A Preliminary Study on Automatic Algorithm Selection for Short-Term Traffic Forecasting

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

Abstract Despite the broad range of Machine Learning (ML) algorithms, there are no clear baselines to find the best method and its configuration given a Short-Term Traffic Forecasting (STTF) problem. In ML, this is known as the Model Selection Problem (MSP). Although Automatic Algorithm Selection (AAS) has proved success dealing with MSP in other areas, it has hardly been explored in STTF. This paper deepens into the benefits of AAS in this field. To this end, we have used Auto-WEKA, a well-known AAS method, and compared it to the general approach (which consists of selecting the best of a set of algorithms) over a multi-class imbalanced classification STTF problem. Experimental results show AAS as a promising methodology in this area and allow important conclusions to be drawn on how to improve the performance of ASS methods when dealing with STTF.

1 Introduction

Urban development and population growth have risen the levels of congestion that in turn generate socio-economic and environmental issues. One contemporary policy

Juan S. Angarita-Zapata
Faculty of Engineering, University of Deusto, Av. Universidades, 24, 48007, Bilbao
DeustoTech-Fundación Deusto, Deusto Foundation, Av. Universidades, 24, 48007, Bilbao
e-mail: js.angarita@deusto.es

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

Antonio D. Masegosa
Faculty of Engineering, University of Deusto, Av. Universidades, 24, 48007, Bilbao
DeustoTech-Fundación Deusto, Deusto Foundation, Av. Universidades, 24, 48007, Bilbao
IKERBASQUE, Basque Foundation for Science, 48011, Bilbao, Spain
e-mail: ad.masegosa@deusto.es

to deal with congestion is the development of STTF systems. STTF is the prediction of near future traffic measures for fixed locations, road segments, or entire links [15]; which in consequence allows users to plan ahead their movements along the roads.

The recent emergence of telecommunications technologies integrated to transportation infrastructure generates vast volumes of traffic data. Such unprecedented data availability and growing computational capacities have increased the use of ML to approach STTF. The main strength of ML, with respect to traffic theory models, is its ability to predict short-term traffic using current and historical data without the need of knowing theoretical traffic mechanisms.

The literature on STTF reports a great variety of ML algorithms applications as Neural Networks (NNs), Support Vector Machines (SVMs), k-Nearest Neighbors (k-NN) or Random Forest (RF) [5, 15]. Nevertheless, given the broad range of ML methods there are no baselines to select the most appropriate algorithm and its best hyper-parameter setting given the characteristics of an STTF problem. In ML, this is known as the MSP, and AAS has been one of the most successful approaches to address it so far. It aims at automatically finding the ML algorithm and hyper-parameters configuration pair which maximize a performance measure on given data, using an optimization strategy that minimizes a predefined loss function.

Although AAS methods have approached the MSP with high performance in other research areas [7], to the best of authors' knowledge only the work in [14] has tackled the MSP in STTF. The proposed AAS method aims at predicting average speed in a time horizon of 5 minutes using a time series regression approach. In this research, we conduct a preliminary study to keep exploring the benefits of AAS in STTF focusing on a classification STTF problem in multiple time horizons and using a different AAS method, Auto-WEKA. To this end, we compare the AAS method versus the general approach in STTF, which consists of selecting the best of a set of commonly used ML algorithms. Concretely, we compare Auto-WEKA with four state-of-the-art ML algorithms (NN, SVM, k-NN and RF) in the task of forecasting traffic Level of Service (LoS) using real data.

The rest of this paper is structured as follow. Section 2 presents related work in ML and AAS algorithms applied to STTF. Section 3 shows the methodology of this research. Then, Section 4 exposes results followed by the Conclusions in Section 5.

2 Related Work

This section reviews literature related to ML and AAS in the context of STTF. We start by describing how STTF can be addressed from a ML perspective. Then, ML methods for STTF are discussed, and finally we review existing AAS methods.

Short-Term Traffic Forecasting from a Machine Learning perspective

In recent years, STTF is being influenced by the great availability of data provided by Intelligent Transportation Systems. 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). These technologies and contemporary computational advances have caused a leap in the way STTF is approached switching from a traffic theory-based perspective to a data-driven one, with a special focus on ML. In this work, we center on ML applied to VDS data, because it is the most common type of data available and used in literature [9].

