Decentralized Intelligent Transport System with distributed intelligence based on classification techniques

E. Osaba*, E. Onieva, A. Moreno, P. Lopez-Garcia, A. Perallos, P. G. Bringas

Deusto Institute of Technology (DeustoTech), University of Deusto, Av. Universidades 24, Bilbao 48007, Spain

*e.osaba@deusto.es

Abstract: This paper is focused on a decentralized ITS with distributed intelligence based on classification techniques. The rationale behind this architecture is to offer a fully distributed, flexible and scalable system. The architecture encompasses the entire process of capture and management of available road data, enabling the generation of services to promote transportation efficiency. Besides that, thanks to the embedded classification techniques, the system is capable of predicting and reacting to certain events, facing them in an appropriate way. The aim of this work is to demonstrate how the system works in two different real-world use cases. To achieve this objective, how the architecture acts to deal with some incidences is proven. Additionally, both use cases serve to show the effective communication between the different components of the system. Besides this, this work demonstrates the fundamental role played by the artificial intelligence techniques working in the system. The well-known C4.5 algorithm has been used for the accurate prediction of traffic congestion and pollution level. We explain in this work the reasons for using this classification technique, and the previous experiments performed.

1. Introduction

The architecture of classical Intelligent Transportation Systems (ITS) [1] is purely hierarchical, with sensed data flowing from the leaves (i.e. road-side or vehicle-installed sensors) to the root (i.e. the traffic management centre) [2]. Usually, this kind of approach presents some disadvantages, showing a lack of flexibility and scalability in supporting an incremental growth of ITS elements [3]. This traditional approach also exhibits latency and availability issues because all sensor data has to be communicated back and forth to the central management centre, thereby turning it into a single point of failure. For these reasons research activities in ITS have changed the vision behind the definition of new architectures [4], switching from the hierarchical and vertical approach to a new vision, which is more horizontal and distributed [5].

In Cooperative ITS (C-ITS), vehicles communicate with each other and with the road infrastructure without the involvement of a central server, thus removing this bottleneck while maintaining the reliability of information about the vehicles, their location and the road environment.

Research projects such as CVIS [6] or Compass4D [7] have adopted and developed this strategy to achieve cooperativeness. In these architectures, all the agents involved (vehicles, roadside infrastructures, central systems, personal devices, etc.) are seen as nodes belonging to a common

network. Projects like Coopers, Safespot [8] and COMeSafety2 [9] use this approach to increase road safety through direct communication between vehicles (C2C).

The DRIVE C2X European project [10] is another example of recent implementations of C-ITS architectures, establishing a common reference system for C2X communications and performing successful large-scale field tests. These projects have developed and demonstrated both the supporting technology and numerous applications for cooperative infrastructures involving two-way communication of data between vehicles and road networks.

This work is focused on a fully distributed architecture to enable cooperative sensing and management in ITS environments. The system encompasses the entire process of capture and management of available road data, enabling the generation of services to promote transportation efficiency. In addition, thanks to the embedded artificial intelligence, the system is able to predict and react to certain incidents and events, giving an appropriate solution in a reasonable time. Specifically, in this work, that artificial intelligence will be composed of classification techniques. The feasibility of this architecture has been proven within the participation in the ICSI European project[1] developing a reference end-to-end implementation targeted to both urban and highway scenarios. This participation is described in this paper. The main idea behind the project relies on a local distributed intelligence, operating on a limited geographical scale, where data is timely distributed and processed without the need to contact a central subsystem.

In this document, a description of the overall system architecture is provided. Furthermore, the way how the intelligence and the sensed-data are provided is explained. The main contribution of this work is the practical application of the recently proposed system to two different real world scenarios. Both scenarios are related with the prediction of traffic data, the first one with traffic congestion in a highway and the second one with the pollution level prediction in a city. Additionally, for the prediction the well-known C4.5 classification has been used [11]. Besides that, the performed demonstration environment is also detailed in this work. This environment counts with a web application which is capable of displaying the complete operation of the system in a real scenario simulated in a laboratory. Thanks to this platform, it can be seen how the architecture predict and face certain events and incidents.

The remainder of this paper is structured as follows. In the following section (Section 2), a brief description of the architecture is performed. In Section 3 two of the most important components of the architecture are described, which are responsible of distributing all the information through the architecture. In Section 4 the proposed real scenarios are deeply described, followed by the used technique for the prediction (Section 5). Additionally, the experimentation carried out in this scenarios using real data is described in Section 6. After that, the developed demonstration environment is presented in Section 7. Finally, conclusions and future work are explained in Section 8.

2. Global system architecture

As has been mentioned in the introduction, a fully distributed architecture is used in this work, which provides an added value to improve transportation efficiency. This architecture has been recently presented by Moreno et al. in [12], and it is designed with the aim of enabling cooperative sensing and management in ITS environments. In this cooperative architecture, the intelligence is distributed over some elements of the infrastructure, which host a software platform for running ITS applications. Communication with the remote centres happens only for the transmission of

[1] ICSI: Intelligent cooperative sensing for improved traffic efficiency, http://www.ict-icsi.eu/

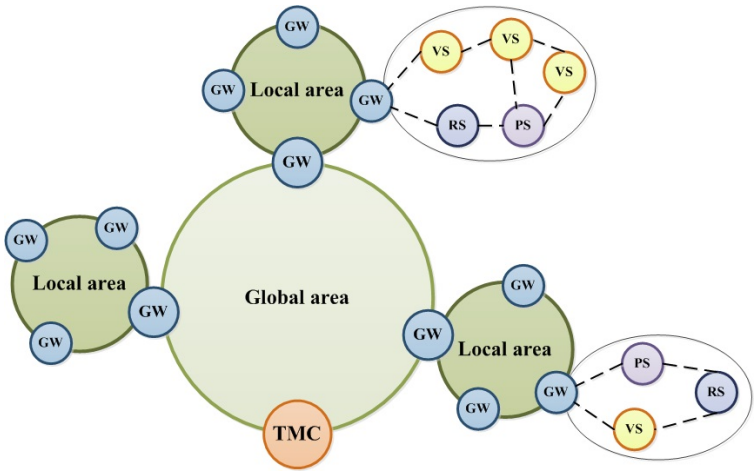

Fig. 1. *Global system architecture. TMC: Traffic Management Centre, VS: Vehicular Subsystems, PS: Personal Subsystems, RS: Road-side Subsystems.*

aggregated data for long-term operations (e.g., data mining, software upgrades, and logging). Moreover, real-time data is processed and stored locally in the system infrastructure, nearby the source of the events.

