Available online at www.sciencedirect.com

ScienceDirect

Transportation Research Procedia 14 (2016) 4458 – 4467

Transportation Research Procedia

www.elsevier.com/locate/procedia

6th Transport Research Arena April 18-21, 2016

A comparative study on the performance of evolutionary fuzzy and crisp rule based classification methods in congestion prediction

E. Onieva [a,*], P. Lopez-Garcia [a], A.D. Masegosa [a,b], E. Osaba [a], A. Perallos [a]

[a]*Deusto Institute of Technology (DeustoTech), University of Deusto, Bilbao 48007, Spain*
[b]*IKERBASQUE, Basque Foundation for Science, Bilbao 48011, Spain*

Abstract

Accurate estimation of the future state of the traffic is an attracting area for researchers in the field of Intelligent Transportation Systems (ITS). This kind of predictions can lead to traffic managers and drivers to act in consequence, reducing the economic and social impact of a possible congestion. Due to the inter-urban traffic information nature, the task of predicting the future state of the traffic requires, in most cases, a non-linear patterns search in the input data. In recent years, a wide variety of models has been used to solve this problem in the most accurate way. Due to that, models generated to provide information about the future state of the road are, usually, incomprehensible to a human operator, making impossible to give him/her an explanation about the causes of the prediction. Given the capacity of rule based systems to explain the reasoning followed to classify a new pattern, the advantages and disadvantages of such approaches are explored in this work.

To conduct such task, datasets recorded from the California Department of Transportation are created. A 9-kilometer section of the I5 highway of Sacramento is used for this research. Two different types of datasets are built for the experimentation. One of them contains the entire information recorded. The other one contains with a simplified version of the information, considering only the first, middle and last monitored points of the road. Twelve prediction horizons, from 5 to 60 minutes, were considered for prediction. An experimental comparative study involving 16 state of the art techniques is performed. Techniques tested include those that fall within the categories of Evolutionary Crisp Rule Learning (ECRL) and Evolutionary Fuzzy Rule Learning (EFRL). These methods were selected since they offer to the final user, not only a prediction, but also a legible model about the way in which the decision was taken. Techniques are compared in terms of accuracy and complexity of the models generated.

* Corresponding author.
E-mail address: enrique.onieva@deusto.es

Peer-review under responsibility of Road and Bridge Research Institute (IBDiM)
doi:10.1016/j.trpro.2016.05.368

Keywords: Intelligent Transportation Systems; traffic congestion prediction; traffic forecast; genetic algorithms; fuzzy logic; evolutionary fuzzy rule learning; evolutionary crisp rule learning

1. Introduction

Getting a fully sustainable mobility is one of the biggest challenges of modern traffic management. Sustainable mobility refers to social and ecological objectives associated with the transport. Today, traffic levels are reaching high values. This fact leads to serious problems associated with congestion, especially during peak hours (Steenbruggen et al., 2013). According to the European Commission, the share of road transport in total freight is at the level of 76.9% (http://epp.eurostat.ec.europa.eu). The current capacity of networks is not able to meet the growing demand, which causes congestion in urban areas and transit roads (Golinska and Hajdul, 2012). Congestion costs are estimated to increase by about 50%, to nearly 200 billion € annually (European Commission, 2011).

Therefore, the proper prediction of traffic congestion is an attracting area for researchers in the field of Intelligent Transportation Systems (ITS) (Cobo et al., 2014, Chen and Cheng, 2010). Accurate predictions can lead to traffic managers and drivers to act in consequence, reducing so the economic and social impact of the occurrence of congestion.

Over the last decades, the literature on short-term traffic flow forecasting has undergone great development (Hong, 2011). Many works describing a wide variety of different approaches have been published. Some of the most used methodologies lie in Kalman state space filtering models (Okutani and Stephanedes, 1984, Stathopoulos and Karlaftis, 2003), and the Autoregressive Integrated Moving Average (ARIMA) methodology, initially developed at (Box et al., 2008), and widely used since then. Given that the rapid variational process changes underlying traffic flow is complicated to be captured by a single linear statistical algorithm, more recent techniques such as Artificial Neural Networks (ANNs) (Schalkoff, 1997) or Support Vector Machines (SVMs) (Hearst et al. 1998) have proven their good performance in this task (Vlahogianni et al., 2005, Zhang et al., 1998).

The issue with imbalance in the class distribution became more pronounced with the applications of the machine learning algorithms to the real world. These applications range from telecommunications, bioinformatics, text classification, or speech recognition, to detection of oil spills in satellite images. The imbalance can be an artifact of class distribution and/or different errors applied over examples of different classes. A dataset is called imbalanced if it contains many more samples from one class than from the rest of the classes. Datasets are imbalanced when at least one class is represented by only a small number of training examples while other classes make up the majority (Ganganwar, 2012).

