On the Analysis of Machine Learning Classifiers to Detect Traffic Congestion in Vehicular Networks

Lucas Augusto de Carvalho[1], Maycon Ferreira da Silva[1], Edimilson B. dos Santos[1], Daniel Ludovico Guidoni[1],

[1]Department of Computer Science – Federal University of São João del-Rei (UFSJ)
São João del-Rei – MG – Brazil

{lucas.carvalho, maycon, edimilson.santos, guidoni}@ufsj.edu.br

Abstract. *Problems related to traffic congestion and management have become common in many cities. Thus, vehicle re-routing methods have been proposed to minimize the congestion. Some of these methods have applied machine learning techniques, more specifically classifiers, to verify road conditions and detect congestion. However, better results may be obtained by applying a classifier more suitable to domain. In this sense, this paper presents an evaluation of different classifiers applied to the identification of the level of road congestion. Our main goal is to analyze the characteristics of each classifier in this task. The classifiers involved in the experiments here are: Multiple Layer Neural Network (MLP), K-Nearest Neighbors (KNN), Decision Trees (J48), Support Vector Machines (SVM), Naive Bayes and Tree Augmented Naive Bayes.*

1. Introduction

The growing of vehicles on streets and highways has caused congestion and traffic management problems. Considering some factors, such as time and location, especially in large centers, congestion results in wasted time, fuel as well as people's quality of life [Djahel et. al., 2015].

Another problem is the suggestion of individual alternative routes that create new traffic bottlenecks simply transferring congestion from one point to another. In this scenario, Intelligent Transport Systems emerged to solve several problems in traditional transportation systems using new technologies of information and communication.

An intelligent transport system relies on information obtained from vehicular networks or Vehicular Ad hoc NETworks (VANETs). The vehicular networks are a special type of mobile ad hoc networks composed of vehicles with onboard units (with storage, processing and communication capabilities) and fixed infrastructures with wireless capabilities deployed at the roadside, named RoadSide Unit (RSU) [Bila et. al., 2017][Cunha et. al., 2016]. One of the main characteristics of this type of network is the high node mobility (vehicles), intermittent links among vehicles and strict latency requirements [Araujo et. al., 2014]. These networks use information from vehicles within a short time, propagating quickly such information and thus enabling vehicle routing applications to avoid congestion. A major challenge for these applications is to change the route of some vehicles without transferring the congestion to other places.

One way to avoid the creation of traffic bottleneck is to correctly classify which roads are congested, by then selecting non-congested roads to create alternate routes.

Machine learning techniques have been applied in order to classify and reduce the problem of congestion transfer from one point to another. More specifically, methods that perform vehicle re-routing (alternative paths for vehicles inside a congestion area) [Meneguette et. al., 2016][Souza et. al., 2016][Van den Haak et. al., 2010] have been proposed. These methods apply classifiers to verify road conditions and assess the level of congestion before and after re-routing.

It is necessary to emphasize that different classifiers can present different results in each domain. Also, it is always important to remember that there is no classifier better than all the others on all the problems, as discussed in [Wolpert, 1996]. Therefore, for each problem, one must select the right algorithm. In this sense, this paper presents an evaluation of state-of-art classification algorithms applied to the identification of the level of road congestion. In order to evaluate and identify the behavior of the classifiers in different traffic scenarios, we performed experiments with six classification algorithms: Multilayer Perceptron neural network (MLP), K-Nearest Neighbors (KNN), Decision Trees (J48), Support Vector Machines (SVM), Naive Bayes and Tree Augmented Naive Bayes (TAN). The obtained results demonstrate that the classifiers are significantly different and thus the choice of the classifier may influence the result of a re-routing method.

The remainder of this paper is organized as follows. In section 2, the theoretical reference about classification algorithms is presented. Section 3 presents some related works and Section 4 describes the development methodology. Section 5 presents the analysis of results. Finally, section 6 presents the conclusions and future work.

