

Non-cooperative Vehicular Density Prediction in VANETs

Lourdes Portugal-Poma Costa[1], Cesar A. C. Marcondes[2], and Hermes Senger[1(✉)]

[1] Computer Science Department, Federal University of São Carlos – UFSCar, São Carlos, SP 13565-905, Brazil
hermes@ufscar.br

[2] Computer Science Division, Aeronautics Institute of Technology – ITA, São José dos Campos, SP 12228-900, Brazil
cmarcondes@ita.br

Abstract. In large urban spaces, like cities, VANETs are formed by vehicles of highly variable speed and uneven geographic node distribution. Due to the ad-hoc nature of such environments, communication systems must seamlessly adapt to abrupt topology changes to keep the vehicular network organized. Maintain connectivity is hard; a possible naive strategy is based upon expensive on-demand reconnections. Another approach relies on controlled message epidemics. Both of them need to adjust communication behavior under different density situations. Thus, infrastructure-free density estimation methods are becoming popular solutions for this problem. Our paper contributes to this area using a unique density estimation method, independent of beaconing and neighbor discovery (which might generate network congestion), free of cooperative orchestration and based on long-term stability metrics. Our method is validated using vehicular mobility traces, showing outstanding group prediction and stability.

Keywords: VANETs communication · Density estimation · Clustering

1 Introduction

Estimates reveal that the total number of vehicles worldwide reached more than 1.4 billion, including cars, trucks, buses, and others [5]. Some forecasts estimate that this number can reach nearly 2.8 billion vehicles by 2036 [26]. Modern intelligent transport system (ITS) applications will be increasingly necessary to improve safety, efficiency, and comfort in traffic. Last decade, to support novel applications and innovation, the automotive sector promoted dramatic evolution in hardware, software, and sensor technologies incorporated into vehicles [13].

This scenario creates an opportunity for innovation and many new applications and services focusing on road safety, traffic management, smart cities, information services, and entertainment. Besides sensors and computational resources incorporated into vehicles, communication is the basic underlying technology

© Springer Nature Switzerland AG 2021
O. Gervasi et al. (Eds.): ICCSA 2021, LNCS 12952, pp. 201–217, 2021.
https://doi.org/10.1007/978-3-030-86973-1_14

that can enable integration to support new applications and information services. Aiming to support messaging and information propagation among vehicles, information services, and applications in Vehicular Adhoc Networks (VANETS), communication has been studied under different approaches including Vehicle-to-Vehicle (V2V) [25], Vehicle-to-Infrastructure (V2I) [14], Vehicle-to-Everything (V2X) [22] and Vehicle-to-Pedestrian (V2P) [7]. On the other hand, cloud computing and related approaches such as fog computing have also been addressed for the integration of VANETS applications [13].

In VANET, many conditions that affect communication can change over time, such as vehicle location, speed, and time between encounters. A major challenge is the vehicular density, which may be defined as the number of vehicles per measurement unit in a defined area [6]. Vehicular density information is not only important to prevent vehicular traffic congestion, it is also the primer to make VANETs communication possible. For example, message broadcasting must be avoided in dense regions, while it can be useful in low-density regions where scarce communication opportunities can disrupt the spreading of messages.

In order to make vehicles aware of the local density, density estimation algorithms can be used. Those that are infrastructure-free, like V2V based algorithms, are of particular interest in VANETs since they do not depend on network infrastructure (for sensing and gathering of information), have lower cost, and cover great areas. However, infrastructure-free density estimation approaches still present performance challenges to be addressed. Mostly, because the algorithms require information gathered by the neighbor discovery (ND), which could be expensive depending on the vehicular density and how frequently the vehicles update their local knowledge base [21]. During the neighborhood discovery (ND), each vehicle broadcasts short messages with its ID and some useful metadata to the directly connected vehicles. The means such messages are transmitted are the base of these algorithms, and currently, a research topic in MANETs [4].

Because VANETs do not have rigorous power constraints, Global Navigation Satellite Systems (GNSS) can help the ND process create collaborative and synchronized solutions. The gathering of information by cooperative effort allows vehicles to update their local knowledge-base with information obtained by themselves and nearest neighbors. Once that information is processed (e.g., aggregation or summarization), it could be propagated in a recurrent process, allowing vehicles to be aware of the density after more than one-hop of connectivity. But, VANETs are highly dynamic networks, vehicles' knowledge could be outdated or misinformed rapidly. That is because any orchestration in the collaborative process could be interrupted under intermittent connectivity conditions. Thus, vehicles could not update their local knowledge-base and keep outdated information, which could be shared on opportunistic occasions (e.g., store-and-forward), leading the system to make wrong decisions.

Another problem in a collaborative information gathering process is information privacy. Neighboring vehicles share location, speed or even travel routes. So, it is necessary a scheme that considers vehicles' identification, anonymity, trusted authorities (TA), and other privacy issues. Relying on vehicles to execute security tasks could overload VANETs with different responsibilities that are difficult to perform effectively in a VANET environment. Because of that,

VANETs should leverage other technologies, such as Cloud Computing, which is an important component for IoT to succeed. Vehicular using Clouds (VuC) [1] is being recently considered as an extension of VANETs to facilitate the sharing and processing of information [20], relieving VANETs from the active management effort, improving performance, and decreasing privacy risks.