From a ML perspective, STTF is approached by building a model from traffic historical data to make predictions on new and unseen data. Depending on the type of input and output (predicted) data, different ML approaches can be used. When the traffic measure to be forecasted is continuous (e.g. speed or flow) it should be dealt as a regression problem. In the case that input variables are ordered by time, the approach is time series regression, which requires defining the prior time steps and the number of lagged variables to predict the forecasted traffic measure. On the other hand, when the predicted value is discrete, the prediction should be addressed as a classification problem (e.g. Traffic LoS).

ML algorithms for STTF using VDS data

ML methods applied to STTF can be categorized into single or hybrid. The first type corresponds to adaptations of existing ML algorithms and in turn, they can be classified as parametric and non-parametric. 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 used are NNs, SVMs, k-NN, and RF [15]. As mentioned before, the other approach of ML algorithms is hybridization. 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 [8], where authors integrate a Boltzmann Machine with Recurrent NNs, and [6], where Genetic Algorithms are integrated with Fuzzy Systems.

Nevertheless, despite the great variety of ML methods, dealing with the MSP in STTF is not a trivial task, as mentioned before. The general approach to tackle the MSP in STTF 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, AAS has received a lot of attention in ML because of its promising results in dealing with the MSP with low human intervention.

Automatic ML algorithm selection in STTF

As stated above, AAS method deals with MSP as an optimization problem whose objective consists of finding the ML algorithm, from a pre-defined base of algorithms, and its hyper-parameter configuration that maximizes an accuracy measure on a given ML problem. The first method in tackling simultaneously the selection of algorithm and hyper-parameters in ML was Auto-WEKA [13]. It uses Bayesian optimization to search for the best pair (algorithm, hyper-parameter setting) and its base of algorithms are the 39 ML methods implemented in WEKA (a well-known open-source ML software that contains algorithms for data analysis and predictive modelling). Subsequently, Komer et al. [3] and Feurer et al. [1] developed Hyperopt-sklearn and Auto-sklearn, respectively. They automatically select ML algorithms and hyper-parameter values from scikit-learn[1]. In the case of [3], the AAS method uses Hyperopt Python library for the optimization 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 chooses a starting point for a sequential optimization process. More recently, Sparks et al. [12] proposed a method that supports distributed computing for AAS, and Sabharwal et al. [10] 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 the combination is promising. For this research, we select Auto-WEKA because of the wider variety of the base of algorithms in comparison with the others approaches reviewed. Furthermore, unlike the methods presented by Sparks et al. and Sabharwal et al. that only consider a pre-defined set of hyper-parameters combinations, Auto-WEKA has no limitations in the hyper-parameter space to be explored.

To the best authors' knowledge, only one work has tackled the MSP in STTF [14]. In this work, Vlahogianni proposed a meta-modelling technique that, based on surrogate modelling and a genetic algorithm with an island model, optimizes both the algorithm selection and the hyper-parameter setting. The AAS 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 time series regression approach. The main differences between this work and Vlahogianni's one lay on the addressed problem and the AAS method. Regarding the problem, we predict traffic LoS along multiple time horizons using a classification approach; and with respect to the method, we use an AAS method that considers a much broader base of algorithms, which is an important aspect in the MSP as we will discuss later.

3 Methodology

This research seeks to keep exploring the benefits AAS can bring to STTF. To accomplish this, we compare to what extent the results of AAS differ from the general approach in STTF, in which a set of Reference Algorithms (RAs) is tested over the forecasting problem in hand and the one with best performance metrics is chosen. We select Auto-WEKA as AAS method for the reasons explained above; and NN, SVM, $k-NN$, and RF as the RAs that represent the general approach. These algorithms are the most commonly used one in recent STTF literature. Due to space limitations, the details of Auto-WEKA are omitted. The interested reader is referred