2. Background

Classification is a basic task in data analysis and pattern recognition and requires the construction of a classifier, that is, a function that assigns a class variable to an instance described by a set of attributes [Cirelo and Cozman, 2005]. There are several approaches available in the literature to build a classifier. Each approach is capable of inducing efficient classifiers in certain application domains. However, there is no an approach which induces an efficient classifier in all application domains. Each classifier has its specific characteristics, such as the adopted inductive bias, and may have different behavior in each domain. Thus, some classifiers can achieve better results than others in a specific domain.

Following, some state-of-art classification algorithms, which are applied here to classify the level of road congestion, are presented.

2.1. Multilayer Perceptron Neural Networks

Artificial Neural Networks (ANN) are a system composed of relatively simple processing elements, called nodes or units, which are connected, forming a network.

According to the structure, the units of an ANN can be arranged in layers, such that each unit receives input only from units in the immediately preceding layer. A Multilayer Perceptron Neural Network (MLP) has one input layer, one or more hidden layers, and the output layer [Russel and Norvig, 2013]. The input layer receives the values of the input attributes. The hidden layer does most of the processing, using weighted connections. The output layer has the final result of the neural network processing (forecast response).

ANN are suited for practical problems such as handwriting recognition, spoken word recognition, and face recognition. Some of their main characteristics are the robustness to noise, ability to learn with examples and the speed of processing. However, the learning often requires a high processing time. In addition, it is difficult to define the appropriate number of hidden layers of the structure to represent a specific problem, as well as the number of hidden units in each layer.

2.2. K-Nearest Neighbours

The K-Nearest Neighbors (KNN) algorithm is based on the concept of neighborhood, in which the neighbors are similar. Thus, it is possible to classify the elements of an n-dimensional space into K sets. This parameter K represents the number of neighbors and is defined by the user in order to obtain a better classification.

This method requires little effort for training and it does not depend on the construction of a structure to store knowledge. The algorithm stores the training examples in memory. However, it takes more time to classify a new element. This is due to the fact of comparing the new information with all the examples of the training set in the worst case.

According to Russel and Norvig (2013), the classifier can get good results when there is an abundance of data in a low dimension (domains with few attributes). However, in large dimensional spaces, usually the closest neighbors are distant.

2.3. Decision Trees

Decision Trees is one of the most successful methods of machine learning [Russel and Norvig, 2013]. A decision tree represents a function that receives as input a vector of attribute values and returns a unique output value (a "decision").

The decision tree returns a response after executing a test sequence. Each internal node in the tree corresponds to a test of the value of one input attribute and the branches of the nodes are sorted with the possible values of the attribute. Each leaf node in the tree specifies the value to be returned by the function.

Decision tree learning methods are robust to noise in training data. However, some functions cannot be represented concisely, although it is always preferable to obtain the smallest possible tree that is consistent with the training examples.

2.4. Support Vector Machine

The Support Vector Machines (SVMs) has become popular, especially when there is no prior expert knowledge about a domain [Russel and Norvig, 2013].

SVMs construct a decision boundary (maximum margin separator) with the greatest possible distance to example points. The idea of SVMs is to focus on points more important than others that lead to the best generalization. For this, a linear separation in hyperplane is created, even if the data are not separable linearly in the original input space, because they can incorporate the data in a space of superior dimension, using kernel trick. The linear dimension separator is actually nonlinear in the original space.

SVMs are robust in domains with many attributes (large dimensionality). The main limitations of SVMs are their sensitivity to parameter value choices and the interpretation difficulty of the generated model.

2.5. Naive Bayes

Naive Bayes (NB) is a classifier based on a strong constraint: all attributes are considered independent, given the value of the class variable. The structure of NB is fixed so that only the class variable is parent of the other attributes and it is not necessary to induce it from data. Thus, an NB classifier is automatically obtained by induction of the numerical parameters of the model, which requires only information about the attributes and their corresponding values to estimate the probabilities.