The architecture relies on a local distributed storage and artificial intelligence methods (in this work, classification techniques), which operates on a limited geographical scale. Data are distributed and processed in real-time without contacting the control centres. Additionally, sensed data are treated in a cooperative way performing content aggregation and integration since the earliest stages. Two concepts are defined to achieve distributed intelligence and cooperative sensing: the gateways (GW) and the Local/Global areas.

On the one hand, GWs are physical entities that implement the reference architecture, the Data Distribution Platform (DDP) and the Collaborative Learning Unit (CLU). GWs are able to join Local/Global Areas, and they are connected to different subsystems. Additionally, they analyze the information gathered and determine the best traffic strategies for dealing with roadway incidents, enhancing the scalability of the system and overcoming the weakness of centralized approaches.

On the other hand, a Local/Global area is composed by a set of GWs (at least one), communications among them (when multiple gateways are present), and a criteria to define the area perimeter (e.g. based on the density of population, traffic, ICT elements, etc.).

Once described these two components, it is interesting to highlight the capacity of the system to properly scale. Thanks to the decentralized nature of the architecture, the addition of new GWs, sensors of Local areas only implies their addition to the Traffic Management Centre.

Finally, in order to facilitate understanding of the system, a logical view of the architecture is shown in Figure 1, which depicts the previously introduced concepts. Furthermore, Figure 2 shows how the different components of the system are connected. For the correct comprehension of this figure, it should be introduce the concept of connectors, which are used to extend the GW interoperability via interface integration with external subsystems and technologies. Additionally, ITS applications are high level services or end-user applications which implement travel and models to respond to roadway incidents. Lastly, external components are software components running in the same OS, and which are required by other elements to work properly.

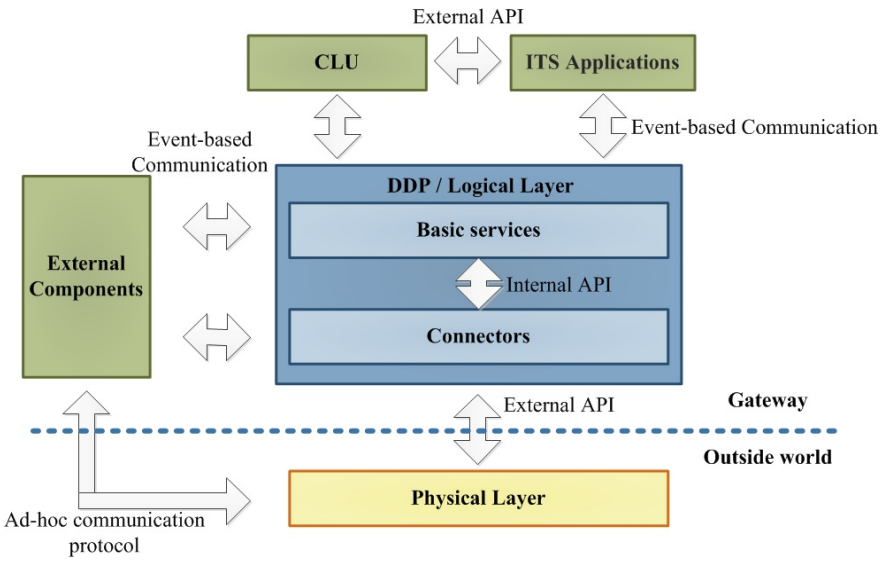

Fig. 2. *Relation of the different components of the system*

3. Distributed intelligence through CLUs

The two key elements of the architecture are the DDP and the CLUs. On the one hand, the DDP is responsible for providing the mechanisms for the communication between the different layers using a publisher/subscriber events-based architecture. In this way, other software components would only have to be concerned to inform the DDP about the events in which they are interested, according to their services. Therefore, the DDP considers these subscriptions to decide who has to redirect messages to the interested components, inside the Local Area or in other Local Areas.

Furthermore, CLUs are able to learn from each situation in order to provide different actions plans in real time depending on the events and the state of the infra-structure reported by the deployed Wireless Sensor Networks and Vehicular Networks. Since CLUs act in real time, they are constantly waiting to receive data from the available sensors and networks, and they decide every time they receive new data.

CLUs are responsible for responding to incidents that arise on the road, such as accidents, delays, CO2 levels, etc. As the situations that may occur are rich and varied, services implemented by the CLU will have different behaviours. These behaviours are dynamically adjusted based on the learning capabilities provided by artificial intelligence techniques integrated in the CLUs, such as metaheuristics, classification or machine learning methods.

One of the main challenges of a CLU is to develop stable and distributed algorithms based on probabilistic reasoning and not requiring very high computational resources. The CLU has to implement some techniques to solve the different problems that could appear. These problems and techniques have been categorized in Table 1 according with the role of the CLU and the kind of problem. It is noteworthy that these problems are the ones addressed in ICSI project.

Despite the system contemplates contingency plans to solve other problems (Section 4 and Section 7), this work is focused in problems related with prediction (Section 5 and Section 6). This category contains all problems and techniques that estimate the probability of some event to appear, like congestion and high pollution levels. These prediction problems use classification techniques to solve them. As shown in Table 1, the prediction problems could be faced with the

Table 1 Different problematic being faced by CLU

Category	Problems Faced	Techniques
Routing	Alternative routes Alternative transport Route guidance and emergency support	Genetic Algorithms Fuzzy Decision Support Systems Probabilistic Models
Regulation	Ramp-metering algorithms Access to pollution-monitored area	Fuzzy Control Hybrid Models Particle Swarm Optimization
Prediction	Level of congestion Travel time for vehicles Pollution	Probabilistic Models Bio-inspired optimization Time-series predictions
Monitoring	State of traffic Free parking slots Incidents	Particle filters I2V Communications

use of probabilistic models, bio-inspired optimization and time-series prediction. In this paper, classification techniques have been used to address two different scenarios related with congestion and pollution prediction.

4. Description of the test scenarios

This section presents the process followed with the aim of testing the CLU functionality using real data coming from predefined ICSI scenarios. Two scenarios have been selected, corresponding to the two field trials scheduled to take place in Lisbon (Portugal) and Pisa (Italy).

The first scenario (Section 4.1) corresponds to a highway mobility environment. Real data about the vehicles traffic flow in the A5 highway, connecting Lisbon to Cascais, has been incorporated to the experimentation process in order to test the CLU in a context as close to the real one as possible. Analogously, real data about the pollution levels in a restricted city area has been incorporated to the second scenario (Section 4.2), simulating an urban mobility environment.

4.1. Highway Scenario in Lisbon, Portugal

The A5 highway of Lisbon is a 25 km (16 miles) long motorway which connects the capital city of Portugal to Cascais. The motorway is also known as Estoril Coast Motorway. The first section of this infrastructure was opened in 1944 (Lisbon - Estadio Nacional), becoming the first motorway in Portugal and one of the firsts in the world. Nowadays, it is the most travelled motorway of the country and one of the most congestion prone ones.