This paper presents a novel density estimation proposal based on Machine Learning (ML) to be supported by Cloud Computing (CC) for decoupling processing and storage from on-board units. Differently from common density measures: based on connectivity (e.g., node degree, two-hops measures [15]), speed properties (e.g., average velocity, speed direction), average fraction of vehicle stop time [19], we used group stability metrics that describe long time properties. Our method allows vehicles to detect changes in density (without on-edge cooperative effort), suggesting if vehicles tend to be alone or in groups in specific places and times. We reason that particular group behavior is related to specific city places, and consequently, it is predictive. We simulate our novel method using real taxi trace replays and epidemic communication patterns to show the method's benefits. We present results that our method was able to find great cluster stability related to each group behavior. Finally, we limit our study and do not approach privacy issues since Physical VuC architecture is out of scope, but we suggest [1] and [18] for more details.

The rest of the paper is organized as follows. In Sect. 2 we present background and literature review on density estimation. Our novel method for density estimation is presented in Sect. 3, while experiments that validate the proposal are presented in Sect. 4. Section 5 has a discussion on results, and our concluding remarks are presented in Sect. 6.

2 Background and Related Work

Vehicular distribution is not uniform in urban scenarios due to the topology of the cities, traffic regulation, time influence, the interaction between vehicles, among other factors. Because of that, the communication systems require to adapt their operation according to the form of communication. Beaconing, geocasting -or multicasting- and unicast routing are basic communication forms that could support VANETs applications. Beaconing is used among all neighboring nodes to announce their presence (e.g., in the ND process) or continuously update information. Geocasting is advertising information inside a target area and is used by many safety applications to send high priority warnings triggered by external events. Finally, unicast routing is the transmission of messages through the ad-hoc network to a specific destination.

Despite these, VANETs communication systems could be further improved when informed about vehicular density. Under high-density conditions, beaconing messages can be lost, erroneous, or higher latency beacon reception. To solve these beaconing problems, one could adapt MAC/PHY layer related parameters (e.g., message frequency, transmission power [11]) according to the region's density. Density estimation aims to reduce beaconing loads under high-density scenarios, increasing reliability of beacon delivery, and reducing application degradation.

There is no guarantee that vehicles within the target area will receive the message in *geocasting* on low-density vehicular scenarios. Under that condition, vehicles could apply flooding-based, static-nodes assistance, message replication, and caching techniques. On the other hand, under high vehicular density, data redundancy and overhead problems are likely to happen. In that condition, farthest vehicle selection, probability-based, timer-based, and priority-based mechanisms could be applied to make communication effective. In terms of routing and connectivity, with high density, *unicast* routing may present transmission collisions and errors. While, under lower density and faster mobility, end-to-end connectivity is more intermittent or even absent. Thus, if routing algorithms were augmented with density information, there is an opportunity to choose between different routing paradigms, such as Store Carry and Forward (SCF) paradigm [24], cluster-based routing [17] or even hybrid routing [16].

2.1 Density Estimation Algorithms

Density algorithms' performance is influenced by two main elements: which information is used to represent density and how it is gathered and processed to recognize a dense region. Throughout literature, most proposals regarding density estimation can be organized depending on the information used. Some use metrics of the transmission medium (e.g., channel busy ratio, bit error rate, interference level, number of received beacons [2]). Moreover, mobility information from vehicles as individual nodes (e.g., velocity, movement direction, acceleration) or from the vehicular traffic as a whole (e.g., traffic flow, traffic stream speed, road density) can also be used. Ultimately, some proposals use graph-based connectivity information (e.g., node degree, link duration, and cluster coefficient).

Nevertheless, these parameters' measurements change rapidly in VANETs, so it is important to update these parameters frequently or use predictive algorithms to estimate their future values. That is the reason why an adequate infrastructure for gathering and process that information is important. Thus, if we focus our attention on how density algorithms gather and process the information, infrastructure-free density estimation approaches are preferred because of the lower cost and greater coverage area. According to [6], infrastructure-free density estimation algorithms are classified into three types. The first category includes the algorithms that use *statistical modeling*, which could be applied when specific requirements are fulfilled (e.g., exponential vehicles inter-arrival times [10]). However, such algorithms are not suitable for complex urban VANETs (whose vehicles follow many mobility patterns) and their performance depends on the sampling process, being difficult to apply in low-density regions with a low amount of collected samples. The second category includes the algorithms that use VANETs and traffic flow information for density estimation. They employ V2V (vehicle-to-vehicle) communications [21] and also could work collaboratively with RSUs (roadside nits). In addition, these RSUs can support the collection and distribution of information and provide access to external infrastructure with data aggregation and processing services.

Finally, a third category considered in [6] includes the *group or cluster-based density estimation algorithms*. The detection of clusters of vehicles is a current research topic not only in density estimation but also in vehicular communication, to improve performance using load balance and message aggregation. In density estimation and once the clusters are detected, the members entrust the density estimation to a specific node (Cluster Head - CH), which will aggregate and process the gathered information inside the cluster and propagate the density results to the members. In this category, the main challenge is to form and maintain the clusters, based on the connectivity stability which is determined by the similarity of the members' behavior (e.g. pairwise link duration, movement direction, and velocities ratio [9]). Thus, the processes of cluster formation, its maintenance, the allocation of the CH and member roles, have an additional cost due to the transmission of control messages and metadata with information that could even outdated, because of the VANET dynamic behavior.