Although NB has provided good results in several domains [Friedman et. al., 1997], its estimates of probability are unrealistic and its classification performance can be improved. Moreover, the induced NB model cannot capture the actual relationships between the attributes, and thus it is considered a simple model.

2.6. Tree Augmented Naive Bayes

In [Friedman et. al., 1997], the authors introduced the Tree Augmented Naive Bayes (TAN) algorithm in order to obtain better classifiers than the NB algorithm. For this, the TAN algorithm relaxes the constraint imposed in the construction of the NB structure and considers the dependencies among the other attributes, in addition to the class attribute. Thus, in the structure of a TAN classifier, the class attribute has no parents and each attribute has as its parent the class attribute and at most one other attribute.

3. Related Work

There are vehicle re-routing methods in the literature that apply classifiers to identify road congestion. However, these methods do not consider the individual characteristics of the applied classifiers since the main interest is only in the traffic management.

In the work of [Meneguette et. al., 2016], a new solution called CARP (Congestion AwaRe Protocol) was proposed for the detection and control of congested roads based on inter-vehicle communication. CARP uses ANN to detect and classify congestion levels on highways. Simulation results showed that the proposed solution has obtained a relevant rate of adjustment in the level of congestion, reducing CO2 emission, fuel consumption and travel time in the simulated scenarios.

In [Souza et. al., 2016], a cooperative routing solution called CO-OP is proposed and the solution applies the KNN algorithm to classify the congestion levels of roads. The results showed the effectiveness of the method, where it was able to reduce the average travel time, average congestion time.

In [Van den Haak et. al., 2010], the authors present a hybrid model that applies Bayesian networks to predict traffic congestion. The model was tested at a major transit point in Amsterdam. The experiments involved a comparison between the proposed model and Naive Bayes. The authors concluded that the results of the hybrid model are promising and outperform the Naive Bayes models.

4. Methodology

This work presents a study on the application of classifiers in the task of identifying the level of road congestion. The evaluated classifiers are: Multilayer Perceptron Neural Network (MLP), K-Nearest Neighbors (KNN), Tree Decision (J48), Support Vector Machine (SVM), Naive Bayes and Tree Augmented Naive Bayes (TAN). These classifiers were trained with information obtained from simulations performed with the SUMO[1] simulator.

SUMO (Simulation of Urban Mobility) is a simulator of VANET, which allows simulating the traffic of vehicles and thus possible congestions. SUMO provides information on vehicles and roads in a traffic simulation, both on urban roads and highways, allowing the generation of datasets for the induction of classifiers.

To simulate vehicle traffic, SUMO allows representing the roads as a Grid (see Figure 1). There are back-and-forth traffic routes where vehicles can transit inside the Grid. When the vehicle can not proceed on a path due to congestion or to give preference to other vehicles in an intersection, they are stopped one after another. Vehicles stopped in this way cause congestion on that road.

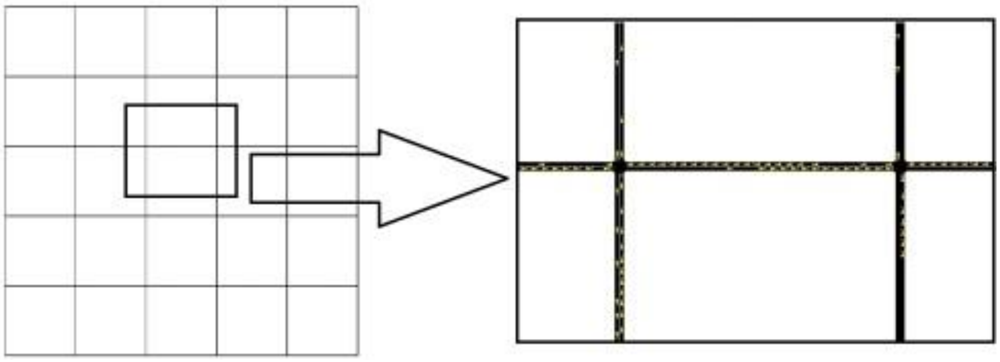

Figure 1. Example of a Grid 6x6 provided by SUMO to represent the vehicles traffic.