In this context, the proposed test scenario includes the implementation of these use cases:

- Alternative paths signaling / route guidance
- Monitoring of anomaly in traffic flows (congestion)
- Traffic jam / accident warning
- Road works warning

The architecture provides an in-route traveler information about traffic and road conditions according to both static and dynamic rules based on real-time traffic. Warning notifications are

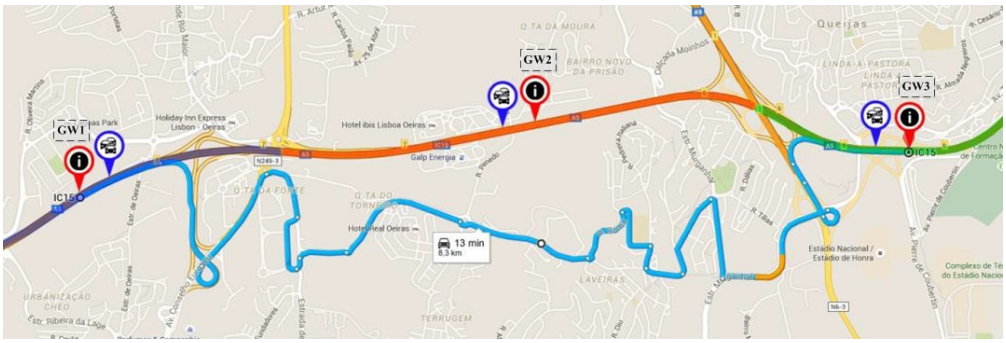

Fig. 3. *Highway scenario map, A5 Portugal, with an alternative route in case of abnormal traffic.*

Gateway	Data Model	Data Source
GW1	Abnormal Traffic	GW2
GW1	Congestion level	GW1, GW2
GW2	Abnormal Traffic	GW3
GW2	Congestion level	GW2, GW3
GW3	Congestion level	GW3
GW1,2,3	Vehicle Counter	Traffic Sensors (GW1, 2, 3)

Table 2 Traffic Model set of rules for the urban scenario.

communicated in case of congestion due to the high flow of vehicles or accidents. The system warns drivers coming to a traffic jam area and it is able to suggest alternative routes which may enable the driver to go around congested roads.

The information on recommended routes may be provided via the highway traffic management centre in order to maintain the overall traffic management in the area, or by the use of floating car data collected by the system. Figure 3 shows a map of the highway scenario with the location, the attached sensors and the area of influence of the deployed GWs on the road.

In this scenario, the tree different gateways have the same configuration. Thus, each GW is configured to get *Abnormal Traffic* (e.g. accident or roads work warning) events from the next GW on the road. Additionally, each GW also gets *Vehicle Counter* events. These events come from the own GWs attached sensors and they are delivered to the CLU in order to detect congestion using the implemented AI. If congestion is detected a *Congestion Level* event is launched. Each GW is listening for *Congestion Level* events from itself and from the next GW on the road in order to act with foresight and warn the drivers about expected traffic jumps. The characteristics of each GW is represented in Table 2

For the demonstration, two sources of data have been used for this scenario. The first one is data about incidents on the road, which is artificially generated for this demonstration. The second source is the data gathered by sensors counting the number of vehicles. For this source, real data coming from BRISA[2] is used.

In this way, and based on the available data, the implemented CLU is able to alert the drivers and/or the emergency services in case of incidents on the road, and predict the evolution of congestion based on a big dataset of historical data.

[2]BRISA is a Portugal-based international transportation company. Its largest business area is highway management.

4.2. Urban Scenario in Pisa, Italy

Pisa is a city in Tuscany, Central Italy, on the right bank of the mouth of the River Arno on the Tyrrhenian Sea. It is the capital city of the Province of Pisa. Some city centre areas have a controlled access through restricted traffic zones (RTZs) and Low Emission Zones (LEZ). These RTZs are closed to non-residential vehicle traffic. Only city buses, taxis, residents with a valid permit and other authorized vehicles (i.e. delivery vehicles, couriers, etc.), can drive there. The boundaries of the zone are well marked. At the access points, special displays indicate if access is authorized or not at that time. These access points are controlled by cameras and sensors.

Furthermore, LEZs are a way to reduce the pressure of non-residential traffic in highly touristic destinations. The objective of LEZs is to control the pollution level in high congested and populated zones.

In this context, the proposed test scenario includes the implementation of these use cases:

- Alternative transport services
- Monitoring and reduction of air pollution
- Alternative paths signaling / route guidance
- Cooperative parking slots monitoring

The system constantly monitors the pollution of the roads in the RTZ and LEZs of Pisa. When it predicts that the level of pollution will exceed the threshold, the system suggests leaving the car in the parking area and continuing the trip using alternative transport services. In addition, a portion of the parking lot dedicated to private vehicles is monitored. The plan for the equipment installation includes the deployment of 12 different sensors on 6 poles for monitoring up to 71 slots. Furthermore, flow monitoring is performed at the entrance of the city, measuring flows in a location at about 1 km far from the parking lot. Figure 4 illustrates the performed installation process. It should be born in mind that a photovoltaic panel has been installed in the top of each pole. These panels provide the sensors with the energy they need to work properly.

(a) Example of installation of two sensors on a single pole

(b) Mounting the sensor and the photovoltaic panel on the top of the pole

Fig. 4. *Sensor installation for the urban tests*

Besides that, figure 5 shows a map of the urban scenario with the location of the deployed GWs.