2.2 The Role of Cloud Computing

Given that vehicular density recognition is challenging to be performed in a cooperative and pure VANET manner, the gathering and processing of information could be supported by services on the cloud. Vehicles could access Cloud Computing services through RSUs or other network devices and inter-networking technologies. Cloud Computing in VANETs has two major research lines, Vehicular cloud computing (VCC) and Vehicle using Cloud (VuC). VCC includes VANETs as an extension of the Cloud services (e.g., for data storage or VANETs as networking service), VuC is the efficient use of cloud resources to support VANETs. Vehicles using Clouds (VuC) can enable applications that are unlikely to be developed in a pure VANETs, such as smart ITS applications including real-time traffic and predictive capabilities. Thus, we can consider VANETs as part of a perception layer [1] that gather density related parameters and take them to the cloud, without on-edge data sharing and using more advanced and self-controlled resources on the cloud. Also, as the work in [20] propose, we can make use of Cloud resources for the sharing of sensitive information (e.g., vehicles trajectories), which will not only be protected by on-edge privacy schemes (e.g., RSUs recognized by a trusted authority) but also by additional elements on the cloud that ensure that this information will not be compromised.

3 Non-cooperative Density Category Prediction

In this section, we propose a novel method based on machine learning (ML) and Vehicles using the Cloud (VuC) to predict when and where vehicles tend to form groups and how stables are those. That information alone can support different communication forms (i.e., beaconing, geocasting, unicasting) to improve their performance.

Our method is based on a VuC architecture, as shown in Fig. 1. The lowest/first layer, or the perception layer, includes all devices involved in gathering

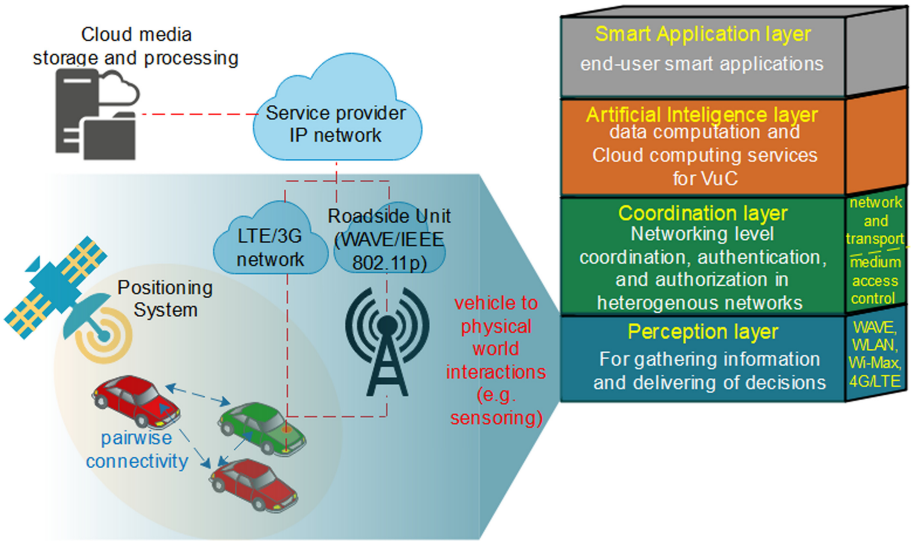


Fig. 1. Architecture for our method (based on the layered VuC structure in [1])

information such as in-vehicle sensors, GPS receivers, and RSU collecting traffic and environmental data. The second layer represents all the coordination for the transmission and hand-off of heterogeneous networks that support VANETs. However, as we modify how VANETs transmission is performed, second layer's devices would act as *networking actuators*. Hence, networking devices that modify the beaconing, geocaching, or unicasting normal behavior according to the vehicular density by using a produced model in upper/third layer (or Intelligence layer). The third layer is where the entire machine learning is executed to recognize groups of vehicles. It includes conventional Cloud Computing devices for storage and processing (e.g., location and temporal information). Finally, the highest layer, or Smart Application layer, includes end-user smart applications such as safety, efficient-driven, and infotainment, supported by beaconing, geocasting, or unicasting communications.

Our proposal consists of using a traditional VuC architecture, where the actuators will be located on the second layer or Coordination layer, which will use density inference rules produced by a trained model to modify beaconing, geocaching, or unicasting behavior. We assume that vehicles can transmit GPS coordinates and temporal data to the cloud through a car-to-cloud communication system. Once the data is stored on the cloud, and with the Cloud resources in the artificial intelligence layer, we can apply our method in three main key steps: *i*) the detection of moving groups and isolated nodes, *ii*) the clustering of groups of vehicles, which establish density categories using hierarchical clustering, and *iii*) training and prediction of group density conditions, using a model based on ensemble learning of the Random Forest algorithm.

3.1 Step 1 - Moving Groups and Isolated Nodes Detection

In the first step, we use the positioning and temporal information, stored in the cloud, to identify vehicle groups that existed before, at different places and times. Once the groups are identified, we detect moving groups using snapshots information. At first, we interpolated the reported temporal and GPS data, so we can reconstruct vehicles' trajectories and produce snapshots of their locations at a fixed interval of time. Thus, beginning at the moment of time i and interpolating at a time interval of h seconds, we obtain a set of n snapshots $S = \{S_i, S_{i+h}, S_{i+2h}, \cdots, S_{i+nh}\}$, where each snapshot S_j is the set of locations and time information tuples $((time, coordinate_x, coordinate_y))$ of the m vehicles that are transiting at the moment of time j, that is

$$S_j = \{(time_j, v1_x, v1_y), (time_j, v2_x, v2_y), \cdots, (time_j, vm_x, vm_y)\} \quad (1)$$

Once vehicles' trajectories are timestamped, we detect moving groups of vehicles doing the following: *i*) the detection of isolated vehicles and groups in every single snapshot, and *ii*) recognition of moving groups of vehicles by observing consecutive snapshots.

