



**HAL**  
open science

# Selection of relays based on the classification of mobility-type and localized network metrics in the Internet of Vehicles

Philippe Fabian, Abderrezak Rachedi, Cédric Guéguen

► **To cite this version:**

Philippe Fabian, Abderrezak Rachedi, Cédric Guéguen. Selection of relays based on the classification of mobility-type and localized network metrics in the Internet of Vehicles. Transactions on emerging telecommunications technologies, 2021, 32 (4), 10.1002/ett.4246 . hal-03190430

**HAL Id: hal-03190430**

**<https://hal.science/hal-03190430>**

Submitted on 2 Dec 2021

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Selection of relays based on the classification of mobility-type and localized network metrics in the Internet of Vehicles

Philippe Fabian\*<sup>1</sup> | Abderrezak Rachedi<sup>1</sup> | Cédric Guéguen<sup>2</sup>

<sup>1</sup>Gaspard Monge Computer Science  
Laboratory (UMR 8049), University  
Gustave-Eiffel, France

<sup>2</sup>IRISA (UMR 6074), University of Rennes  
1, France

## Correspondence

\*Philippe Fabian. Email:  
philippe.fabian@univ-eiffel.fr

## Present Address

Cité Descartes - 5, boulevard Descartes,  
F-77454 Marne-la-Vallée

## Summary

The ever increasing amount of connected mobile devices and users in the Internet of Vehicles (IoV) and the fact bicycles, electric scooters and users' smartphones are also connected poses a real challenge in terms of ensuring Quality of Service (QoS). The constantly changing topology due to the high-mobility of devices and users greatly impacts its stability and the connectivity of devices. Furthermore, to ensure safety of people in the case of autonomous vehicles it is of paramount importance to ensure excellent reliability. This is why we have developed a solution that is based on Machine Learning (ML) to classify devices according to their mobility profile and that uses a scoring system to select the best candidates to act as Mobile Relays (MR) amongst devices with a suitable mobility profile. The scoring system allows to find critical locations in terms of user density. This solution does not require a dedicated infrastructure such as Road Side Units (RSUs). Simulations results will show the proposed solution increases the packet delivery ratio by up to 6%, reduces the energy consumption by up to 30% and increases the efficiency of bandwidth usage without sacrificing the end delay of users and devices compared to the state of the art.

## KEYWORDS:

Internet of Vehicles, Machine Learning, Selection of Relay, Vehicle-to-Vehicle

## 1 | INTRODUCTION

Mobile connected devices are more and more present<sup>1 2 3</sup> : pedestrians with mobile phones, motorized vehicles equipped with an on board computer or public connected bicycles and scooters. All these devices form the Internet of Vehicles (IoV). They can potentially connect to one another or they can directly connect to Internet using for instance either a 4/5G connection to a cellular antenna or a WiFi connection through an Access Point (AP). However, in such a dynamic network of mobile devices it is challenging to ensure topology stability and thus Quality of Service (QoS). Indeed, the topology is constantly changing because users and devices travel in different directions at different velocities, they connect and disconnect from the network according to their needs or to save battery life, or because objects such as vehicles cause multipath fading, resulting in some devices gaining or losing connectivity. Furthermore, users and devices have different needs including high throughput for a multimedia stream on a smartphone, minimum delay for autonomous vehicles to avoid a collision or low energy consumption to ensure enough remaining residual energy on a device until the next recharge.

It is not always possible for a device to have a direct connection to a cellular antenna because one might not be available from a given Internet Service Provider (ISP) in low populated areas (deployment costs too high), the user has no subscription plan (too expensive or because the user is traveling abroad) or because the radio conditions are bad (path loss, shadowing, multipath

fading). It is possible to resolve all those issues by using relays to forward data from and to an end user/device. Several solutions propose the deployment of static relays (dedicated architecture such as Road Side Units (RSU)), Mobile Relays (MR)<sup>45</sup> or the usage of parked vehicles<sup>6</sup>. Static relays have limitations such as deployment cost, lack of flexibility and mandatory regular maintenance of the equipment to ensure reliability. On the other hand, MRs allow for more flexibility and can lower deployment costs. In our vision, a dynamic solution using regular devices as relays is the next step compared to dedicated MRs as it allows to implicitly adapt the number of relays and select new ones whenever radio conditions or the needs of users and devices evolve and has the advantage of no maintenance costs. This solution also offers great flexibility by removing the need for a dedicated infrastructure. It allows devices without cellular access to connect to Internet through others. Local communications need not to be forwarded to cellular antennae, which can potentially reduce overhead. Furthermore, this solution can be deployed on top of an existing routing protocol<sup>789</sup> for even better QoS.