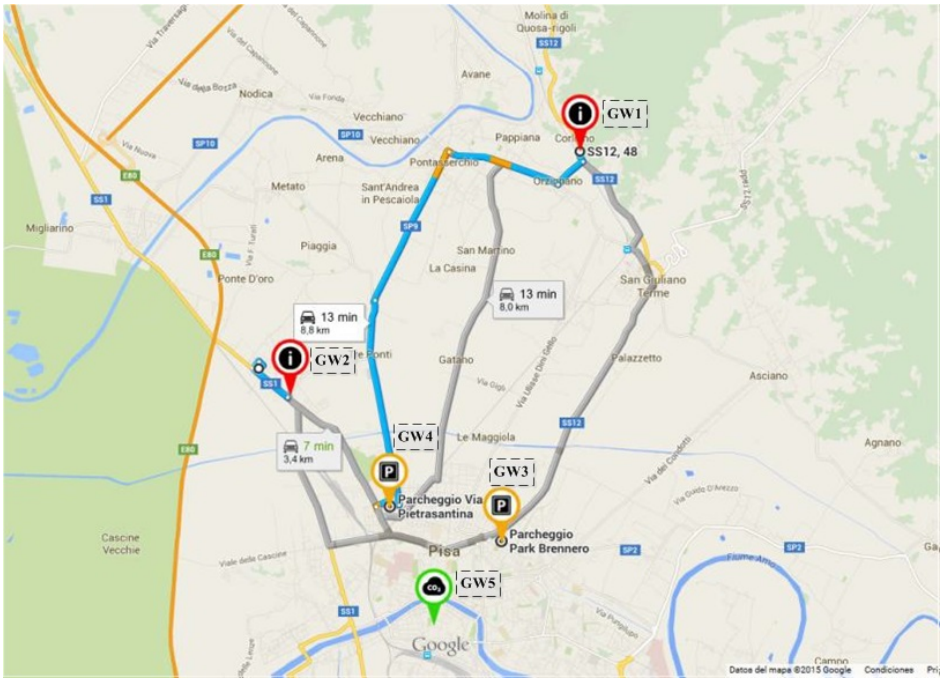

Fig. 5. *Urban scenario, location of the GWs.*

Gateway	Data Model	Data Source
GW1,2	Pollution Level	GW5
GW1,2	Parking Occupancy	GW3
GW1,2	Parking Occupancy	GW4
GW3	Parking Monitoring	Parking Sensors (GW3)
GW4	Parking Monitoring	Parking Sensors (GW4)
GW5	Pollution	Pollution Sensors (GW5)

Table 3 Traffic Model set of rules for the urban scenario.

Five different GW are available in this scenario. On the one hand, GW3 and GW4 (receiving the status of parking slots) and GW5 (receiving data about actual pollution) are configured to listen for events coming from their own attached sensors. All these GWs calculate the *Parking Occupancy* and predict *Pollution Levels*, and they publish the information to the DDP. On the other hand, GW1 and GW2, located at the entrance of the city, are listening for these events in order to act consequently, redirecting the drivers if necessary. Table 3 shows the characteristics of each GW of this scenario.

For this urban scenario two sources of data have been used. The first one is the number of free parking slots. In this case, and due to the lack of real data when implementing the scenario, simulated data are generated. The second source is the data gathered by sensors of pollution in the LEZ situated in the city centre. Real data coming from INTECS[3] about pollution in the city of Pisa is used here. As has been noted, and based on this data, the implemented CLU is able to know the percentage of parking occupancy and to predict the evolution of pollution levels.

[3]INTECS is an Italian ICT Company focused in design and production of SW/HW electronic components, Software engineering and Quality

5. Classification techniques

As has been said before, in this paper a classification technique has been used to perform two different kinds of predictions: traffic congestion and pollution level. As can be logical, these predictions have been conducted with the data obtained in the scenarios described in the previous section.

The proper prediction of information related to the traffic (traffic congestion, pollution level, incidents...) is an area which attracts considerable interest from researchers in the field of ITS. This kind of predictions can lead to traffic managers and drivers to act in consequence, reducing so 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, search for non-linear patterns in the input data.

Many different approaches can be used to perform an adequate prediction of traffic information. As has been mentioned before, this study is focused in classification techniques. Classification is a data mining trend which consists on mapping data into predefined groups or classes [13]. A classification method is a supervised learning method that requires labelled training data to generate rules for classifying test data into predetermined groups or classes [14]. The goal of these methods is to build a concise model of the distribution of class labels in terms of predictor features. The resulting classifier is then used to assign class labels to the testing instances where the values of the predictor features are known, but the value of the class label is unknown [15]. Some of the most well-known classification technique are the artificial neural networks [16], support vector machines [17], and decision trees [18]. These approaches have been used in many fields along the history, such as geology [19], medical sciences [20] or computer sciences [21].

In the present work, predictions are performed with the C4.5 classification technique. The C4.5 is a well-known algorithm used to generate decision trees from a set of training data [11]. In this sense, the decision tree is constructed top-down. In each step a test for the actual node is chosen (starting with the root node), which best separates the given examples by classes. The objective of this technique is to determine a decision tree that on the basis of answers to questions about the input attributes predicts correctly the value of the target attribute. It is noteworthy that the C4.5 is an extension of the ID3 algorithm [22], and that it is used to overcome its disadvantages. The improvements that the C4.5 offers comparing with the ID3 are the following ones:

- C4.5 accounts for unavailable or missing values in data.
- C4.5 handles continuous attribute value ranges.
- C4.5 chooses an appropriate attribute selection measure (maximizing gain)
- C4.5 prunes the result decision trees

Arguably, C4.5 is among the most popular of inductive inference algorithms, and it has been successfully applied to a broad range of tasks, from learning to diagnose medical cases [23, 24] to school performance prediction [25].

It should be highlighted that the choice of the C4.5 has not been arbitrarily done. For this choice a preliminary experimentation has been performed with different classification techniques, using some of the datasets that are explained in the following section [26]. In order not to increase the extension of the paper too much, a small portion of this study is depicted, showing the performance of six different classification techniques among 10 different datasets. In addition to the above mentioned C4.5, five different additional techniques have been used for the experimentation.

Table 4 Small portion of the preliminar tests. Results shown in terms of % matching over the test datasets.

Dataset	C4.5	C-SVM	PUBLIC	TARGET	SGERD	DT-GA-C
CL_400	**98.89**	91.71	95.58	82.87	88.39	96.13
CL_600	**92.26**	83.97	86.74	74.58	79.55	87.84
CL_3600	**93.92**	90.05	92.81	79.00	80.66	93.37
CL_7100	**96.13**	93.37	90.60	82.87	79.00	95.02
CL_20900	91.16	81.21	**92.26**	80.66	79.00	90.05
LC_400	**96.13**	92.26	95.02	79.00	88.39	94.47
LC_600	86.74	74.03	**89.50**	70.71	76.79	85.08
LC_1505	**94.47**	93.37	**94.47**	76.24	87.84	**94.47**
LC_1980	**93.90**	93.07	92.79	91.68	75.90	**93.90**
LC_3600	**97.23**	90.05	94.47	82.32	87.84	96.68

These techniques are a C-Support Vector Machine (C-SVM) [27], PUBLIC decision tree, which integrated building and pruning [28], TARGET decision tree [29], a Fuzzy Rule Based Steady-State Genetic Algorithm (SGERD) [30], and a Hybrid Decision Tree-Genetic Algorithm (DT-GA-C) [31].

It is noteworthy that KEEL[4] framework has been used for these tests, using the default parametrization for each technique. In Table 4 results obtained by all the techniques for the mentioned 10 datasets are shown in terms of percentage matching over the test datasets. It should be reminded that the datasets used in this preliminary study are drawn from the complete set described in the following section (Section 6.1).