Detection of Isolated Vehicles and Groups: Our method can process the locations of a specific snapshot S_i to detect groups of vehicles by using density-based clustering algorithms. These algorithms "classify" clusters as dense areas in the data space, where the observations are close to each other and separate from regions of low density. To detect groups of connected vehicles (that sustain pairwise connectivity) and isolated vehicles, we employ a simplification of the DBSCAN algorithm (Density-based spatial clustering with noise)[1]. Our simplified DBSCAN is based on two main parameters: the *neighborhood* and a *minimum number of observations* inside the neighborhood, that is necessary to consider the neighborhood as dense (or as a group) ($MinPts$). We can express the neighborhood of a vehicle like this:

Definition 1. *The ξ-Neighborhood of a vehicle v ($N_\xi(v)$) is the subset of vehicles inside its neighborhood of radius ξ, from the set D of vehicles in the scenario:*

$$N_\xi(v) = \{q \in D | distance(v, q) <= \xi\} \quad (2)$$

Once defined the neighborhood, the next definition is related to reachability:

Definition 2. *A vehicle v is Directly density reachable from another vehicle q, with respect to ξ and $MinPts$, if:*

- $v \in N_\xi(q)$ *and,*
- $|N_\xi(q)| \geqslant MinPts$.

[1] Because in pairwise connectivity two is the minimum number of points to form a group, so there will be no difference between core and border nodes (see [8]).

Considering the *directly density reachable* definition, two vehicles p and q are *density-connected* if there is a vehicle o such that both p and q are directly density reachable from o. Thus, all density-connected vehicles are members of the cluster, and any other connected to them will also be part of it. Hence, as we can see, vehicle group connectivity can be captured using DBSCAN if we set the parameters $MinPts$ to two (for representing pairwise connectivity) and ξ to an approximate value of the effective transmission rate in V2V communication.

Detection of Moving Group of Vehicles: A moving cluster is a set of entities that move together and closely for some time. The entities do not need to remain together, as nodes can join or leave the group during their lifetime. For instance, a movement cluster is a convoy of vehicles, whose movement is restricted by the roads and traffic signs in a city. Having identified all groups of vehicles in each snapshot S_i using DBSCAN, we can consider a threshold of the minimum percentage of vehicles that should remain in a group to be considered as a moving cluster. Thus we can define a moving cluster formally as follows:

Definition 3. *Let $S = \{S_1, S_2, \cdots, S_n\}$ be a sequence of snapshots, and let $C = \{C_1, C_2, \cdots, C_n\}$ be a set of C_is, where each C_i is a set of clusters found at snapshot S_i. Then, m is a moving cluster with respect to a threshold θ, where $0 < \theta \leq 1$, if:*

$$\frac{|c_i \cap c_{i+1}|}{|c_i \cup c_{i+1}|} > \theta \text{ , where } c_i \in C_i \text{ and } c_{i+1} \in C_{i+1} \tag{3}$$

Finally, we obtain as the result of this step the moving groups of vehicles already identified, the list of their members, and the isolated vehicles (not members of any DBSCAN cluster) with temporal and geographic information.

3.2 Step 2 - Clustering of Groups of Vehicles

Given the moving groups of vehicles, already identified in the previous step, our objective is to describe their stability and use ML clustering techniques to cluster those groups into similar stability categories. Thus, this step is performed in two parts: *i*) describing vehicle group behaviors, using a good set of features, and *ii*) clustering vehicle groups' behaviors. Also, from now on, and to avoid misunderstandings, we use the term "group" to refer to each group of vehicles found in the previous step, and "cluster" refers to each cluster (in the machine learning function) composed of groups of vehicles with similar stability.

Afterwards, to describe the stability of vehicles' groups, we use *long-time cluster stability* measurements, calculated to represent the survival of vehicles as a group, and as members of any group in consecutive snapshots. These measurements are: *cluster lifetime*, which is the time that a group of vehicles existed according to Eq. 3, the *average membership time*, and the *average number of members* that a vehicle group had during its existence. We only use information about the groups of vehicles that have disappeared (dead groups) to include

complete information. Hierarchical clustering with complete linkage is used in the clustering of the vehicle groups, based on their stability metrics. As a result, we obtained clusters of vehicle groups with similar stability behavior.

3.3 Step 3 - Training and Prediction of Group Density Condition

The last part of our method is based on the following insight: *Group motion and their properties are closely related to specific regions and times of the day.* Thus, only geographic and temporal information can be enough to predict the categories of groups of vehicles found by hierarchical clustering.

To test our hypothesis and allow vehicles to recognize how the group behavior is in a specific place and time, we could label each cluster of the hierarchical clustering result and train an ML classifier to predict these categories in the future. We use Random Forest classifier, with the following attributes: (*i*) the number of the day (from 1 to 5, meaning Monday to Friday), (*ii*) time of the day in seconds, (*iii*) x-coordinate, (*iv*) y-coordinate, and (*v*) the cluster label.

The cluster label is our target feature, which would indicate a particular cluster behavior. In Table 1, we show an example of the dataset used for training. The dataset also includes observations that belong to isolated vehicles (i.e., identified as noise in Step-1) in moments where vehicles cannot form groups. Isolated vehicles are labeled with "−1". As we can observe, the stability metric used for clustering were excluded from the classification dataset. While, the cluster label represents the vehicle groups' behavior based on these metrics, and it is the target feature of the classifier predictions.

Table 1. Entries of hierarchical clustering and classification observations.