In this work, the problem of dealing with traffic information as a machine learning problem is considered. When dealing with traffic information, with the objective of detecting or predicting abnormal traffic situation, data collected became highly imbalanced, due to the reason that, in most of the time, the traffic will flow in a normal way. For this reason, it is normal not to find congestion or incidents in most of the time the road is being monitored.

The rest of the work is organized as follows. Section 2 presents the process used to capture and prepare data for the application of the selection of methods to study in this article. After that, Section 3 presents the methods to be tested. Section 4 is dedicated to explain the performance measures used to compare among techniques. In Section 5, results of the comparative study are shown, in terms of the parameters explained until then. Finally, some conclusions and future works are presented in Section 6.

2. Datasets used

Data used in this work was collected from the Performance Measurement System (PeMS) platform (http://pems.dot.ca.gov/). PeMS is a real-time database from the California Department of Transportation that offers over 10 years of historical traffic measurement for analysis. A 9-kilometers section of I5 highway in Sacramento, California, is used for this research.

A schematic graphic of the scenario used in this work is provided in Figure 1. In this figure, loop detectors are distributed in 13 points along the main road (MS_i, $\{i = 1, 2, 3 \ldots 13\}$). In addition, four loop detectors are located in each one of the off-ramps (OS_i, $\{i = 1, 2, 3, 4\}$), and another four loop detectors in each one of the on-ramps (IS_i, $\{i = 1, 2, 3, 4\}$). Data from 0:00 September 1st, 2013 until 23:55 September 30th, 2013 was collected for this study, obtaining 7938 samples in total. Each sample contains the following attributes:

- F_x, $\{x = 1, 2, 3 \ldots 13\}$: Flow reported by sensor in the road at point i, measured in number of vehicles.
- O_x, $\{x = 1, 2, 3 \ldots 13\}$: Occupancy reported by sensor in the road at point i. Percentage of time the sensor has detected a vehicle.
- S_x, $\{x = 1, 2, 3 \ldots 13\}$: Average speed of the vehicles passing through the point i, in km/h.
- iF_x, $\{x = 1, 2, 3, 4\}$: Flow reported by each sensor located at the on-ramps of the road or, in other words, number of vehicles that entered the highway.
- oF_x, $\{x = 1, 2, 3, 4\}$: Flow reported by each sensor located at the off-ramps of the road or, in other words, number of vehicles that leaved the highway.

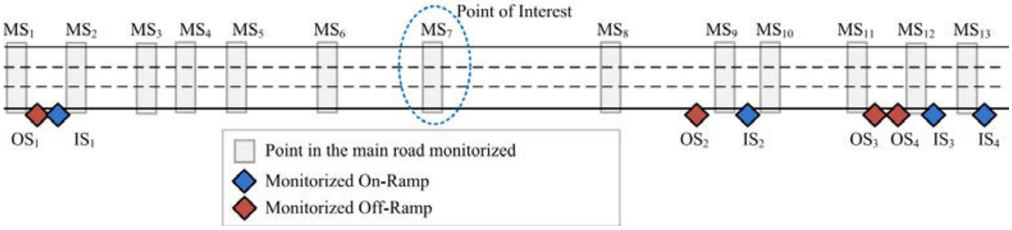

Fig. 1. Scheme of the scenario used in this study.

It is important to note that ramps do not report values of occupancy or speed, so only flow values are associated with them. With all of them, the total number of variables involved in the prediction is 47. Two different datasets have been built from this information; the first one is named *Complete,* and involves all the 47 attributes collected. The second one, named *Simplified*, includes 13 attributes; 11 of them are directly inherited from the Complete dataset, denoting the flow, occupancy and speed at the first and last sensors in the section of the road, and the same at the point of interest $\{F_1, O_1, S_1, F_7, O_7, S_7, F_{13}, O_{13}, S_{13}\}$. The input and output flows before the interest points $\{iF_1, oF_1\}$ are also included. Last two attributes denote the aggregation of input and output flows after the interest point, and they are calculated as presented at Equations 1 and 2.

$$iF_2' = iF_2 + iF_3 + iF_4 \quad (1)$$

$$oF_2' = oF_2 + oF_3 + oF_4 \quad (2)$$

A calculated value of congestion associated with the point of interest (S_7) is added as the last column of the datasets. This congestion level is calculated according with the extended HCM LOS F rating (Maryland, 2009), reported in Table 1. Finally, in order to generate datasets with different time horizons, the congestion value is translated one by one to the previous set of attributes, obtaining an increment in the prediction horizon of 5 minutes (but losing one sample) each time. This process is illustrated in Figure 2. This procedure was repeated until a prediction horizon of 60 minutes was reached. In summary, 24 datasets were finally obtained, whose names are Com_h for the complete ones (47 attributes), and Sim_h for the simplified ones (13 attributes). $h = \{10, 20 \ldots 60\}$ represents the prediction horizon.