Looking at the results displayed in Table 4 can be seen how C4.5 clearly outperforms the other alternatives. Anyway, two different statistical tests have been conducted with the results obtained in order to obtain rigorous and fair conclusions. The guidelines given by Derrac et al. in [32] have been followed to perform this statistical analysis. First of all, the Friedman's non-parametric test for multiple comparisons has been used to check if there are any significant differences among all the techniques. As can be seen Table 5, the resulting Friedman statistic has been 43.028571. Taking into account that the confidence interval has been stated at the 99% confidence level, the critical point in a χ^2 distribution with 5 degrees of freedom is 15.086. Since 43.028571>15.086, it can be concluded that there are significant differences among the results reported by the five compared algorithms, being C4.5 the one with the lowest rank. Finally, regarding this Friedman's test, the computed p-value has been 0.0.

To evaluate the statistical significance of the better performance of the C4.5, the Holm's post-hoc test has been conducted using C4.5 as control algorithm. The unadjusted and adjusted p-values obtained through the application of Holm's post-hoc procedure can be seen in Table 6. Analyzing this data it can be concluded that C4.5 is significantly better than TARGET, C-SVM and SGERD at a 95% confidence level, and better than PUBLIC and DT-GA-C.

Table 5 Average rankings returned by the Friedman's non-parametric test

Algorithm	Ranking
C4.5	1.3
C-SVM	3.9
PUBLIC	2.7
TARGET	5.7
SGERD	5.2
DT-GA-C	2.2

[4] http://www.keel.es/

Table 6 Unadjusted and adjusted p-values obtained through the application of Holm's post-hoc procedure using C4.5 as control algorithm.

Algorithm	Unadjusted p	Adjusted p
TARGET	0	0.000001
SGERD	0.000003	0.000013
C-SVM	0.001886	0.005659
PUBLIC	0.094264	0.188529
DT-GA-C	0.282059	0.282059

Table 7 Existing columns in received data.

Attribute	Meaning
NAME	Name of the station
PK	Identifier
LANE	Lane
DATE	Date:YYYYMMDD
HOUR	Hour of the day (integer)
TOTAL	Total number of vehicles
CLASSE_A	Motorbikes counted
CLASSE_B	Cars counted
CLASSE_C	Trucks counted
CLASSE_D	Buses counted
CLASSE_O	Other vehicles counted

6. Experimentation

In this section the experimentation performed about the predictions described in the previous section is shown. To perform these tests an Intel Core i5 2410 laptop, with 2.30 GHz and a RAM of 4 GB has been used. As has been previously mentioned, the default parametrization provided be KEEL has been used for the C4.5. This section is divided into two subsections, the first one (Section 6.1) related with the first scenario, and the second one with the pollution scenario (Section 6.2).

6.1. Experimentation with the first scenario

Data collected by sensors located in the A5 highway, connecting Lisbon and Cascais (Portugal) were provided by BRISA. Data was referred to November 1st, 2014 to November 30th, 2014. In order to make data suitable for its use by the learning algorithms, and with the aim of incorporating them to the CLU, data was saved as a text file. Data contained the attributes explained in Table 7.

Is important to note that PK value codes both the position of the sensor measuring the pass of vehicles, and the direction. PK can obtain the 12 different values listed in Table 8. Values code the kilometric point, starting from Lisbon, in which the sensor station is located, as well as the direction: C means Crescente, i.e. PK ascending direction (Lisbon to Cascais), D means Decrescente (Cascais to Lisbon).

Moreover, since data is received lane by lane, aggregated data is needed to be extracted. For this, the maximum LANE value for each one of the PK represents the total number of lanes. Here, it is important to note that sensor stations count vehicles in both senses, so the process of separating lanes from one direction from the other one was implemented.

In Figure 6 a visual example of this situation is represented. In this case, two possible sensor station located at PK=X and PK=Y are represented. Both sensors monitor all the 4 lanes in the road, but each one of them is in a side of the road (denoted by the DIRECTION). In this case, it would be erroneous to use the sum of the four values measured by the station, since not all the sensors are located in the same part of the highway. Instead of that, single sensors are separated in

Table 8 Different values that the PK attribute can take.

ID	Distance from Lisbon	Direction
'A5 PK 0+400_C'	400 m	Lisbon-Cascais
'A5 PK 0+600_D'	600 m	Cascais-Lisbon
'A5 PK 1+505_D'	1505 m	Cascais-Lisbon
'A5 PK 1+980_C'	1980 m	Lisbon-Cascais
'A5 PK 3+600_D'	3600 m	Cascais-Lisbon
'A5 PK 4+000_D'	4000 m	Cascais-Lisbon
'A5 PK 6+800_C'	6800 m	Lisbon-Cascais
'A5 PK 7+100_D'	7100 m	Cascais-Lisbon
'A5 PK 8+050_C'	8050 m	Lisbon-Cascais
'A5 PK 9+400_D'	9400 m	Cascais-Lisbon
'PK20+900'	20900 m	Lisbon-Cascais
'PK22+600'	22600 m	Lisbon-Cascais

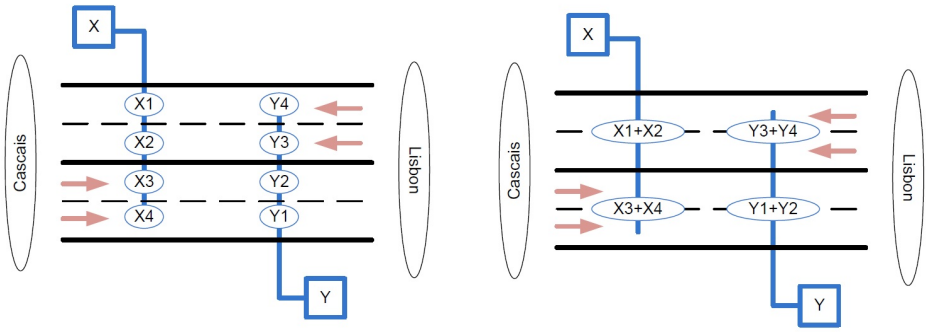

Fig. 6. *Example scenario for lane conversion.*

function of the real direction of the road in which they are located (not the one of the station), and aggregated accordingly.

In this way, aggregated values are added to the dataset under the notation $sum\_C\{A,B,C,D,O\}$ (denoting $CLASSE\_\{A,B,C,D,O\}$ respectively), and non-useful any more information is deleted. Since each one of the sensors presented in Table 8 has been spliced depending on the direction of the lanes, and then aggregated, 24 measure points are available.