For simplicity, in this paper, the Grid was set to equal sizes roads during traffic simulations by SUMO. From this, three Grid configurations were performed to obtain the necessary information for the construction of the datasets. In the first Grid configuration, the roads were defined as 200 meters and the vehicles had a maximum speed of 13.9 m/s (50.04 km/h). The simulation performed with this Grid made it possible to generate the dataset called "Base200". In the second Grid configuration, it was defined that the roads had 400 meters and the vehicles had a maximum speed of 9.9 m/s (35.64 km/h). The simulation performed with this Grid made it possible to generate the dataset called "Base400". Finally, the Grid was configured with the roads having 1000 meters and a maximum speed of 19.9 m/s (71.64 km/h). Thus, the dataset called "Base1000" was built.

Initially, the datasets have instances representing traffic routes with information of two attributes, according to some works found in literature [Araujo et. al.,

[1] <http://sumo.dlr.de/wiki/Simulation_of_Urban_MObility_-_Wiki>

2014][Meneguette et. al., 2016]: i) average speed of the roads and ii) the number of vehicles present in this road. These two attributes allow us to classify the level of congestion as: weak, moderate and severe. Next, two other attributes were added: (iii) the time that the vehicle was stationary on the road and (iv) the road occupancy rate. The addition of two more attributes to datasets aimed at providing more information to define congestion and to facilitate the learning of classifiers.

Table 1 shows the number of instances (examples of routes) entered in each dataset, representing different levels of congestion (having 2 and 4 attributes). In each traffic route (instance), the level of congestion (weak, moderate and severe) was identified as follows:

- weak: up to 33.333% of the maximum number of vehicles and a reduction in 33.333% of the maximum speed;
- moderate: between 33.334% and 66.666% of the maximum speed of the road and the maximum number of vehicles;
- severe: more than 66.667% of the maximum number of vehicles and a reduction more than 66.667% of the maximum speed.

Table 1. Number of instances in training datasets.

Datasets	Road size (meter)	Weak	Moderate	Severe	Total of Instances
Base200	200	5513	1607	3236	10356
Base400	400	5447	2685	2098	10230
Base1000	1000	3738	2679	4503	10920

The classification algorithms have been trained with the datasets of Table 1 to induce the classifiers. After, these classifiers were evaluated with a test dataset which was also built using the SUMO simulator. This test dataset contains information obtained from a map of the city of Los Angeles, USA.

Figure 2 presents an outline of the Los Angeles map and shows how different it is from the Grid adopted to represent traffic. It is possible to notice the streets with different sizes, allowing varying the maximum capacity of vehicles in each route and their maximum speeds.

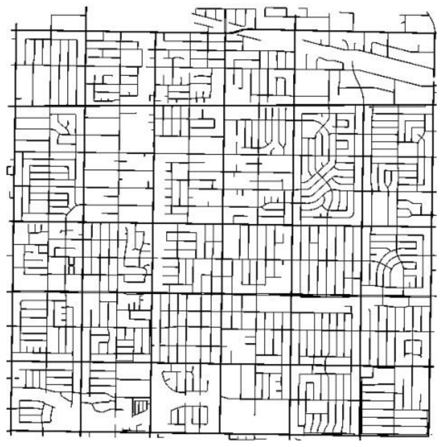

Figure 2. Outline of the Los Angeles city map.

In this scenario, there are about 470 different path sizes – some only appear once – and up to 6 maximum speed variations. There is no direct relationship between road sizes and vehicle speed. Using this dataset for testing, with diversified information, it is possible to analyze with more confidence the results of the classifiers, which are discussed in Section 5.

The classification algorithms have been trained and tested using Weka[2] software [Hall et. al., 2009]. Weka (Waikato Environment for Knowledge Analysis) is a collection of machine learning algorithms implemented in Java and free code. Thus, it has been possible to perform several experiments to define some learning parameters of the algorithms in search of the configuration that has obtained the best results.