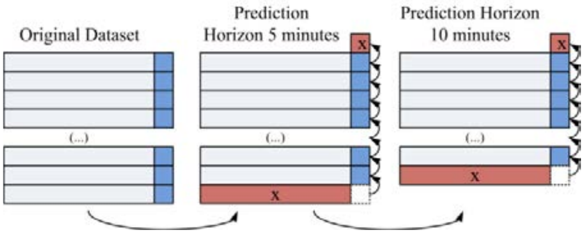

Fig. 2. Graphical representation of the process followed in order to obtain datasets with different prediction horizon.

Table 1. Levels of Congestion and their calculus.

Congestion	Flow/Speed (ve/km/ln)	Speed (km/h)
Severe	> 50	< 40
Moderate	[37–50]	[24–64]
Slight	[29–37]	[48–80]
Free	Other cases	

3. Techniques used

Experiments have been developed using the KEEL (http://www.keel.es) software (Alcalá-Fdez et al., 2009). Among the methods available in KEEL, a selection has been made among those that fall within the categories of Evolutionary Crisp Rule Learning (ECRL) and Evolutionary Fuzzy Rule Learning (EFRL). These categories include methods that make use of any Evolutionary computation (EC) mechanism for the generation and tuning of sets of rules to perform classification tasks. Methods included in the ECRL category used in this work are:

- C4.5 (Quinlan, 1993): This is a well-known algorithm used to generate decision trees from a set of training data in the same way as the ID3 algorithm.
- Bioinformatics-oriented Hierarchical Evolutionary Learning (BioHEL) (Bacardit et al., 2009): This system applies an almost standard generational Genetic Algorithm (GA). In this case, classification rules are the evolving individuals. The learning process creates a rule set by iteratively learning one rule at a time using a GA.
- GAssist_ADI (Bacardit and Garrell, 2003): The core of the system consists of a GA whose population is composed by a set of production rules. Individuals are evaluated according to the proportion of correct classified training examples.
- GAssist_Intervalar (Bacardit and Garrell, 2007): This method is an extension of the previous one, incorporating a rule deletion mechanism and a selection operator designed to guide the search to both accurate and short individuals.
- Hierarchical decision rules (Hider) (Aguilar-Ruiz et al., 2003): This method produces a hierarchical set of rules by means of a real coded GA. Two genes will define the lower and upper bounds of the rule attribute. One rule is extracted from the GA every iteration and all the examples covered by that rule are removed for the next iteration.
- Incremental Learning with Genetic Algorithms (ILGA) (Guan and Zhu, 2005): It follows the incremental learning approach supported by a GA with different initialization schemes. In addition, ILGA iteratively searches in one dimension each time, inheriting the information obtained step by step.
- Memetic Pittsburgh Learning Classifier System (MPLCS) (Bacardit and Krasnogor, 2009): This method hybridizes a GA with local search operators in the context of a Pittsburgh learning classifier system. Two different policies of integration are used, either applying the operators to the whole population or only to the best individual of the population.

- Ordered Incremental training with Genetic Algorithms (OIGA) (Zhu and Guan, 2004): This method works in two steps: first, it learns one-condition rules for each one of the attributes. Then it optimizes their values using a GA. Once all the attributes have been explored in a separated way, it joins the obtained rule sets ordered by fitness.
- Real Encoding Particle Swarm Optimization (REPSO) (Liu et al., 2004): This method uses a Particle Swarm Optimization for rule discovery following a Michigan approach, where an individual encodes a single rule.

It is important to note that C4.5 is not formally an ECRL algorithm, since it does not apply any kind of evolutionary process. Anyway, for this study, it has been considered because of being one of the most recognized techniques in the field of machine learning. Methods in the EFRL category used in the comparative study are:

- GFS_GCCL (Ishibuchi et al., 1999): In this method, each fuzzy rule is handled as an individual. The technique uses linguistic values with fixed membership functions as antecedent of the rules.
- GFS_SP (Sánchez et al., 2001): In this approach, a simulated annealing is used to learn a fuzzy classifier with tree structure that can use any combination of conjunction and disjunctions in the antecedent part of the rules.
- GFS_LogitBoost (Otero and Sánchez, 2006): This method uses the boosting paradigm to fuzzy rules extracted from data by means of a GA. Each time a new rule is added to the classifier, the examples in the training set are re-weighted. In this way, future rules will focus on the most difficult examples.
- Steady-state GA for Extracting fuzzy classification Rules from Data (SGERD) (Mansoori et al., 2008): This method uses a steady-state GA that generates a specified number of rules per class. In each generation, candidate rules are divided according to their consequent class, and they are ranked with respect to their fitness. This technique uses multiple fuzzy partitions simultaneously with different granularities and a *don't care* condition for fuzzy rule extraction.
- Structural Learning Algorithm on Vague Environment (SLAVE) (González and Pérez, 1999): This approach extracts a set of fuzzy rules from a set of examples through an iterative process in which a rule is selected each time. It uses a GA to select the rule which best represents the system. The rule obtained is incorporated into the final set of rules. In order to obtain new and different rules, the rule previously got is penalized, and the process is repeated.
- Chi_RW (Chi et al., 1996): This method generates a fuzzy rule for each one of the examples, using a predefined, normalized partition of the universe of discourse of each one of the variables. Once the initial complete rule base, weights of the individual rules are adjusted.
- Fuzzy Association Rule-based Classification model for High-Dimensional problems (FARCHD) (Alcalá-Fdez et al., 2011): This method mines fuzzy association rules limiting the order of the associations in order to obtain a reduced set of candidate rules with less attributes in the antecedent. Finally, a genetic rule selection and lateral tuning are applied to select a small set of fuzzy association rules with high classification accuracy.