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

Table 1 Speed and Density data sets

Data-sets	# Instances	# Attributes	# Instances per class	Imbalance Ratios
SD_5m_5, SD_5m_15, SD_5m_30, SD_5m_45, SD_5m_60	9979	12	A = 4534, B = 3681, C = 867, D = 891, E = 6	A/B=1.2; A/C=5.2; A/D=5.1; A/E=755.7
SD_1h_60, SD_1h_120	2159	7	A = 984, B = 790, C = 268, D = 117	A/B=1.2; A/C=3.7; A/D=8.4
DD_5m_5, DD_5m_15, DD_5m_30, DD_5m_45, DD_5m_60	9979	14	A = 2194, B = 559, C = 1075, D = 3541, E = 261	D/A=1.61; D/B=6.3; D/C=3.3; D/E=1.3
DD_1h_60, DD_1h_120h	2158	9	A = 471, B = 125, C = 267, D = 822, E = 473	D/A=1.7; D/B=6.6; D/C=3.07; D/E=1.7

to [13] for further details. The following part of the section is devoted to exposing how short-term forecasting of traffic LoS can be approached as a classification problem; as well as to describe the data-sets and the experimental set-up used.

Short-term forecasting of traffic service quality as a classification problem

In this work, STTF is focused on predicting the quality of traffic service, at a specific location, through a categorical measure named LoS. For freeways, LoS is used to categorize the quality levels of traffic, through letters from A to E in a gradual way[2], based on performance measures such as speed, density, and volume/capacity [11].

In this sense, we are approaching the forecasting of traffic service quality from a classification approach, concretely, as a multi-class classification problem. Based on speed and density data (calculated as flow/speed), which are continuous variables, we estimate how congested will be the road at the detector location, in different short-term time horizons through the LoS measure. This categorical measure is estimated from two univariate approaches, which means that speed and density are independently used to predict LoS using the intervals defined for them in [11].

It is important to clarify that the forecasting of LoS could be also addressed as a regression or time series problem predicting either speed or density and then discretizing the results to obtain a categorical interval of LoS. However, we chose the classification approach to explore a different problem to the one published in [14], to deepen into the benefits of AAS in a different area of STTF.

Data-sets and Experimental set-up

Data used in this work is provided by PeMS[3]. According to recent literature, this data source is highly used in the area of STTF because of its high quality data, availability of various traffic measures and its public accessibility. The route selected for our experiments is the California Interstate I405-N. Particularly, we focus on the detector VDS 771826 located at the post-mile 0.11 on this freeway. Traffic measures collected from the detector are speed and flow in aggregation times of 5 minutes and 1 hour for both measures.

[2] Category *A* indicates light to moderate traffic, whereas a category *E* means extended delays

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

From this data, we generate 14 data-sets: seven using speed and seven using density as traffic data, respectively. Time horizons in which LoS is predicted are 5, 15, 30, 45, 60, and 120 minutes using data granularity of 5 minutes or 1 hour depending on the case (granularity means how often the traffic measures are taken and aggregated). To better identify the data-sets, they are named following the next structure: *TrafficData_Granularity_TimeHorizon*.

Attributes of data-sets with 5 minutes granularity are *Day of the week; Hour of the day; Minute of the Hour; Quarter hour of the day; traffic measure at past 5, 15, 30, 45, and 60 minutes; current traffic measure; and current LoS*, where traffic measure could be average speed or density depending on the respective data-set. In the case of data-sets with 1 hour granularity, these are *Day of the week; Hour of the day; traffic measure at past 1 and 2 hours; current traffic measure; and current LoS*; again, the traffic measure could be average speed or density. In addition, Table 1 presents the number of instances and attributes of each data-set, together with the Imbalance Ratios (IRs) calculated dividing the number of instances of the majority class over the number of instances of each of the rest of classes. IR values show that the generated data-sets are imbalanced, although with different degrees.

For the experimentation with Auto-WEKA, three execution times were considered: 15, 150, and 300 minutes. These correspond to the time that the method can take 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 RAs, we test them using WEKA. The process of evaluating every RA over a data-set was done with 20 repetitions with different initial seeds, and using the default hyper-parameter setting offered by WEKA. We have not performed any optimization or extra-adjustment of the RAs' hyper-parameters because our aim is to compare the performance of AAS versus RAs 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 AAS method and the RA using the metric *G-measure (mGM)* that is applied for multi-class imbalanced data in classification problems [4]. Its calculation is expressed as $mGM = \sqrt[M]{\prod_{i=1}^{M} precision_i \cdot recall_i}$ wherein $G-measure$ on $i-th$ class is estimated as $GM_i = \sqrt{precision_i \cdot recall_i}$, and M is the total number of classes.