The J48 (decision tree), Naive Bayes, TAN and SVM algorithms have been executed with the Weka default parameters. The KNN has been configured with K = 3 (number of neighbors) and Euclidean distance function. For neural networks (MLP), we have defined: i) a structure with one hidden layer having two units, when the training dataset contain 2 attributes; and ii) a structure with one hidden layer having 3 units, when the training dataset contain 4 attributes.

5. Experiments and Analysis of Results

The experiments performed with the classification algorithms have used the datasets of Table 1 for training and the dataset obtained from the Los Angeles map for validation (described in Section 4).

Initially, the classifiers were trained using the Base200, Base400 and Base1000 datasets having two attributes: i) average speed of the road and ii) the number of vehicles present in this road. After, some experiments were carried out joining the data of these 3 datasets to diversify information: first the Base200 and Base400 data were merged (Base200-400), after Base200 and Base1000 (Base200-1000) and then Base400 and Base1000 (Base400-1000). Finally, all three datasets have been joined on one dataset (Base200-400-1000). Then, the classifiers were trained using the datasets having 4 attributes: i) average speed of the road and ii) the number of vehicles present in this road, iii) the time that the vehicle was stationary on the road and (iv) the road occupancy rate. In this case, the datasets were also merged. Sections 5.1 e 5.2 present and discuss the results obtained in the experiments using the datasets having two and four attributes, respectively.

After training, the classifiers were evaluated with the test dataset of the city of Los Angeles, which provided scenarios closer to reality. In this evaluation, three different routes were configured to represent the traffic scenarios. In each route, the start point, end point and path of each vehicle are changed. Thus, three distinct scenarios that have allowed the presence or absence of congestion in different points of the city were created. In each scenario, the classification rate of the congestion level of Los Angeles city roads obtained by each classifier was stored.

[2] <http://www.cs.waikato.ac.nz/ml/weka/>

5.1. Training using two-attributes datasets

In these experiments, the classifiers were trained using the datasets having two attributes. After, the classifiers were assessed using the Los Angeles dataset. Tables 2, 3 and 4 present the classification rates obtained by the classifiers in scenarios 1, 2 and 3, respectively.

Table 2. Classification rate obtained by classifiers (when trained using dataset with two attributes) in the Los Angeles dataset in first scenario.

Dataset	**NB**	**TAN**	**KNN**	**MLP**	**J48**	**SVM**
Base200	**62.55**	62.02	62.09	62.02	62.01	61.78
Base400	59.74	**60.88**	59.51	59.51	59.51	59.51
Base1000	59.21	59.21	59.21	59.21	59.21	59.21
Base200-400	**69.6**	61.71	61.49	60.73	61.25	61.41
Base200-1000	**79.61**	61.71	61.71	60.05	61.71	61.71
Base400-1000	59.44	59.44	59.44	59.44	59.43	59.43
Base200-400-1000	**79.61**	61.71	59.44	59.97	61.33	61.25
Rank (Friedman)	**1.71**	**2.57**	**3.42**	**4.42**	**4.28**	**4.42**

Observing the results presented in Tables 2, 3 and 4, it is possible to notice that the classifiers obtained similar results in the three defined scenarios. This shows that information about the congestion level available in datasets allow to induce classifiers able to evaluate any suggested route. However, all classifiers did not get a good classification rate, especially when they are trained with datasets having single-sized streets (Base200, Base400 and Base1000). This is understandable since the streets of the city of Los Angeles have different sizes. When the information in these datasets is joined (Base200-400, Base200-1000, and Base200-400-1000), there is a reasonable improvement in the accuracy rate of the Naive Bayes classifier. Although this classifier is considered simpler (due to its strong assumption about the independence of attributes, given the class variable), its performance has been higher than the other classifiers with more complex models in the three scenarios.