It is important to note that both Chi_RW and FARCHD are not included in the EFRL category in the KEEL distribution. Chi_RW is one of the first and most recognized methods for the automatic learning of fuzzy systems, despite not having the EC component. In the case of FARCHD, it appears under the associative classification category, but it has been included in this list because it considers fuzzy association rules and uses an EC process in its work-flow. All the methods were run considering default configurations given by KEEL, which are the same suggested by authors in the publications in which those methods were presented.

4. Performance evaluation measures

In the four-class problem faced here, the confusion matrix (shown in Table 2) records the results of correctly and incorrectly recognized examples of each class after the execution of the method. Since a large number of methods use the accuracy rate (Eq. 3) as empirical measure for the quality of the models, a first comparison in this term will be provided. However, in the framework of imbalanced data-sets, as the one presented here, it does not distinguish between the numbers of correctly classified examples of different classes.

$$Acc = \frac{FF+LL+MM+SS}{F'+L'+M'+S'} \tag{3}$$

Table 2. Confusion matrix for the four-class problem used in this work.

Actual/Predicted	Free	Light	Medium	Severe	Total
Free	FF	FL	FM	FS	F’
Light	LF	LL	LM	LS	L’
Medium	MF	ML	MM	MS	M’
Severe	SF	SL	SM	SS	S’

To save this inconvenience, a generalized version of the averaged accuracy measure (Kubat et al., 1998) for more than two classes (Eq. 4) is used. This value measures the balanced performance of the model between the different classes of the problem, allowing to simultaneously maximizing the accuracy in each one of them (Gu et al., 2009).

$$Aacc = \frac{1}{4}\left(\frac{FF}{F'} + \frac{LL}{L'} + \frac{MM}{M'} + \frac{SS}{S'}\right) \tag{4}$$

In order to compare the complexity of the models, both number of rules generated and average length of rules were used for comparison purposes. Since most of the techniques presented here use a format of rules where the antecedent is composed by a set of and-linked atomic clauses (crisp or fuzzy, depending on the method), the length of the rule becomes equal to the number of clauses in its antecedent.

5. Experimentation and results

All the experiments conducted in this work have been performed on an Intel Core i5 2410 laptop, with 2.30 GHz and a RAM of 4 GB. In order to evaluate the performance of the models with independence of the particular instances used for training them, 10-fold cross validation partition was applied over each dataset. Along the present section, results obtained by all the techniques shown in Section 3 applied over the datasets described in Section 2 are presented. First, results in terms of accuracy and averaged accuracy per class are analyzed. After that, a comparison in terms of time to generate models and complexity is performed. Finally, a discussion among different results obtained by crisp and fuzzy techniques in overall is provided.

Figure 3 presents, in a graphical way, accuracy measures obtained by each one of the techniques when applied over the datasets. From Figure 3, it can be appreciated that lower results in accuracy (darker) are mainly obtained by Hider and GFS_GCCL, in addition to GFS_LogitBoost, in the case of complete datasets. Another conclusion that can be extracted from Figure 3 is the fact that all the methods obtain, in all the cases, accuracy values beyond 0.95. Additionally, no clear (with exception in some particular cases) differences can be appreciated. This can be translated in a percentage of matching of the level of congestion higher than 95% of the examples contained in the dataset. But, as commented in previous section, this assumption may result in mistaken conclusions.

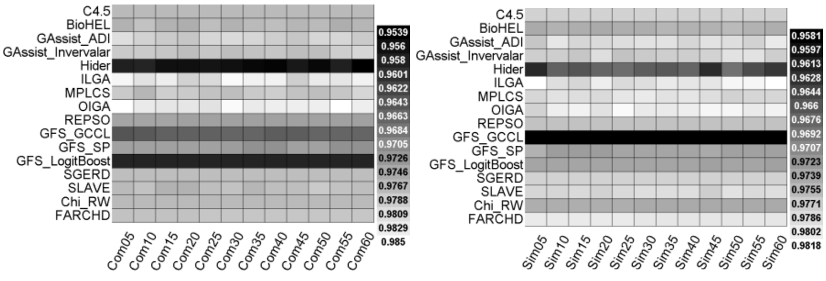

Fig. 3. Accuracy obtained for each technique in complete (left) and simplified (right) datasets.