Table 2 shows the mean and standard deviation (between brackets) of the *mGM* values obtained by both the RAs and Auto-WEKA over all repetitions for each data-set. *mGM* values in bold indicate the best result in every data-set achieved from either any of the RAs or any of the Auto-WEKA's execution times. As it can be seen, in general, the AAS method performs better than $k-NN$, NN and SVM; but worse than RF that is the best RA along most of the data-sets. Nevertheless, the

Table 2 Mean *mGM* values and their standard deviations (in brackets) obtained for density and speed data-sets by RA and AAS method.

	Reference Algorithms				**Auto-WEKA**		
Data-sets	k-NN	NN	RF	SVM	15mET	150mET	300mET
DD_5m_5	0.485 (0.02)	0.577 (0.03)	**0.585 (0.02)**	0.523 (0.02)	0.564 (0.02)	0.563 (0.02)	0.563 (0.02)
DD_5m_15	0.47 (0.02)	0.531 (0.03)	**0.559 (0.01)**	0.465 (0.02)	0.495 (0.02)	0.485 (0.01)	0.529 (0.02)
DD_5m_30	0.433 (0.01)	0.472 (0.03)	**0.535 (0.01)**	0.339 (0.02)	0.462 (0.02)	0.451 (0.02)	0.463 (0.04)
DD_5m_45	0.424 (0.02)	0.460 (0.03)	**0.530 (0.01)**	0.332 (0.03)	0.460 (0.02)	0.478 (0.03)	0.479 (0.03)
DD_5m_60	0.436 (0.01)	0.441 (0.03)	**0.531 (0.01)**	0.356 (0.01)	0.427 (0.03)	0.455 (0.06)	0.481 (0.05)
DD_1h_60	0.601 (0.02)	0.568 (0.05)	**0.679 (0.02)**	0.476 (0.13)	0.622 (0.01)	0.637 (0.03)	0.646 (0.03)
DD_1h_120	0.498 (0.03)	0.445 (0.04)	0.576 (0.03)	0.36 (0.03)	0.543 (0.02)	0.557 (0.02)	**0.578 (0.01)**
SD_5m_5	0.671 (0.04)	0.773 (0.01)	**0.775 (0.02)**	0.756 (0.01)	0.746 (0.03)	0.763 (0.03)	0.764 (0.03)
SD_5m_15	0.587 (0.04)	0.557 (0.03)	**0.610 (0.02)**	0.499 (0.02)	0.572 (0.01)	0.571 (0.02)	0.572 (0.02)
SD_5m_30	0.528 (0.02)	0.394 (0.06)	**0.566 (0.02)**	0.322 (0.02)	0.453 (0.07)	0.469 (0.07)	0.452 (0.05)
SD_5m_45	0.454 (0.05)	0.356 (0.06)	**0.545 (0.01)**	0.572 (0.07)	0.408 (0.0)	0.471 (0.08)	0.453 (0.06)
SD_5m_60	0.467 (0.02)	0.368 (0.10)	**0.550 (0.01)**	0.538 (0.01)	0.461 (0.0)	0.469 (0.02)	0.482 (0.03)
SD_1h_60	0.424 (0.03)	0.473 (0.03)	0.482 (0.04)	0.40 (0.03)	0.532 (0.02)	**0.553 (0.03)**	0.462 (0.03)
SD_1h_120	0.286 (0.03)	0.322 (0.08)	0.366 (0.03)	0.259 (0.13)	0.372 (0.03)	**0.422 (0.03)**	0.399 (0.03)

improvement of *RF* w.r.t Auto-WEKA is small in most cases, ranging from 0.02 to 0.097, being even negative in three cases ($DD\_1h\_120, SD\_1h\_60, SD\_1h\_60$) where the AAS method obtains better *mGM* values than *RF*. This result is interesting because, in order to get to the conclusion that *RF* is the best RA, the user should run all RAs over all data-sets and compare their performance. However, according to these results, running Auto-WEKA only once, and therefore employing less human effort, the user can expect results very similar to the best RA and even better in some cases. Regarding data-sets characteristics, we can see that they do influence the differences between results of Auto-WEKA and *RF*. Concretely, for data-sets with a granularity of 5 minutes, for both traffic and density data-sets, the divergences between these two methods are greater for long-term time (in favour of *RF*). If we take into account the granularity of the data-sets, Auto-WEKA works especially well on those with 1 hour granularity improving *RF* in all cases except DD_1h_120.

