. Mohr, F., Wever, M., Hüllermeier, E.: ML-Plan: Automated machine learning via hierarchical planning. Machine Learning **107**(8), 1495–1515 (2018)
20. Oh, S., Byon, Y.J., Jang, K., Yeo, H.: Short-term Travel-time Prediction on Highway: A Review of the Data-driven Approach. Transport Reviews **35**(1), 4–32 (2015)
21. Olson, R.S., Bartley, N., Urbanowicz, R.J., Moore, J.H.: Evaluation of a Tree-based Pipeline Optimization Tool for Automating Data Science. In: Proceedings of the Genetic and Evolutionary Computation Conference 2016, pp. 485–492. ACM, New York, NY, USA (2016)
22. Park, J., Murphey, Y.L., McGee, R., Kristinsson, J.G., Kuang, M.L., Phillips, A.M.: Intelligent Trip Modeling for the Prediction of an Origin–Destination Traveling Speed Profile. IEEE Transactions on Intelligent Transportation Systems **15**(3), 1039–1053 (2014). DOI 10.1109/TITS.2013.2294934. URL http://ieeexplore.ieee.org/document/6728714/
23. Pell, A., Meingast, A., Schauer, O.: Trends in real-time traffic simulation. Transportation Research Procedia **25**, 1477 – 1484 (2017)
24. Rahimipour, S., Moeinfar, R., Hashemi, S.M.: Traffic prediction using a self-adjusted evolutionary neural network. Journal of Modern Transportation pp. 1–11 (2018)
25. Sabharwal, A., Samulowitz, H., Tesauro, G.: Selecting Near-optimal Learners via Incremental Data Allocation. In: Proceedings of the Thirtieth Conference on Artificial Intelligence, pp. 2007–2015 (2016)
26. Skycomp, I.B.M.: Major High- way Performance Ratings and Bottleneck Inventory. Maryland State Highway Administration, the Baltimore Metropolitan Council and Maryland Transportation Authority, State of Maryland (2009)
27. Sparks, E.R., Talwalkar, A., Haas, D., Franklin, M.J., Jordan, M.I., Kraska, T.: Automating Model Search for Large Scale Machine Learning. In: Proceedings of System-on-Chip Conference, pp. 368–380 (2015)
28. Swearingen, T., Drevo, W., Cyphers, B., Cuesta-Infante, A., Ross, A., Veeramachaneni, K.: ATM: A distributed, collaborative, scalable system for automated machine learning. In: 2017 IEEE International Conference on Big Data, pp. 151–162 (2017)
29. Thornton, C., Hutter, F., Hoos, H.H., Leyton-Brown, K.: Auto-WEKA. In: Proceedings of the 19th International conference on Knowledge discovery and data mining, pp. 847–855 (2013)
30. Vlahogianni, E.I.: Enhancing Predictions in Signalized Arterials with Information on Short-Term Traffic Flow Dynamics. Journal of Intelligent Transportation Systems **13**(2), 73–84 (2009)
31. Vlahogianni, E.I.: Optimization of traffic forecasting: Intelligent surrogate modeling. Transportation Research Part C: Emerging Technologies **55**, 14 – 23 (2015)
32. Vlahogianni, E.I., Golias, J.C., Karlaftis, M.G.: Short-term traffic forecasting: Overview of objectives and methods. Transport Reviews **24**(5), 533–557 (2004)
33. Vlahogianni, E.I., Karlaftis, M.G., Golias, J.C.: Short-term traffic forecasting: Where we are and where we're going. Transportation Research Part C: Emerging Technologies **43**, 3–19 (2014)
34. Xia, J., Huang, W., Guo, J.: A clustering approach to online freeway traffic state identification using ITS data. Journal of Civil Engineering **16**(3), 426–432 (2012)
35. Yao, Q., Wang, M., Chen, Y., Dai, W., Yi-Qi, H., Yu-Feng, L., Wei-Wei, T., Qiang, Y., Yang, Y.: Taking Human out of Learning Applications: A Survey on Automated Machine Learning. CoRR (2018)
36. Yu, H., Wu, Z., Wang, S., Wang, Y., Ma, X.: Spatiotemporal Recurrent Convolutional Networks for Traffic Prediction in Transportation Networks. Sensors **17**(7) (2017)
37. Zöller, M.A., Huber, M.F.: Survey on Automated Machine Learning. CoRR (2019)