In order to provide a more clear view of the differences on results obtained in this experimentation, Tables 3 and 4 present results in terms of averaged accuracy for complete and simplified datasets, respectively. The last column, named *R*, denotes the averaged ranking obtained by the technique in the total group of datasets. In all the cases, the three best values for each column are boldfaced. It can be seen that, for both types of datasets, C4.5 obtains, in most of the cases, the best value. After it, FARCHD gets, in almost all the cases, the second or third best value. Apart from them, second or third best values are usually shared among MPLCS, OIGA and ILGA, in the case of complete datasets. While for simplified datasets, MPLCS, ILGA and Gassist_Intervalar appear among the three best techniques. In addition, it is interesting to remark that, for all the cases, averaged accuracy is not deteriorated as the prediction horizon increases.

Table 3. Accuracy (Eq. 4) value obtained for each technique in normal datasets.

	Com$_{5}$	Com$_{10}$	Com$_{15}$	Com$_{20}$	Com$_{25}$	Com$_{30}$	Com$_{35}$	Com$_{40}$	Com$_{45}$	Com$_{50}$	Com$_{55}$	Com$_{60}$	R
C45	**0.604**	**0.604**	**0.604**	**0.604**	**0.604**	**0.604**	**0.604**	**0.604**	**0.604**	**0.604**	**0.604**	**0.604**	**1.2**
BioHEL	0.538	0.567	0.544	0.529	0.537	0.529	0.535	0.542	0.540	0.543	0.520	0.538	9.2
Gassist_ADI	**0.601**	**0.588**	0.567	0.554	0.582	0.590	**0.583**	0.569	0.560	0.585	0.569	0.555	4.5
Gassist_Invervalar	0.574	0.569	0.551	0.543	0.557	0.540	0.557	0.560	0.542	0.553	0.565	0.576	6.5
Hider	0.493	0.493	0.492	0.498	0.491	0.494	0.494	0.492	0.494	0.491	0.491	0.490	13.0
ILGA	0.564	0.553	0.578	0.518	0.519	**0.598**	0.573	0.554	0.500	0.550	0.558	0.573	7.3
MPLCS	0.579	0.549	**0.581**	**0.567**	**0.585**	0.561	0.581	**0.585**	**0.590**	0.570	0.5809	**0.604**	**4.1**
OIGA	**0.603**	0.552	0.578	0.563	0.535	**0.605**	0.562	0.5754	0.581	**0.586**	**0.607**	0.583	4.2
REPSO	0.527	0.527	0.527	0.512	0.512	0.512	0.512	0.512	0.512	0.512	0.512	0.512	11.8
GFS_GCCL	0.338	0.349	0.364	0.364	0.364	0.371	0.371	0.364	0.364	0.380	0.380	0.380	15.0
GFS_SP	0.479	0.479	0.479	0.479	0.479	0.477	0.451	0.451	0.483	0.483	0.451	0.451	14.0
GFS_LogitBoost	0.254	0.254	0.254	0.254	0.254	0.254	0.254	0.254	0.254	0.254	0.254	0.254	16.0
SGERD	0.531	0.531	0.531	0.531	0.531	0.531	0.531	0.531	0.531	0.531	0.531	0.531	10.3
SLAVE	0.533	0.547	0.525	0.527	0.545	0.554	0.530	0.542	0.545	0.564	0.552	0.549	8.8
Chi_RW	0.551	0.551	0.551	0.551	0.551	0.551	0.551	0.551	0.551	0.551	0.551	0.551	7.5
FARCHD	0.578	**0.586**	**0.586**	**0.586**	**0.593**	0.597	**0.593**	**0.593**	**0.593**	**0.593**	**0.590**	**0.590**	**2.7**

Table 4. Averaged accuracy (Eq. 4) value obtained for each technique in simplified datasets.