Another interesting aspect is the relation between the execution time and the performance of the models provided by Auto-WEKA. For density data-sets, longer execution times contribute to obtaining better results although the improvements are very small. In the case of speed data-sets, 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. Through some analyses, 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 AAS method to deal with over-fitting, especially when execution times are high.

Another important aspect is the low performance of Auto-WEKA and the RAs, with *mGM* values below 0.7 in all data-sets, except for SD_5m_5. In analyses that are not included here because of space limitations, we corroborated that this behavior is mainly due to the high imbalance of the data-sets, especially in density ones, and the poor accuracy and recall of the methods when predicting the minority classes. This situation shows us that in the design of AAS methods for STTF problems, it

Table 3 Average Rankings of Friedman Test and Adjusted p-Values obtained through Holm post-hoc test

	Friedmans's Avg. Ranking			**Holm's adj. p-values**		
Algorithms	ET_{15min}	ET_{150min}	ET_{300min}	ET_{15min}	ET_{150min}	ET_{300min}
k-NN	3.5	3.7143	3.6429	**3.93e-4**	**8.6e-5**	**1.45e-4**
NN	3.1786	3.2857	3.2143	**2.026e-3**	**1.056e-3**	**1.636e-3**
RF	1.2143	1.2143	1.2143	-	-	-
SVM	4.2857	4.3571	4.3571	**1e-6**	**1e-6**	**1e-6**
Auto-WEKA	2.8214	2.4286	2.5714	**7.161e-3**	**4.2165e-2**	**2.3151e-2**

is necessary to include mechanisms that allow addressing imbalance either by data pre-processing techniques or by adjusting the hyper-parameters of the ML methods.

To assess whether the differences in performance observed among the RAs and Auto-WEKA variants are significant or not, we made use of non-parametric statistical tests. Two statistical tests have been applied, following the guidelines proposed in [2]. First, the Friedman's test for multiple comparisons has been applied to check whether there are significant differences among the studied methods. Given that the p-value returned by these tests is 0, the null hypothesis can be rejected in all cases. The mean ranking returned by this test is displayed in Table 3, confirming the better global results of *RF* against both the rest of RAs and Auto-WEKA, and also the better global results of Auto-WEKA versus $k-NN$, NN and SVM.

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 with respect to the other RAs and Auto-WEKA. Table 3 also presents the adjusted p-values returned by this test. In order to highlight significant differences, those p-values lower than 0.05 are in bold. Looking the Table 3 there are important differences in the test's outcomes. It can be said that *RF* results improve significantly all the RAs including Auto-WEKA with their three execution times, although their p-values are greater than the ones of the other RAs.

To finalize with this section, we analyze the classifiers selected by Auto-WEKA over all data-sets. Table 4 summarizes how many times an algorithm is selected to forecast congestion 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. In Table 4, *RF* is in general the most chosen algorithm except for DD_5m_5, where it is the second most selected algorithm behind *Logistic*, and for SD_1h_60 and SD_1h_120 where *RF* is not chose any time. In those data-sets Auto-WEKA improves *RF* because is able to find alternative methods that work better than *RF*, especially $DecisionTable$ and LMT. Another interesting fact is that the RAs *k-NN* and *SVM* (named by Auto-WEKA as *IBk* and *SMO*, respectively) are only chosen once and four times, in that order, and *NN* is not suggested any time despite of being three of the most widely used algorithms in literature. Moreover, in the case of the base of algorithms used in [14], only SVM is

Table 4 Classifiers selected by Auto-WEKA and absolute frequency in which they were suggested for density and speed data-sets