Table 3. Classification rate obtained by classifiers (when trained using dataset with two attributes) in the Los Angeles dataset in second scenario.

Dataset	**NB**	**TAN**	**KNN**	**MLP**	**J48**	**SVM**
Base200	**58.47**	57.67	57.67	57.67	57.67	57.05
Base400	54.94	**55.82**	54.59	54.5	54.58	54.49
Base1000	54.23	54.23	54.23	54.23	54.23	54.23
Base200-400	**65.7**	57.32	57.14	56.79	56.79	56.79
Base200-1000	**77.25**	57.32	56.97	55.2	56.96	56.96
Base400-1000	54.41	54.41	54.32	54.32	54.23	54.32
Base200-400-1000	**77.42**	56.23	54.23	55.11	56.34	56.43
Rank (Friedman)	**1.57**	**2.5**	**3.71**	**4.57**	**4.21**	**4.42**

The performance of the other classifiers in relation to the Naive Bayes can be explained by the little knowledge on the congestion levels represented in the training datasets. Neural networks (MLP) and SVM generate complex models that need more information to define the appropriate parameters that correctly represent the problem. The decision tree algorithms construct a tree structure based on the information gain of the attributes. The KNN algorithm compares the new examples to be classified (the Los Angeles dataset) with the training dataset examples. The TAN algorithm, because it is an improvement on Naive Bayes, needs to identify the relationships between the attributes. Therefore, the training datasets did not offer sufficient knowledge about congestion levels for these classifiers to obtain more accurate results in the Los Angeles dataset.

Table 4. Classification rate obtained by classifiers (when trained using dataset with two attributes) in the Los Angeles dataset in third scenario.

Dataset	**NB**	**TAN**	**KNN**	**MLP**	**J48**	**SVM**
Base200	**64.83**	64.37	64.37	64.37	64.36	64.06
Base400	61.84	**63.06**	61.61	61.61	61.6	61.6
Base1000	61.38	61.38	61.38	61.38	61.38	61.38
Base200-400	**71.72**	64.06	63.52	63.37	63.6	63.98
Base200-1000	**80.31**	64.06	64.06	62.53	64.06	64.06
Base400-1000	61.53	**61.76**	61.53	61.61	61.53	61.53
Base200-400-1000	**80.54**	63.68	61.53	62.22	63.75	63.9
Rank (Friedman)	**2**	**2.5**	**4.07**	**4.14**	**3.92**	**3.78**

In general, to provide a better perspective on the relative performance of the algorithms under study, we further report the results of comparisons of statistical tests. Following Demšar (2006), we applied the well-known Friedman's statistical test and, when applicable, the post-hoc Nemenyi test to assess the obtained results. Under the null hypothesis, which states that all algorithms are equivalent, we have $p = 0.002$, so we reject the null hypothesis (e.g., for α=5%) and proceed with the Nemenyi post-hoc tests. In this case, it is worth mentioning that the critical difference between six average ranks for the Nemenyi post-hoc test is equal to 2.85. These tests show that the performance of Naive Bayes did not bring significantly different results when compared to TAN, KNN and J48, although there is a tendency. In the second scenario, the Naive Bayes classifier achieved better results than SVM ($4.42-1.57=2.85 \geq 2.85$ – see the last row of Table 3) and the MLP ($4.57-1.57=3 \geq 2.85$). Thus, it is possible to conclude that Naive Bayes can find better solutions than the other algorithms when there is little traffic information.

5.2. Training using four-attributes datasets

In these experiments, the datasets having four attributes have been used for training the classifiers. The addition of two more attributes to datasets aims to bring more information to help improve the learning of the classifiers. However, it is important to note that, in practice, this information would have a high associated cost, as more sophisticated equipment would be needed to collect this data. In these experiments, these costs are not taken into account, since the data were obtained through a simulator.