Samples of observations used for the clustering				
Time of life	Avg. membership time		Avg. Num. of members	
840	435		2.1	
1860	1470		2.4	
Samples of observations used for classification (class label: cluster ID)				
Day	Time of day	x-coordinate	y-coordinate	Cluster ID
5	32490	292275.2	4641412	−1
2	14940	292329.1	4642126	1

4 Case Study and Experimental Validation

This section explains how our method could be applied using a simulated scenario based on real traces of taxis from the city of Rome [3]. The simulation performed stands as a method validation approach since it augments other proposals. Hence it is not comparable directly to other proposals of message delivery improvement.

Then, the Rome traces are used as input into the Opportunistic Networking Environment (ONE) simulator [12] which is designed for evaluating DTN routing and application protocols. We simulate a VANET transmitting messages using the naive Epidemic algorithm. Epidemic [23] performs routing under the Store-Carry-Forward paradigm and message replication. Once a message is created in the epidemic, each vehicle transmits a copy of each neighbor's message in opportunistic contacts. The general idea is that some vehicle finds the destination and delivers a copy to it. As in other VANETs routing algorithms, Epidemic routing performance is affected by vehicular density, so it is expected that our density predictions will be related to the Epidemic performance.

Initially, we interpolate the movement traces to obtain snapshots of the locations of the vehicles. Next, we applied our method to produce a Random Forest (RF) model that will be executed throughout the simulation. To query the RF trained classifier, we create an interface between the ONE simulator (made in Java) and R. It allows us to execute scripts with the necessary programming code to receive the vehicles' geographic and temporal information in the simulator. Thus, each vehicle queries the classifier every 15 s and starts from its first movement. As a result, the classifier returns a predicted value, indicating the region's group density category where the vehicle is located.

4.1 VANETs Scenario and Communication

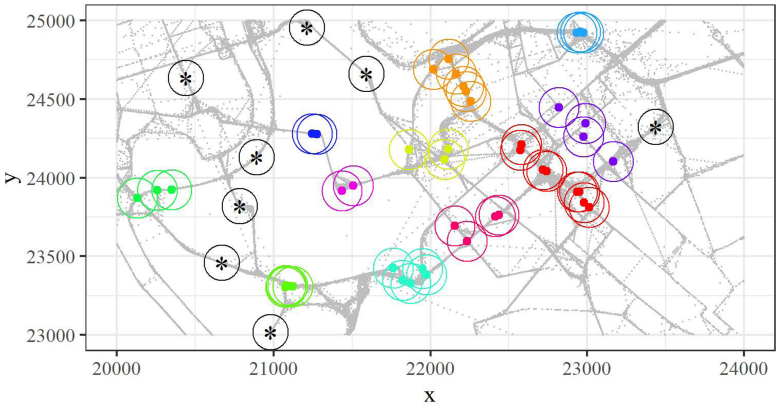


Fig. 2. Groups of vehicles reconigzed by DBSCAN (isolated vehicles in black).

We cleanse the data by pre-processing the Rome traces to remove GPS misreadings, exclude incomplete short trajectories, and interpolate and complete partitioned trajectories applying a map-based approach. The number of active taxis in our dataset depends on the time of the day. The number of taxis from Monday to Friday is plotted in Fig. 3. We observe three peaks of activity: one

in the early hours, another in the morning, and the last at night. Our final dataset includes movement traces of twelve hours, from 7 a.m. to 7 p.m. Then, we simulate a VANET transmitting messages using the Epidemic algorithm.

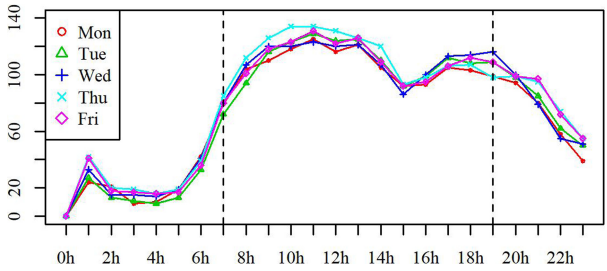


Fig. 3. Number of vehicles from Monday to Friday in different times of the day.

Regarding which vehicles were considered as source and destination of messages, we aim to select (randomly) between vehicles with long times of activity (according to the traces dataset). New messages are generated only from 7 a.m. to 2 p.m. Afterwards, throughout simulation, only copies of the messages created initially will be transmitted. Also, Epidemic flooding of copies is not restricted as we show in Table 2 where we also include other simulation parameters.

Table 2. Simulation parameters.

Transmission rate	5 Mpbs	Transmission range	250 m
Storage capacity	Unlimited	Message size	128 KB
Message creation	$\approx$ each 60 s	Number of nodes	Variable

4.2 Settings and Method Application

We set the DBSCAN parameters ξ and $MinPts$ of 200 m and 2 units, respectively. Therefore, it allows us to identify vehicles connected inside a transmission range of 200 m as a group. In Fig. 2 we show, as an example, a region of the city of Rome with isolated and group of vehicles identified by DBSCAN in a single snapshot. To detect moving clusters, we set the θ value to 0.5 so that the required quantity of common vehicles (in two consecutive snapshots) be at least half of all the vehicles in the snapshots. After the moving clusters have been identified, we measure how stable are those using the metrics of *cluster lifetime*, *average membership time*, and the *average number of members*. Subsequently, on the obtained dataset, we proceed to perform hierarchical clustering and, we finish the pipeline by including a cluster identifier (or category), which will be the target variable of the RF classifier.