With the aim of training (in a supervised way) the used C4.5, for each one of the measure points a value of congestion predicted for the next hour is calculated as follows:

- LOW: the total number of vehicles counted is below the percentile 25 measured by the sensor.
- MEDIUM (MED): the number of vehicles is above percentile 25 but below percentile 50.
- HIGH: the total number of vehicles counted is above the percentile 50.

This criterion has been taken so that the congestion level is variable depending on the position of the sensor. This approach allows the system to predict the congestion without the need of any additional attribute, such as the speed limit. In this way, prediction can be made faster, and the system can have a more efficient performance. Additionally, and thanks to this approach, we do not need any initial configuration of the system. Furthermore, a manual setup can be made whether it is necessary.

Once calculated that, the level of congestion reached in the following hour is added to the data, resulting in the format presented in Figure 7.

Where values in the different columns represent:

- Day of the week [1,7]: Monday to Sunday.

```
7, 0, 22, 1222, 1, 0, 0, 1245, MED
7, 1, 25, 948, 0, 0, 3, 976, MED
7, 2, 14, 649, 1, 0, 2, 666, MED
7, 3, 9, 480, 2, 0, 0, 491, MED
7, 4, 3, 348, 1, 0, 0, 352, LOW
```

Fig. 7. *Data format for its use by C4.5 technique.*

- Hour [0,23]: Hour of the day.
- Sum_CA [0,254]: Number of motorbikes counted during the last hour.
- Sum_CB [43,3198]: Number of cars counted during the last hour.
- Sum_CC [0,61]: Number of trucks counted during the last hour.
- Sum_CD [0,27]: Number of buses counted during the last hour.
- Sum_CO [0,554]: Number of other types of vehicles counted during the last hour.
- Total [49,3444]: Total number of vehicles counted during the last hour.
- Next level ($\{$LOW,MED,HIGH$\}$): Level of congestion achieved during next hour.

With all this, 23 datasets have been generated in overall to feed the C4.5 algorithm implemented under the KEEL framework. Training and test partitions were done by using the 3 first weeks of the month for building the models, as well as the last week to validate it (test partition). C4.5 returns a tree formed by concatenated ifs that reach to the final state of congestion predicted. In Table 9 it can be seen the percentages of success obtained by the trees generated for each one of the different datasets. As can be seen in this table, in most of the cases, the accuracy in the prediction of the level of congestion is above 95%.

Regarding runtimes, despite it is not a critical issue in this experimentation, it is interesting to mention that every execution of the C4.5 needs less than a second to build a model. This runtime is more than enough to permit the system work in a proper way. Additionally, every dataset is composed by data obtained in one month. Although being sufficient for these scenarios, a promising scalability of the technique could be expected.

It is important to highlight that the decision trees built by the C4.5 can be read by the CLU, in order to execute its codified logic using the actual state of the road as input. Thus, it can to provide predictions about the traffic density during the next hour.

6.2. Experimentation with the second scenario

In this scenario, data regarding pollution measurements in the city centre of Pisa were received from INTECS. Files contained data regarding hourly measures for years 2012 and 2013. As in the previous case, data was saved in a text format, in order to make it easier to automatically process. For each measurement, 6 different values are used to feed the classification technique: day of the week, hour, and one value for each kind of pollution measured (NO2, NO, NOX and O3). These pollutions are measured in $\mu g/m^3$ at 20°C. Figure 8 shows the final data format used.

For this study, the levels of pollution are considered high in case three out of the four levels overpass the percentile 66 of the measures. Additionally, levels are assumed medium when more

Table 9 Results obtained by designed decision trees in prediction of the next level of traffic.

Dataset	Number of leaf nodes in the tree	Average deep of the nodes	% of matching over the training dataset	% of matching over the test dataset
CL_400	17	5.58	97.59	98.89
CL_20900	28	5.89	97.22	91.16
CL_22600	9	4.44	97.03	95.02
LC_400	14	4.21	99.44	96.13
LC_600	22	6.22	94.44	86.74
LC_1505	8	3.25	97.59	94.47
LC_1980	3	1.66	93.24	93.90
LC_3600	16	4.50	98.14	97.23
LC_4000	3	1.66	92.96	92.79
LC_6800	14	4.50	99.44	97.23
LC_7100	18	6.16	94.81	90.60
CL_600	24	7.00	94.44	92.26
LC_8050	5	2.40	92.59	91.96
LC_20900	17	5.00	96.66	88.39
LC_22600	11	3.90	98.88	95.58
CL_1505	14	4.64	98.70	95.02
CL_1980	3	1.66	91.66	92.24
CL_3600	20	5.80	97.59	93.92
CL_4000	3	1.66	91.66	93.07
CL_6800	25	5.96	96.66	87.29
CL_7100	19	5.00	98.33	96.13
CL_8050	5	2.80	91.48	91.68
CL_9400	9	3.77	99.81	87.84

DAYOFWEEK	Hour	NO2	NO	NOX	O3
1	2	43	10	58	5
1	3	40	5	47	5
1	4	37	5	45	5
1	6	29	1	31	9
1	7	29	3	34	11
1	8	38	12	57	3

Fig. 8. *Data format for its use by C4.5 technique.*

than two levels overpass the percentile 33 of the registered measures. In any other case, the pollution is considered low.

As has been pointed before, C4.5 has been used to build decision trees able to determine the next value of pollution, given the current measures of data. For this case, 5 partitions were built following the cross validation process: split the data in 5 pieces and use each one for testing the solution, after training with the remaining 80%. Experimental results are presented in Table 10. As can be seen in this table, in most of the cases, the accuracy in the prediction of the level of congestion is about 75%. Regarding runtimes and scalability of the technique, same results and conclusions explained in the previous scenario can be drawn.

As has been previously said, the decision trees generated by the C4.5 can be read by the CLU,

Table 10 Results obtained by designed decision trees in prediction of the next level of pollution.