After training, the classifiers were assessed using the Los Angeles dataset. Observing the results showed in Tables 5, 6 and 7 (scenarios 1, 2 and 3, respectively), it is possible to notice that all the classifiers trained using the datasets having 4 attributes have obtained a higher classification rate than the classifiers trained using the datasets having 2 attributes, except the SVM classifier. As in the previous experiments, the classifiers have also obtained similar results in the three defined scenarios. It is also noticed that the classifiers induced by the decision tree (J48) have obtained the best results in the three scenarios, followed by the MLP. The Naive Bayes classifier has maintained the tendency to get good results when the datasets are merged.

Table 5. Classification rate obtained by classifiers (when trained using dataset with four attributes) in the Los Angeles dataset in first scenario.

Dataset	**NB**	**TAN**	**KNN**	**MLP**	**J48**	**SVM**
Base200	69.83	74	64.9	82.33	**97.19**	30.55
Base400	72.63	68.54	60.42	87.79	**96.81**	59.43
Base1000	59.89	63.91	59.29	62.93	**98.55**	52.76
Base200-400	92.34	68.84	79.98	96.21	**97.8**	28.8
Base200-1000	92.65	83.32	96.47	97.5	**97.57**	40.78
Base400-1000	91.81	68.01	82.11	**98.26**	91.81	47.83
Base200-400-1000	93.18	71.57	95.68	95.83	**97.64**	39.42
Rank (Friedman)	**3.5**	**4.14**	**4.14**	**2**	**1.21**	**6**

Applying Friedman's statistical test, we have p = 0.0001, so we reject the null hypothesis (e.g., for α=5%) and proceed with the Nemenyi post-hoc tests. In this case, the critical difference between the six ranks for the Nemenyi post-hoc test is 2.85. Thus, these tests show that the classifier J48 obtains a better result than TAN, KNN (except in the third scenario – $4.07-1.92=2.15 \leq 2.85$) and SVM. Neural networks (MLP) also perform better than SVM in all scenarios. The other algorithms, however, do not perform significantly differently from each other.

Table 6. Classification rate obtained by classifiers (when trained using dataset with four attributes) in the Los Angeles dataset in second scenario.

Dataset	**NB**	**TAN**	**KNN**	**MLP**	**J48**	**SVM**
Base200	66.49	69.84	61.02	81.31	**97.35**	34.21
Base400	70.37	64.02	55.56	86.15	**97.08**	54.32
Base1000	55.03	58.64	54.23	57.76	**98.67**	57.31
Base200-400	91.8	65.87	79.28	96.56	**97.79**	33.33
Base200-1000	91.71	80.25	95.59	97.35	**97.88**	44.11
Base400-1000	91.45	63.76	82.36	**98.24**	93.29	51.76
Base200-400-1000	92.42	68.08	96.38	96.38	**97.79**	43.29
Rank (Friedman)	**3.57**	**4.14**	**4.35**	**2.07**	**1.14**	**5.71**

Table 7. Classification rate obtained by classifiers (when trained using dataset with four attributes) in the Los Angeles dataset in third scenario.

Dataset	**NB**	**TAN**	**KNN**	**MLP**	**J48**	**SVM**
Base200	71.57	74.33	67.51	84.75	**97.77**	31.57
Base400	76.01	68.66	62.76	89.12	**96.93**	61.53
Base1000	62.07	65.13	61.38	64.67	**98.69**	53.48
Base200-400	93.1	71.88	82.53	96.86	**98**	31.95
Base200-1000	93.72	83.29	96.47	**98.47**	98	43.29
Base400-1000	92.8	69.35	85.52	**98.47**	93.79	47.5
Base200-400-1000	94.02	72.49	96.7	96.7	**98.31**	42.14
Rank (Friedman)	**3.57**	**4.14**	**4.07**	**1.92**	**1.28**	**6**

6. Conclusions and Future Works

This paper presents a study about the behavior of state-of-art classifiers applied to the identification of the road congestion level. We evaluated six classification algorithms in 3 traffic scenarios: Multilayer Perceptron Neural Network (MLP), K-Nearest Neighbors (KNN), Tree Decision (J48), Support Vector Machine (SVM), Naive Bayes and Tree Augmented Naive Bayes (TAN).