Depending on the number of categories that we obtained in the clustering of vehicle groups, it will be more difficult to distinguish which cluster (or category) has certain properties and how stable they are. For that reason, we choose to cut the hierarchical structure into three clusters, which are three categories or stability behaviors influenced by the vehicular density, as we report in the discussion section. Besides, the greater the number of clusters, the lower the quantity of labeled (or annotated) data for each class to train a classifier with good performance. Thus, as we show in Fig. 4, both the OOB (Out-of-bag) error in the training set and the classification error in the test increase (slowly) as the number of chosen clusters increase. In all cases, the error is less than 16%.

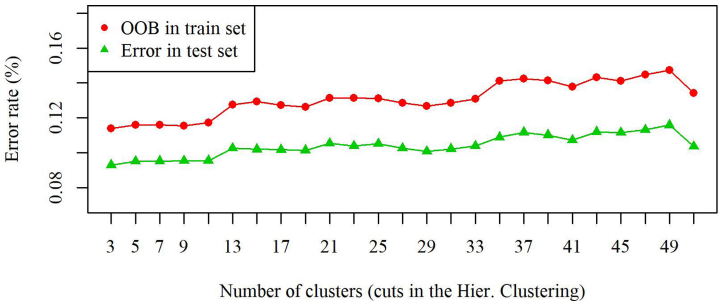


Fig. 4. Error rate in training and testing sets per number of categories for Random Forest with ten trees. The OOB is the "Out Of the Bag" error in the train set.

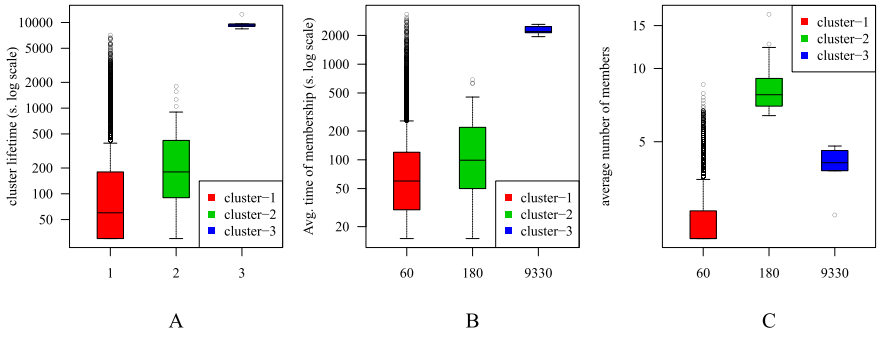


Fig. 5. Clusters of vehicles. (A) Clusters' lifetime. (B) Time of membership per cluster.(C) Average number of members in each cluster type, versus average clusters lifetime.

4.3 Simulation Reports and Evaluation

In our VANET simulations using traces from Monday to Friday, vehicles save all the predicted group density category values, connectivity measurements, and

message transmission information for evaluation purposes. Those values are reported at the simulation end, with the proper timestamp. The purpose is to discover if the predictions have a logical relation with the connectivity properties and the message transmission performance. The reported measures are collected by each vehicle, and the values are obtained every time vehicles perceive (through the classifier) that the group density category has changed. After reporting its measurements, each vehicle resets its variables to perform new measurements for the new group density category detected. In Table 3 showed below, on the left side of the table, the connectivity and message transmission measurements gathered during the simulation. While, on the right side, we include what we want to represent or know with each of the measurements.

Table 3. Measurements from simulation

Variable from simulation	Aim of representing
Same category prediction interval time ($TinC_i$)	Membership time
Average node degree during ($TinC_i$)	Avg. number of members
Avg. connection duration during ($TinC_i$)	Membership time
Number of received messages during ($TinC_i$)	Transmission performance
Number of initiated connections during ($TinC_i$)	Overhead existence or absence

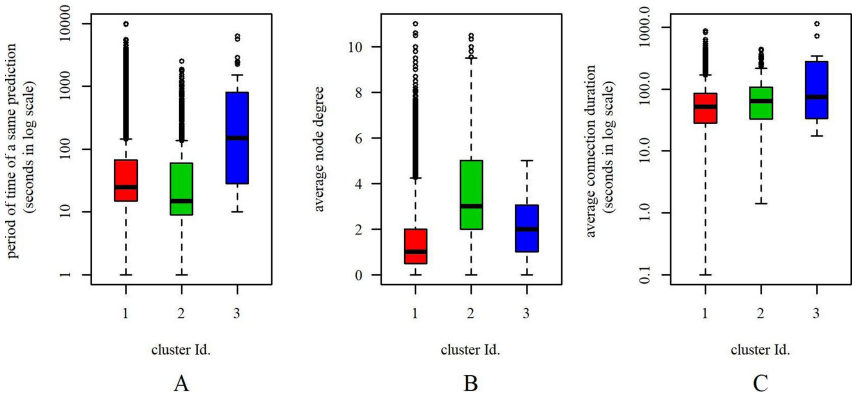


Fig. 6. The classifier's predictions. (A) Duration of a condition group per classifier. (B) Average number of neighbors during time that a vehicle remains in group condition. (C) Average times connections keep stable, when vehicle was under group condition.