Classifiers	*DD_5m_5*	*DD_5m_15*	*DD_5m_30*	*DD_5m_45*	*DD_5m_60*	*DD_1h_60*	*DD_1h_120*	***Subtotal DD***	*SD_5m_5*	*SD_5m_15*	*SD_5m_30*	*SD_5m_45*	*SD_5m_60*	*SD_1h_60*	*SD_1h_120*	***Subtotal SD***	***Total DD+SD***
BayesNet	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	2	2
Logistic	9	0	0	0	0	0	0	9	2	0	0	0	0	0	0	2	11
SMO	0	0	1	1	0	0	0	2	0	2	0	0	0	0	0	2	4
IBk	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	1
Kstar	0	1	0	0	0	0	0	1	0	3	4	3	0	0	0	10	11
LWL	0	0	0	0	0	0	0	0	0	1	0	0	0	1	3	5	5
AdaBoostM1	0	1	0	0	0	0	0	1	0	0	1	0	0	1	0	2	3
Bagging	0	2	1	1	1	2	3	10	3	0	0	0	0	2	2	7	17
RandomCommittee	0	0	0	0	1	2	1	4	1	0	0	0	2	3	1	7	11
RandomSubSpace	2	1	2	1	3	0	0	9	2	0	0	0	0	0	0	2	11
Vote	0	1	0	0	0	1	1	3	0	3	1	0	1	1	1	7	10
DecisionTable	0	1	0	0	0	0	1	2	0	0	0	0	0	0	6	6	8
J48	0	0	0	0	2	0	0	2	0	0	0	0	0	1	1	2	4
LMT	0	0	3	2	0	2	2	9	2	0	0	0	0	4	1	7	16
RandomForest	4	8	8	10	8	8	7	53	5	5	9	12	12	0	0	43	96

selected by Auto-WEKA, which in our opinion stands out the importance of a broad algorithm base to perform AAS in STTF.

5 Conclusions

In this paper, we have focused on deepening into the benefits of AAS in the field of STTF. To accomplish this, we have compared to what extent the results of AAS differs from the general approach in STTF. We used Auto-WEKA as AAS method and *NN*, *SVM*, *k-NN* and *RF* as RAs representatives of the general approach. Concretely, our comparisons were made based on a multi-class imbalanced problem, consisting on the prediction of traffic LoS, at a fixed location, over different time horizons ahead. From the results we drawn interesting conclusions: the AAS method improves three out of the four RAs, and obtain similar results to *RF*, the best RA; with a lower human effort, the user can expect similar o even better results than the best RA;higher execution times for Auto-WEKA not always leads to better results due to over-fitting issues; and the performance shown by the RAs and Auto-WEKA was poor in general because they shown problems to deal with imbalance data.

Further research lines that we aim to explore in the future are: I) including mechanisms, within the base of algorithms used by the AAS method, to deal with imbalanced data-sets and over-fitting; II) testing different optimization strategies for finding the best pair (algorithm and hyper-parameter setting); III) integrating more data preprocessing techniques to the AAS process; and IV) approaching the forecasting of LoS also as a time series regression problem to explore what algorithms are more suitable depending on the modelling approach chosen.

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. This work has been also supported by the research projects TIN2014-56042-JIN from the Spanish Ministry of Economy and Competitiveness.

References

1. 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)
2. 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)
3. Komer, B., Bergstra, J., Eliasmith, C.: Hyperopt-sklearn: automatic hyperparameter configuration for scikit-learn. In: Proceedings of SciPy, p. 33–39 (2014)
4. Krawczyk, B., McInnes, B.T., Cano, A.: Sentiment classification from multi-class imbalanced twitter data using binarization. In: Hybrid Artificial Intelligent Systems, pp. 26–37 (2017)
5. Lana, I., Ser, J.D., Velez, M., Vlahogianni, E.I.: Road traffic forecasting: Recent advances and new challenges. IEEE Intelligent Transportation Systems Magazine **10**(2), 93–109 (2018)
6. 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)
7. 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)
8. Ma, X., Yu, H., Wang, Y., Wang, Y.: Large-scale transportation network congestion evolution prediction using deep learning theory. PloS one **10**(3), e0119,044 (2015)
9. 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)
10. Sabharwal, A., Samulowitz, H., Tesauro, G.: Selecting near-optimal learners via incremental data allocation. In: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, AAAI'16, pp. 2007–2015 (2016)
11. 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)
12. 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 SoCC'15, pp. 368–380 (2015)
13. Thornton, C., Hutter, F., Hoos, H.H., Leyton-Brown, K.: Auto-WEKA. In: Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '13, pp. 847–855 (2013)
14. Vlahogianni, E.I.: Optimization of traffic forecasting: Intelligent surrogate modeling. Transportation Research Part C: Emerging Technologies **55**, 14 – 23 (2015)
15. 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)