Dataset	Number of leaf nodes in the tree	Average deep of the nodes	% of matching over the training dataset	% of matching over the test dataset
Partition1	305	11.01	80.99	77.32
Partition2	353	11.44	83.58	72.22
Partition3	291	10.88	80.87	78.89
Partition4	323	11.22	82.91	73.88
Partition5	286	11.69	81.52	75.94

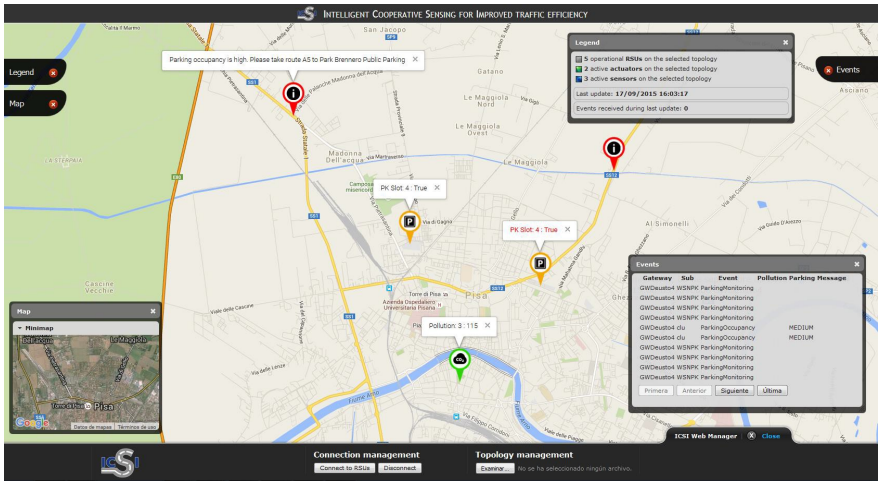

Fig. 9. *ICSI Demo Web application.*

in order to execute its codified logic using the actual state of the road as input. In this way, it can provide predictions about the level of pollution during the next hour. These predictions can be useful for taking premature actions on the access to the restricted area.

7. Demostration

A working platform has been created for the demonstration, which is integrated into the software architecture described in Section 2. This demonstration environment is composed of two main elements: a web service, to manage the communication between the ICSI architecture and the demo environment, and the ICSI Demo Web application. This web application (Figure 9) is able to display the complete operation of the CLUs in a real scenario simulated in laboratory: from the configuration of the CLUs and the generation and loading of the topology files, to the real-time visualization of ICSI events that the application receives from the ICSI Demo Web Service. It should be pointed that for the simulations performed, the data described in Section 4 have been used.

The web manager is a Single Page Application developed in C# with a map-based interface based on Google Maps API. The application is a RIA (Rich Internet Application) [33] and follows MVC architecture. For the presentation layer design HTML5, CSS3 and client programming languages like JavaScript, JQuery and Ajax have been used. A GUI has been also developed for the topology management. Starting with an OSM (Open Street Maps) map data file, the user can set the different elements of the topology and its position in a map enabling the easy replication of the simulated scenario. After setting the scenario, the application starts listening via the provided Web Service for ICSI event updates. Each time an event is received it is displayed in the corresponding GW event list. An information box is also displayed with the actual value of the event data (i.e. the received pollution value or an action message alerting the drivers about imminent congestion).

It is important to understand the contingency plans developed for both scenarios. These contingency plans store information about the set of actions needed to implement the most accurate solution to recover from a specific traffic condition. It establishes the actions to perform in the reception of an event classifying it according to its severity.

Table 11 shows the set of rules that produce the contingency plan configuration file for the

Table 11 Contingency Plan set of rules for the Highway scenario.

Gateway	Received information	Action
GW1,2,3	AbnormalTraffic = ACCIDENT	Emergency Call. Alert drivers to drive carefully.
GW1,2,3	Vehicle counter + Historical info	Set (predicted) Congestion Level
GW1,2,3	CongestionLevel = HIGH	Alert drivers. Suggest alternative routes to avoid the traffic jam.

Table 12 Contingency Plan set of rules for the Urban scenario.

Gateway	Received information	Action
GW1,2	PollutionLevel == HIGH	Reroute to parking. Extra info about public transport routes.
GW1,2	ParkingOccupancy == HIGH	Reroute to the other parking area. Recommend the use of public transport.
GW3,4	Free parking slots	Set Parking Occupancy.
GW5	Pollution data + Historical info	Set (predicted) Pollution Level.

analysed highway scenario (Section 4.1). Each GW will continuously monitor *Abnormal Traffic* events. In case of an accident, the GW makes an immediate emergency call to the medical services indicating the position of the vehicle.

Regarding the *Vehicle Counter* event, the CLU receives the information from its attached sensors and delegate to the C4.5, that uses historical information about congestion levels (described in Section 6.1), to produce an event indicating the expected *Congestion Level*. These *Congestion Level* events will be also received by the GWs coming from the next GWs on the road, so that the GWs can inform drivers in advance in case of an expected traffic jam and provide alternative routes accordingly.

Furthermore, Table 12 shows the set of rules that produce the contingency plan for the urban scenario (Section 4.2). GW3 and GW4 continuously monitor the parking lots status, setting the *Parking Occupancy* according to the total number of free parking slots. When the parking occupancy is high (more than 90% of occupancy) the GWs send an event to the DDP. This *Parking Occupancy* event is received by the GWs 1 and 2 so that the GWs can inform drivers providing them alternative parking slots near their position and/or recommending commuting with public transport.

Regarding the *Pollution* event, GW5 receives the information from its attached sensors and delegate to the implemented C4.5, which use historical information about pollution levels (described in Section 6.2) to produce an event indicating the expected *Pollution Level*. This *Pollution Level* event is also received by the GWs 1 and 2 so that the GWs can inform drivers in advance in case of an expected close of the LEZ providing alternative parking lots near their position and/or recommending commuting with public transport.

8. Conclusions

In this paper, based on the ICSI European project, a decentralized ITS with intelligence based on classification techniques has been presented. This architecture, with the participation of the sensors, the DDPs, the CLUs and, finally, the ITS applications, encompasses the entire process of capture and management of available road data, enabling the generation of services to promote transportation efficiency.

In this research, how the architecture works in two different scenarios related with the prediction of the traffic congestion and pollution is shown. These scenarios are based in real world situations, the first one in Lisbon (Portugal), and the second one in Pisa (Italy). Additionally, the well-known C4.5 algorithm has been used for the accurate prediction of the traffic congestion and the pollution level.

Besides that, the developed demonstration environment has been presented. This environment counts with a web application which is able to display the complete operation of the CLUs in a real scenario simulated in a laboratory.

As future work, we have planned to perform a more complete simulation environment, which will be able not only to offer services of prediction, but to provide services related with vehicle routing optimization and regulation services. Additionally, it is planned to continue the work described in this paper with the real trials which will be conducted in Lisbon and Pisa.

In this paper, prediction problems have been faced. In the near future, we have planned to deal with vehicle routing optimization, where the authors of this study have a wide experience [34, 35]. Currently, this field is one of the most studied ones in the scientific community. The problems arisen in this field have a great scientific interest, since such NP-Hard problems present a tough challenge to solve for the scientists. Furthermore, their social interest is also high, as their applicability to real-world situations is great. Though some appropriate methods can be found in the literature to address such complex problems, arguably the most successful techniques to solve vehicle routing problems are the heuristics and metaheuristics. For this reason, as future work, we will solve real-world routing problems, providing the existing CLUs with approaches such as Genetic Algorithms [36] or Particle Swarm methods [37].