5 Analysis of Results

Before showing our validation results, notice that group density conditions were categorized according to group stability metrics. That information was not used

to train our classifier. In this way, it is important to see how coherent the connectivity and message transmission measurements are from the simulation itself, using the classifier's predictions. In Fig. 6-A, we show the prediction interval time variability ($TinC_i$) within same category. That is, the period of time in which the classifier did not return a different group density (within the $cluster - ID$-C_i-). As expected, $TinC_i$ is related to the membership time attribute used for hierarchical clustering in Step-2. Thus, it was designed to work like that. Since a vehicle receives the same $cluster - ID$ from the classifier, it is probably that the vehicle is part of that group's underlying connectivity properties. Hence the classifier assumes that it is of category C_i. The clearest relationship here happens between the prediction of C_i and the categorization, like the case of category number 3. The category with $cluster - ID$ 3 is, in fact, the most stable density group category; thus, as expected, the membership time is higher. The result highlights the outstanding performance of group stability of our method. In Fig. 6-B, we get other accurate results. If we consider that the average node degree of a vehicle is related to the average number of members that a group has (see Fig. 5-C). Thus, if a vehicle's classifier predicts a density group category C_i, its average number of members should be related to the average number of neighbors that the vehicle has during $TinC_i$. Hence, as we can observe in Fig. 6-B, cluster-2 values are higher than those of cluster-3, which in turn, are higher than the cluster-1 values. Therefore, even though cluster-3 is more stable among all (see Fig. 6-A), clearly cluster-2 was detected in places of more congestion. Finally, in Fig. 6-C, we plot average connection duration between pairs of vehicles. It is an important indicator of cluster stability because vehicles could remain or leave from the group of vehicles and, as soon as the expression in Eq. 3 is fulfilled, and a moving cluster will be considered alive. Thus, as shown in Fig. 6-C, cluster-3 is the most stable considering also the pairwise connectivity between members. On the other hand, although values between clusters 1 and 2 are close, if we consider the minimum values, we can see that cluster-2 is more stable than the group of vehicles of cluster-1 and more congested from Fig. 6-B.

Figure 7 shows overall connectivity and message transmission measurements under each type of group density category predicted by the classifier. Hence, we can verify, in Fig. 7-A that, even though within cluster-3 the number of vehicles is less than in cluster-2, the number of received messages is far greater. The reason is related to the fact that vehicles in groups of type 3 have longer periods of connectivity. Thus, it is sufficient that one vehicle of that pack to receive a message to share it with the group. Moreover, Fig. 7-B shows the number of initiated connections of vehicles under each group density category. For this particular result, We also include the measurements from vehicles whose classifier predicted as isolated vehicles or in a "non-group category. The horizontal labels mean that one vehicle's average time remains in a specific condition (average $TinC_i$ determines it is isolated or in a group of type 1, 2, or 3). We can verify that most observations were classified when vehicles required it. The most classification errors happen between the "non-group" and cluster-1 categories. That is because groups of cluster-1 include very unstable vehicle groups. Besides, due

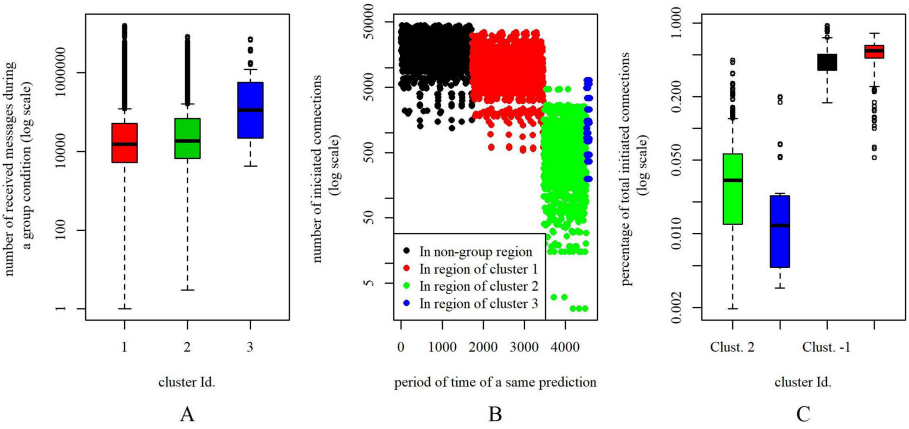


Fig. 7. Connectivity and transmission behavior for each group. (A) Messages received when vehicles remain in the same condition. (B) Average number of neighbors of each vehicle. (C) Percentage of connections that a vehicle establishes under a group.

to the nature of our 30 s snapshots interval, if vehicles' group did not survive such a lower period of time, they were automatically considered as isolate nodes.

Finally, we can observe that more unstable types of clusters established a higher number of connections. The cause of this instability can be interpreted as intermittent connections. These tend to break and establish again among the same vehicles. On the other hand, regarding the more stable category (cluster-3), the classifier can recognize it, despite using an unbalanced dataset for classification. Moreover, Fig. 7-C shows the percentage representing the number of connections under each group density category from the total number of connections established during the simulation. This metric reveals that most of the connections established by vehicles are under unstable conditions. Such finding is consistent with our taxi scenario of Rome.

6 Conclusions and Future Work

Density estimation algorithms could support communication systems to adapt under different connectivity situations. However, collective connectivity awareness is costly and might be inaccurate after a short time, thus density estimation algorithms independent of V2V communication are needed. Therefore, we propose a method that can categorize vehicular density according to long-time stability metrics related to connectivity properties and message transmission performance. This improvement is achieved by using the vehicular location information and identifying a group of vehicles through density-based clustering. Afterward, we categorize the identified groups using group stability measures and hierarchical clustering. We demonstrated using extensive simulations based on real traces, that cluster stability measures capture relevant information related

to group behavior. Furthermore, by analysing the simulation results that characteristic of specific places and times can be predictive. As a result, our trained classifier obtained excellent performance, independently of the number of types of groups to predict. We intend to provide our method a feedback loop to update their classifier with real-time information for future work, and also embed our method in message delivery algorithms as an augmentation.