	Sim$_{5}$	Sim$_{10}$	Sim$_{15}$	Sim$_{20}$	Sim$_{25}$	Sim$_{30}$	Sim$_{35}$	Sim$_{40}$	Sim$_{45}$	Sim$_{50}$	Sim$_{55}$	Sim$_{60}$	R
C45	**0.605**	**0.605**	**0.605**	**0.605**	**0.605**	**0.605**	**0.605**	**0.605**	**0.605**	**0.605**	**0.605**	**0.605**	**1**
BioHEL	0.529	0.529	0.539	0.507	0.511	0.526	0.529	0.531	0.517	0.541	0.524	0.521	10.8
Gassist_ADI	0.535	0.572	0.548	0.550	0.561	0.573	0.558	0.557	0.565	0.571	0.539	0.579	6.0
Gassist_Invervalar	0.541	0.547	0.547	**0.569**	0.525	0.541	0.535	0.541	0.555	**0.581**	0.570	0.540	6.9
Hider	0.458	0.491	0.476	0.488	0.484	0.486	0.463	0.492	0.456	0.490	0.480	0.458	15.0
ILGA	**0.576**	0.533	0.528	0.556	0.516	0.535	0.544	0.545	0.55	0.562	0.566	**0.586**	7.3
MPLCS	0.577	**0.584**	**0.591**	0.566	**0.589**	**0.590**	**0.572**	**0.599**	**0.588**	0.566	**0.571**	0.580	**3.1**
OIGA	0.545	0.558	0.565	0.555	0.571	0.537	0.542	0.566	0.564	0.547	0.550	0.559	6.4
REPSO	0.530	0.530	0.530	0.530	0.530	0.530	0.530	0.530	0.530	0.530	0.530	0.530	10.0
GFS_GCCL	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	16.0
GFS_SP	0.509	0.492	0.492	0.492	0.491	0.491	0.503	0.503	0.498	0.491	0.491	0.497	13.5
GFS_LogitBoost	0.508	0.508	0.51	0.514	0.512	0.506	0.506	0.509	0.506	0.509	0.506	0.504	11.9
SGERD	0.548	0.548	0.548	0.548	0.548	0.548	0.548	0.548	0.548	0.548	0.548	0.548	7.3
SLAVE	0.557	0.564	0.566	0.559	0.566	0.566	0.571	0.565	0.551	0.576	0.568	0.568	5.0
Chi_RW	0.497	0.497	0.497	0.497	0.497	0.497	0.497	0.497	0.497	0.497	0.497	0.497	13.3
FARCHD	**0.591**	**0.591**	**0.591**	**0.591**	**0.591**	**0.584**	**0.584**	**0.584**	**0.584**	**0.584**	**0.584**	**0.584**	**2.4**

With the aim of providing a general view on complexities of the algorithms, as well as the ones for the returned models, Table 5 presents complexity results. In this table the computational time needed by all the methods to return the prediction model is shown, in columns denoted with *T*. With respect to the complexity of the models, the average number of rules (#R) and attributes (#A) are presented. All the measures are separated in values obtained for both complete and simplified version of the datasets. Regarding the complexity, it is important to note that some of the techniques use a fixed number of rules or antecedents. These methods are remarked with an asterisk in the table.

Table 5. Complexity measures of the methods and the models returned.

	T(Com_x)	T(Sim_x)	#R(Com_x)	#R(Sim_x)	#A(Com_x)	#A(Sim_x)
C45	**11.7**	**5**	45.93	39.15	7.85	7.26
BioHEL	48.7	27.1	13.61	14.56	4.01	3.92
Gassist_ADI	865.1	223.4	**5.36**	5.56	3.57	2.46
Gassist_Invervalar	398.4	81.7	**4.7**	**5.12**	46.67	13.21
Hider	331.8	28.1	54.36	**3.63**	43.2	8.47
ILGA	1787.3	364.9	30*	30*	45.48	12.03
MPLCS	4164.1	643.4	13.82	11.73	5.28	4.1
OIGA	2135.7	363.9	30*	30*	45.48	12.03
REPSO	31.5	15.4	7.57	7.42	3.68	3.03
GFS_GCCL	**8.4**	**4**	30.82	19.54	2.8	**1.06**
GFS_SP	1536.3	806.9	**4***	**4***	**2.2**	3.11
GFS_LogitBoost	814.1	212.2	25*	25*	10*	10*
SGERD	**9.9**	6.9	5.41	6.81	**1.93**	**1.94**
SLAVE	1601.7	728.1	31.77	26.68	7.54	4.89
Chi_RW	99.3	**4.3**	2936.82	318.32	46.44	13
FARCHD	626	49.7	25.25	14.34	**2.27**	**2.08**

Looking at Table 5, remarkable differences can be observed in the complete datasets in terms of execution times when comparing the three faster methods with the rest of them. Six methods are capable of returning a model in less than 120 seconds, while five of them do it in more than 1200 seconds. Regarding the simplified ones, nine of the methods finish in less than 120 seconds, and only three of them last more than 600 seconds. The behavior of Chi_RW is remarkable since, while in the simplified datasets is the second less time consuming technique, in the complete one it achieves the sixth position, multiplying the value more than 20 times. Observing the number of rules, Chi_RW obtains the highest number of rules in both complete and simplified datasets, with high difference from the second highest value, which is achieved by C4.5. Finally, regarding the number of antecedents, some methods use almost all the available attributes in their rules (Gassist_Intervalar, ILGA, OIGA and Chi_RW), while other ones use very simple rules, composed by less than 4 attributes, as FARCHD, SGERD, GFS_SP, GFS_GCCL, Gassist_ADI.

5.1. Discussion

In order to provide a final analysis of the study conducted in this work, Table 6 presents all the results grouped by type of technique, distinguishing between the selected crisp and fuzzy learning methods (ECRL and EFRL). Each cell of the table represents the averaged ranking obtained by the techniques of the groups, considering the four criteria managed along the article, distinguishing between the complete and simplified datasets. Next, it is proceeded to provide general conclusions about the results, having into account that these conclusions may not be extended to all the fuzzy or crisp techniques, but limited to the field of study and considering the parametrization used. In addition, it is important to note that these considerations are deduced from results obtained for all the techniques in the group. Without taking into account specific cases of behavior shown from particular techniques.