9. Acknowledgments

This work has been partially funded by the European Commission under the Seventh Framework Programme. Intelligent Cooperative Sensing for Improved Traffic efficiency (ICSI) project. Grant agreement no.: 317671.

10. References

[1] Weiland, R.J., Purser, L.B.: Intelligent transportation systems. Transportation in the new millennium (2000)

[2] Weiß, C.: V2x communication in europe–from research projects towards standardization and field testing of vehicle communication technology. Computer Networks **55**(14) (2011) 3103–3119

[3] Papadimitratos, P., La Fortelle, A., Evenssen, K., Brignolo, R., Cosenza, S.: Vehicular communication systems: Enabling technologies, applications, and future outlook on intelligent transportation. Communications Magazine, IEEE **47**(11) (2009) 84–95

[4] Toulminet, G., Boussuge, J., Laurgeau, C.: Comparative synthesis of the 3 main european projects dealing with cooperative systems (cvis, safespot and coopers) and description of coopers demonstration site 4. In: International IEEE Conference on Intelligent Transportation Systems. (2008) 809–814

[5] Alexander, P., Haley, D., Grant, A.: Cooperative intelligent transport systems: 5.9-ghz field trials. Proceedings of the IEEE **99**(7) (2011) 1213–1235

[6] CVIS: Cooperative vehicle-infrastructure systems. `www.ecomove-project.eu/links/cvis/` (accessed on 18 July 2016)

[7] Compass4D. http://www.compass4d.eu/en/about/ (accessed on 18 July 2016)

[8] SAFESPOT. www.safespot-eu.org (accessed on 18 July 2016)

[9] COMeSafety2. www.ecomove-project.eu/links/comesafety/ (accessed on 18 July 2016)

[10] DRIVE-C2X: Accelerate cooperative mobility. www.drive-c2x.eu/project (accessed on 18 July 2016)

[11] Quinlan, J.R.: C4.5: programs for machine learning. Elsevier (2014)

[12] Moreno, A., Onieva, E., Perallos, A., Iovino, G., Fernández, P.: Cooperative decision-making its architecture based on distributed rsus. In: Ubiquitous Computing and Ambient Intelligence. Sensing, Processing, and Using Environmental Information. Springer (2015) 84–90

[13] Dunham, M.H.: Data mining: Introductory and advanced topics. Pearson Education India (2006)

[14] Bailey, K.: Typologies and taxonomies: an introduction to classification techniques. Newbury Park, CA: Sage Publications (1994)

[15] Kotsiantis, S.: Supervised machine learning: A review of classification techniques. In: Proceedings of the 2007 conference on Emerging Artificial Intelligence Applications in Computer Engineering: Real Word AI Systems with Applications in eHealth, HCI, Information Retrieval and Pervasive Technologies, IOS Press (2007) 3–24

[16] Wang, S.C.: Artificial neural network. In: Interdisciplinary Computing in Java Programming. Springer (2003) 81–100

[17] Steinwart, I., Christmann, A.: Support vector machines. Springer Science & Business Media (2008)

[18] Quinlan, J.R.: Induction of decision trees. Machine learning **1**(1) (1986) 81–106

[19] Razavi, B.S.: Predicting the trend of land use changes using artificial neural network and markov chain model (case study: Kermanshah city). Research Journal of Environmental and Earth Sciences **6**(4) (2014) 215–226

[20] Goodman, K.E., Lessler, J., Cosgrove, S.E., Harris, A.D., Lautenbach, E., Han, J.H., Milstone, A.M., Masey, C., Tamma, P.D.: A clinical decision tree to predict whether a bacteremic patient is infected with an esbl-producing organism. Clinical Infectious Diseases (2016) ciw425

[21] Rebentrost, P., Mohseni, M., Lloyd, S.: Quantum support vector machine for big data classification. Physical review letters **113**(13) (2014) 130503

[22] Quinlan, J.R., et al.: Discovering rules by induction from large collections of examples. Expert systems in the micro electronic age. Edinburgh University Press (1979)

[23] Mašetic, Z., Subasi, A.: Detection of congestive heart failures using c4. 5 decision tree. Southeast Europe Journal of Soft Computng **2**(2) (2013)

[24] Yao, Z., Liu, P., Lei, L., Yin, J.: R-c4. 5 decision tree model and its applications to health care dataset. In: Services Systems and Services Management, 2005. Proceedings of ICSSSM'05. 2005 International Conference on. Volume 2., IEEE (2005) 1099–1103

[25] Adhatrao, K., Gaykar, A., Dhawan, A., Jha, R., Honrao, V.: Predicting students'performance using id3 and c4. 5 classification algorithms. International Journal of Data Mining & Knowledge Management Process **3**(5) (2013) 39

[26] Onieva, E., Lopez-Garcia, P., Masegosa, A., Osaba, E., Perallos, A.: A comparative study on the performance of evolutionary fuzzy and crisp rule based classification methods in congestion prediction. Transportation Research Procedia **14** (2016) 4458–4467

[27] Cortes, C., Vapnik, V.: Support-vector networks. Machine learning **20**(3) (1995) 273–297

[28] Rastogi, R., Shim, K.: Public: a decision tree classifier that integrates building and pruning. In: VLDB. Volume 98. (1998) 24–27

[29] Gray, J.B., Fan, G.: Classification tree analysis using target. Computational Statistics & Data Analysis **52**(3) (2008) 1362–1372

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

[31] Carvalho, D.R., Freitas, A.A.: A hybrid decision tree/genetic algorithm method for data mining. Information Sciences **163**(1) (2004) 13–35

[32] Derrac, J., García, S., Molina, D., Herrera, F.: A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. Swarm and Evolutionary Computation **1**(1) (2011) 3–18

[33] Fraternali, P., Rossi, G., Sánchez-Figueroa, F.: Rich internet applications. Internet Computing, IEEE **14**(3) (2010) 9–12

[34] Osaba, E., Yang, X.S., Diaz, F., Lopez-Garcia, P., Carballedo, R.: An improved discrete bat algorithm for symmetric and asymmetric traveling salesman problems. Engineering Applications of Artificial Intelligence **48** (2016) 59–71

[35] Osaba, E., Yang, X.S., Diaz, F., Onieva, E., Masegosa, A.D., Perallos, A.: A discrete firefly algorithm to solve a rich vehicle routing problem modelling a newspaper distribution system with recycling policy. Soft Computing (2016) 1–14

[36] Goldberg, D.: Genetic algorithms in search, optimization, and machine learning. Addison-Wesley Professional (1989)

[37] Kennedy, J., Eberhart, R., et al.: Particle swarm optimization. In: Proceedings of IEEE international conference on neural networks. Volume 4., Perth, Australia (1995) 1942–1948