Acknowledgment. Authors thank Coord. de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001. H.S. also thanks Stic-AMSUD (project 20-STIC-09), and FAPESP (contracts 2018/22979-2 and 2015/24461-2) for their support.

References

1. Aliyu, A., et al.: Cloud computing in VANETs: architecture, taxonomy, and challenges. IETE Tech. Rev. **35**(5), 523–547 (2018)
2. Barrachina, J., et al.: I-VDE: a novel approach to estimate vehicular density by using vehicular networks. In: Cichoń, J., Gębala, M., Klonowski, M. (eds.) ADHOC-NOW 2013. LNCS, vol. 7960, pp. 63–74. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-39247-4_6
3. Bracciale, L., Bonola, M., Loreti, P., Bianchi, G., Amici, R., Rabuffi, A.: CRAWDAD dataset roma/taxi (v. 2014–07-17), July 2014. https://crawdad.org/roma/taxi/20140717
4. Chen, L., Bian, K.: Neighbor discovery in mobile sensing applications: a comprehensive survey. Ad Hoc Netw. **48**, 38–52 (2016)
5. Chesterton, A.: How many cars are there in the world? (2018). https://www.carsguide.com.au/car-advice/how-many-cars-are-there-in-the-world-70629
6. Darwish, T., Bakar, K.A.: Traffic density estimation in vehicular ad hoc networks: a review. Ad Hoc Netw. **24**, 337–351 (2015)
7. Doone, M.G., Cotton, S.L., Matolak, D.W., Oestges, C., Heaney, S.F., Scanlon, W.G.: Pedestrian-to-vehicle communications in an urban environment: channel measurements and modeling. IEEE Trans. Antennas Propag. **67**(3), 1790–1803 (2018)
8. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: International Conference on Knowledge Discovery and Data Mining, KDD 1996, pp. 226–231 (1996)
9. Grzybek, A., Seredynski, M., Danoy, G., Bouvry, P.: Detection of stable mobile communities in vehicular ad hoc networks. In: IEEE International Conference on Intelligent Transportation Systems (ITSC), pp. 1172–1178 (2014)
10. He, J., Cai, L., Pan, J., Cheng, P.: Delay analysis and routing for two-dimensional VANETs using carry-and-forward mechanism. IEEE Trans. Mobile Comput. **16**(7), 1830–1841 (2017)
11. Jabbarpour, M.R., Noor, R.M., Khokhar, R.H., Ke, C.H.: Cross-layer congestion control model for urban vehicular environments. J. Netw. Comput. Appl. **44**, 1–16 (2014)
12. Keränen, A., Kärkkäinen, T., Ott, J.: Simulating mobility and DTNs with the one. J. Commun. **5**, 92–105 (2010)
13. Kim, T., Min, H., Choi, E., Jung, J.: Optimal job partitioning and allocation for vehicular cloud computing. Future Gener. Comput. Syst. **108**, 82–96 (2020)

14. Kimura, T., Saito, H., Honda, H.: Optimal transmission range for v2i communications on congested highways. In: IEEE Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC), pp. 1–7 (2017)
15. Korn, A., Schubert, A., Telcs, A.: Lobby index in networks. Phys. A **388**(11), 2221–2226 (2009)
16. Lakkakorpi, J., Pitkänen, M., Ott, J.: Adaptive routing in mobile opportunistic networks. In: 13th ACM International Conference on Modeling, Analysis, and Simulation of Wireless and Mobile Systems, MSWIM 2010, pp. 101–109. ACM (2010)
17. Liu, X.: A survey on clustering routing protocols in wireless sensor networks. Sensors **12**(8), 11113–11153 (2012)
18. Pan, J., Cui, J., Wei, L., Xu, Y., Zhong, H.: Secure data sharing scheme for VANETs based on edge computing. EURASIP J. Wireless Comm. Netw. **2019** (2019)
19. Panichpapiboon, S., Pattara-atikom, W.: Exploiting wireless communication in vehicle density estimation. IEEE Trans. Veh. Technol. **60**(6), 2742–2751 (2011)
20. Park, Y., Sur, C., Noh, S.W., Rhee, K.H.: Self-controllable secure location sharing for trajectory-based message delivery on cloud-assisted VANETs. Sensors **18**(7), 2112 (2018)
21. Sun, Y., Luo, S., Dai, Q., Ji, Y.: An adaptive routing protocol based on QoS and vehicular density in urban VANETs. Int. J. Distrib. Sensor Netw. **2015**, 5:5 (2015)
22. Toghi, B., Saifuddin, M., Mughal, M., Fallah, Y.P.: Spatio-temporal dynamics of cellular v2x communication in dense vehicular networks. In: IEEE Connected and Automated Vehicles Symposium (CAVS), pp. 1–5. IEEE (2019)
23. Vahdat, A., Becker, D.: Epidemic routing for partially-connected ad hoc networks. Technical report, Duke University (2000)
24. Vasilakos, A.V., Zhang, Y., Spyropoulos, T.: Delay Tolerant Networks: Protocols and Applications. Wireless Networks and Mobile Communications. CRC Press, Hoboken (2011)
25. Wang, J., Peeta, S., Lu, L., Li, T.: Multiclass information flow propagation control under vehicle-to-vehicle communication environments. Transp. Res. Part B Methodol. **129**, 96–121 (2019)
26. Zeadally, S., Guerrero, J., Contreras, J.: A tutorial survey on vehicle-to-vehicle communications. Telecommun. Syst. **73**(3), 469–489 (2019). https://doi.org/10.1007/s11235-019-00639-8