Table 6. Complexity measures of the methods and the models returned.

	Aacc(Com$_x$)	Aacc(Sim$_x$)	T(Com$_x$)	T(Sim$_x$)	#R(Com$_x$)	#R(Sim$_x$)	#A(Com$_x$)	#A(Sim$_x$)
ECRL	**6.86**	**7.38**	9.22	8.88	**8**	**7.88**	10	9.66
EFRL	10.61	9.91	**7.75**	**8**	9	9.14	**6.42**	**6.85**

As can be seen in Table 6, crisp methods improve results obtained by fuzzy ones in terms of averaged accuracy and number of rules, while the fuzzy ones get better values in terms of time needed for constructing the models and in the length of the rules included in them. It is noteworthy that when a group of techniques gets better results than the other, it does for both complete and simplified datasets.

Bigger differences can be found between values for accuracy and number of antecedents. This is an expected result when dealing with these two paradigms since, usually, crisp methods do not use to consider the readability or interpretability of the model provided, in case it affects to the performance, while fuzzy methods, because of their linguistic nature, are designed to return easily interpretable models.

6. Conclusions and future works

This work presents an empirical study on the application of machine learning methods to the task of predicting a certain level of congestion in a road. The study is oriented to find relevant conclusions and differences among techniques that generate a model consisting in a set of rules, considering both crisp and fuzzy variants.

With the aim of carrying out the study, 16 techniques, nine of them included in the category of crisp and seven in the category of fuzzy, are applied over 24 datasets. Half of the datasets make use of all the available information in the studied road segment, while the other half only uses a reduced number of variables available.

Data used in this experimentation is highly imbalanced, which represents a big challenge for techniques which are mainly designed for dealing with well-distributed data. However, methods used show reasonable good performances, returning models with a large variety of complexities, from those which infer a high number of rules that make use of almost all the attributes, to those which infer a low number of short and interpretable rules.

Next research in the direction of using well-known machine learning algorithms to prediction of the future state of the road will be oriented to solve deficiencies presented in this work. In particular, the main research line will be oriented in adapting those techniques to highly imbalanced domains, or to hybridize those techniques with the ones coming from specialized literature in the field. Other point that will be present in future research will be to extract those aspects of the studied methods that are desirable to get applied in the domain of processing information coming from traffic scenarios, such as the low computational times, even for large datasets, and the equilibrium between the accuracy shown by systems and the size of the generated models.

Acknowledgements

This work has been partially funded by the TIMON Project (Enhanced real time services for an optimized multimodal mobility relying on cooperative networks and open data). This project has received funding from the European Union’s Horizon 2020 research and innovation program under grant agreement No. 636220.

References

Aguilar-Ruiz, J., Riquelme, J., and Toro, M., 2003. Evolutionary learning of hierarchical decision rules. IEEE Transactions on Systems and Man and and Cybernetics – Part B: Cybernetics, 33(2):324–331.

Alcalá-Fdez, J., Alcalá, R., and Herrera, F., 2011. A fuzzy association rule-based classification model for high-dimensional problems with genetic rule selection and lateral tuning. IEEE Transactions on Fuzzy Systems, 19(5):857–872.

Alcalá-Fdez, J., Sánchez, L., García, S., del Jesus, M., Ventura, S., Garrell, J., Otero, J., Romero, C., Bacardit, J., Rivas, V., Fernández, J., and Herrera, F. ,2009. Keel: A software tool to assess evolutionary algorithms for data mining problems. Soft Computing, 13(3):307–318.

Bacardit, J., Burke, E., and Krasnogor, N., 2009. Improving the scalability of rule-based evolutionary learning. Memetic computing, 1(1):55–67.

Bacardit, J. and Garrell, J., 2003. Evolving multiple discretizations with adaptive intervals for a pittsburgh rule-based learning classifier system. In Genetic and Evolutionary Computation Conference, volume 2724 of Lecture Notes on Computer Science, pages 1818–1831.

Bacardit, J. and Garrell, J.M., 2007. Bloat control and generalization pressure using the minimum description length principle for a pittsburgh approach learning classifier system. In Learning Classifier Systems, pages 59–79. Springer.

Bacardit, J. and Krasnogor, N., 2009. Performance and efficiency of memetic pittsburgh learning classifier systems. Evolutionary computation, 17(3):307–342.

Box, G., Jenkins, G., and Reinsel, G., 2008. Time Series Analysis: Forecasting and Control. Wiley.

Chen, B. and Cheng, H., 2010. A review of the applications of agent technology in traffic and transportation systems. IEEE Transactions on Intelligent Transportation Systems, 11(2):485–497.

Chi, Z., Yan, H., and Pham, T., 1996. Fuzzy Algorithms: With Applications To Image Processing and Pattern Recognition. World Scientific.