In the experiments, we tried to create traffic situations with superficial information to induce more general classifiers that could adapt to actual traffic situations. The results of the experiments showed that the Naive Bayes classifier can perform significantly better than the other classifiers when there is little information about traffic (average speed of the road and the number of vehicles present in this road). In scenarios with more traffic information (average speed of the road, the number of vehicles present in this road, the time that the vehicle was stationary on the road and the road occupancy rate), the J48 classifier (decision tree) can outperform all other assessed classifiers. However, Naive Bayes has maintained a good performance in these scenarios as well, even though it is considered a simpler classifier.

In practical applications, we must consider that the traffic scenarios are dynamic and the cost of obtaining more information is high (number of messages). Thus, more accurate and faster classifiers in environments with little information should be considered.

Considering the performance presented by the Naive Bayes classifier in the experiments, we intend to apply it in the identification of congestion levels to assist vehicle re-routing methods, in order to avoid new congestion and reduce the caused damage.

Acknowledgment

The authors would like to thank CNPq and FAPEMIG for the financial support.

References

Araujo, G., Queiroz, M., Duarte-Figueiredo, F., Tostes, A. and Loureiro, A. (2014) “Cartim: A proposal toward identification and minimization of vehicular traffic congestion for vanet", In Computers and Communication (ISCC), IEEE Symposium on, p. 1–6.

Bila, C., Sivrikaya, F., Khan, M. A. and Albayrak, S. (2017) "Vehicles of the future: A survey of research on safety issues", IEEE Transactions on Intelligent Transportation Systems 18 (5), p. 1046–1065.

Cirelo, M. C. and Cozman, F. G. (2005) "Aprendizado semi-supervisionado de classificadores bayesianos utilizando testes de independência", EPUSP: São Paulo.

Cunha, F., Villas, L., Boukerche, A., Maia, G., Viana, A., Mini, R. A. and Loureiro, A. A. (2016) "Data communication in vanets: Protocols, applications and challenges", Ad Hoc Networks 44 (Supplement C), p. 90 – 103.

Demšar, J. (2006) "Statistical Comparisons of Classifiers over Multiple Data Sets", Journal of Machine Learning Research, p. 1-30.

Djahel, S., Doolan, R., Muntean, G. M. and Murphy, J. (2015) "A communications-oriented perspective on traffic management systems for smart cities: Challenges and innovative approaches", IEEE Communications Surveys Tutorials 17 (1), p. 125–151.

Friedman, N., Geiger, D. and Goldszmidt, M. (1997) "Bayesian network classifiers", Machine learning. Springer, v. 29, n. 2-3, p. 131–163.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P. and Witten, I. H. (2009) "The WEKA Data Mining Software: An Update", SIGKDD Explorations, vol. 11, p. 10-18.

Meneguette, R. I., Fillho, G. P. R., Bittencourt, L. F., Ueyama, J. and Villas, L. A. (2016) "A solution for detection and control for congested roads using vehicular networks", IEEE Latin America Transactions. IEEE, v. 14, n. 4, p. 1849–1855.

Russel, S. and Norvig, P. (2013) "Artificial Inteligence". Rio de Janeiro: Elsevier.

Souza, A. M., Guidoni, D., Botega, L. C. and Villas, L. A. (2016) "CO-OP: Uma solução para a detecção, classificação e minimização de congestionamentos de veículos utilizando roteamento cooperativo", In XXXIV Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos. Bahia, Brasil.

Van den Haak, W. P., Rothkrantz, L. J. M. and Wiggers, P. (2010) "Modeling traffic information using bayesian networks", Transactions on Transport Sciences, v. 3, n. 3, p. 129–136.

Wolpert, D. H. (1996) "The Lack of A Priori Distinctions Between Learning Algorithms", Neural Computation, v. 8, n. 7, p. 1341-1390.