Cobo, M., Chiclana, F., Collop, A., de Ona, J., and Herrera-Viedma, E., 2014. A bibliometric analysis of the intelligent transportation systems research based on science mapping. IEEE Transactions on Intelligent Transportation Systems, 15(2):901–908.

Commission, E., 2011. White paper, roadmap to a single european transport area, towards a competitive and resource efficient transport system.

DeJong, K.A. and Spears, W.M., 1990. Learning concept classification rules using genetic algorithms. Technical report, DTIC Document.

Ganganwar, V., 2012. An Overview of Classification Algorithms for Imbalanced Datasets. International Journal of Emerging Technology and Advanced Engineering, 2: 42–47.

Golinska, P. and Hajdul, M., 2012. European union policy for sustainable transport system: Challenges and limitations. In Golinska, P. and Hajdul, M., editors, Sustainable Transport, EcoProduction. Environmental Issues in Logistics and Manufacturing, pages 3–19. Springer Berlin Heidelberg.

González, A. and Pérez, R., 1999. Slave: A genetic learning system based on an iterative approach. IEEE Transactions on Fuzzy Systems, 7(2):176–191.

Gu, Q., Zhu, L., and Cai, Z., 2009. Evaluation measures of the classification performance of imbalanced data sets. In Cai, Z., Li, Z., Kang, Z., and Liu, Y., editors, Computational Intelligence and Intelligent Systems, volume 51 of Communications in Computer and Information Science, pages 461–471. Springer Berlin Heidelberg.

Guan, S. and Zhu, F., 2005. An incremental approach to genetic– algorithms–based classification. IEEE Transactions on Systems and Man and and Cybernetics and Part B, 35(2):227–239.

Hearst, M.A., Dumais, S.T., Osman, E., Platt, J., and Scholkopf, B., 1998. Support vector machines. IEEE Intelligent Systems and their Applications, 13(4):18–28.

Hong, W.-C., 2011. Traffic flow forecasting by seasonal svr with chaotic simulated annealing algorithm. Neurocomputing, 74(12- 13): 2096–2107.

Ishibuchi, H., Nakashima, T., and Murata, T., 1999. Performance evaluation of fuzzy classifier systems for multidimensional pattern classification problems. IEEE Transactions on Systems and Man and Cybernetics and Part B: Cybernetics, 29(5):601–618.

Kubat, M., Holte, R., and Matwin, S., 1998. Machine learning for the detection of oil spills in satellite radar images. Machine Learning, 30(2-3):195–215.

Liu, Y., Qin, Z., Shi, Z., and Chen, J., 2004. Rule discovery with particle swarm optimization. In Content Computing, pages 291–296. Springer.

Mansoori, E., Zolghadri, M., and Katebi, S., 2008. Sgerd: A steady-state genetic algorithm for extracting fuzzy classification rules from data. IEEE Transactions on Fuzzy Systems, 16(4):1061–1071.

Okutani, I. and Stephanedes, Y., 1984. Dynamic prediction of traffic volume through kalman filtering theory. Transportation Research Part B, 18(1):1–11.

Otero, J. and Sánchez, L., 2006. Induction of descriptive fuzzy classifiers with the logitboost algorithm. Soft Computing, 10(9):825–835.

Quinlan, J., 1993. C4.5: Programs for Machine Learning. Morgan Kauffman.

Sánchez, L., Couso, I., and Corrales, J., 2001. Combining gp operators with sa search to evolve fuzzy rule based classifiers. Information Sciences, 136(1-4):175–192.

Schalkoff, R.J., 1997. Artificial neural networks. McGraw-Hill New York.

State of Maryland., 2009. Major highway performance ratings and bottleneck inventory – Skycomp, Inc. in Association with Whitney, Bailey, Cox and Magnani, LLC.

Stathopoulos, A. and Karlaftis, M., 2003. A multivariate state space approach for urban traffic flow modeling and prediction. Transportation Research Part C: Emerging Technologies, 11(2):121–135.

Steenbruggen, J., Krieckaert, M., Rietveld, P., Scholten, H., and van der Vlist, M., 2013. Effectiveness of net-centric support tools for traffic incident management. In Zlatanova, S., Peters, R., Dilo, A., and Scholten, H., editors, Intelligent Systems for Crisis Management, Lecture Notes in Geoinformation and Cartography, pages 217–250. Springer Berlin Heidelberg.

Vlahogianni, E., Karlaftis, M., and Golias, J., 2005. Optimized and meta-optimized neural networks for short-term traffic flow prediction: A genetic approach. Transportation Research Part C: Emerging Technologies, 13(3):211–234.

Zhang, G., Eddy Patuwo, B., and Y. Hu, M., 1998. Forecasting with artificial neural networks: The state of the art. International Journal of Forecasting, 14(1):35–62.

Zhu, F. and Guan, S., 2004. Ordered incremental training with genetic algorithms. International Journal of Intelligent Systems, 19(12): 1239–1256.