This is why we propose in this paper a solution that dynamically selects relays amongst devices present in the area of interest based on a Machine Learning (ML) classification algorithm. The proposed solution is an extension of our previous work<sup>10</sup> where the selection of a dynamic relay is not only based on the classification of mobility but also on localized metrics such as the proposed Expected Packet Count (EPX) and average delay of the different locations. This makes the new solution even better because taking into account localized network metrics allows to select relays in the most critical locations in order to better service users and devices. Furthermore, compared to our previous work, we expanded the types of users in our simulations by adding pedestrians. The highlights of this contribution are :

- We extended the classification algorithm compared to our previous work<sup>10</sup> by adding localized network metrics computed after the classification algorithm has been run in order to take into account the most critical locations in real-time when selecting relays;
- We created a new localized metric, the EPX which represents the packet density, in order to find critical areas where the density of devices is highest;
- The performance evaluation has been further improved by adding pedestrians (along with bicycles and cars) in scenarios considering a light and high density of users, allowing us to test the solution in a very realistic way.

The remaining of this paper is organized as follows : the state of the art is presented in section 2, the proposed solution is explained in section 3, section 4 details the performance evaluation and, finally, the paper is concluded in section 5.

## 2 | RELATED WORK

Existing solutions presented in this section are regrouped into three categories : general, ML based and Cluster Head (CH) based. General approaches (subsection 2.1) concern the selection of relays based on a mathematical model. They are based on Markov chains or cost functions. ML based approaches (subsection 2.2) use machine learning to select relays. Finally, CH based solutions work by electing one relay for each group (or cluster) of users, hence the appellation “cluster head” (subsection 2.3).

### 2.1 | General approaches

Authors in<sup>11</sup> propose a solution to select mobile relays amongst vehicles to allow other vehicles to connect to Internet through them. Neighboring vehicles are connected through WiFi and relays connect to Internet using LTE. Several metrics are used to select relays such as speed, Received Signal Strength (RSS), stability of links and distance between vehicles. Though this solution is good, it does not consider several mobility types and simulations are done in straight lines on highways. In<sup>12</sup>, the authors use Matching Game to predict the radio conditions of Unmanned Aerial Vehicles (UAVs). Once UAVs have determined their own needs, the proposed solution dynamically adapts the mode selection, the time scheduling and the channel allocation to improve performance on the network. However, the authors of this paper studied cases where the number of UAVs is no more than a few tens and only a subset of drones generate data packets. This solution is thus not adapted to a road network where several hundreds of users can generate data at the same time. The solution proposed in<sup>13</sup> uses a relay selection mechanism to help nodes that suffer from high interference. If the radio conditions of a node are too bad, it will scan its surroundings to find suitable neighbors to relay its data. The best neighbor is found using a M/M/1/K first-come-first-serve scheme. This solution allows any node to potentially serve as a relay but it is studied in a context without mobility, which does not make it adapted

to the IoV. The authors in<sup>14</sup> propose to add MRs to help fixed relays during heavy traffic periods. Markov chains are used to predict when heavy traffic periods will happen. With this information, the optimal paths for MRs are computed and they are sent on those paths to assist fixed relays. Although this solution considers mobility of users, MRs are considered to move at a constant speed and end users can only connect to a relay (fixed or mobile). This limits the flexibility of the proposed solution as any user must be in range of a relay and there is a required dedicated architecture (fixed relays).

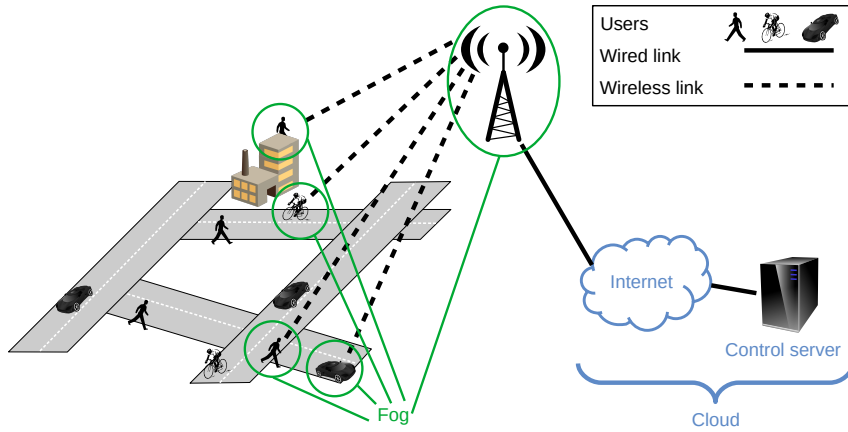
## 2.2 | Machine Learning based solutions

Mchergui et al.<sup>15</sup> propose the use of deep learning to select relays in order to increase performance. The solution is studied on a road network in the context of Vehicular Ad hoc Network (VANET). RSUs are used to allow vehicles to connect to Internet. The combination of several features in a deep learning algorithm is interesting. However, RSUs are static and this makes the proposed solution non-adapted in an environment where it is not always possible to deploy a dedicated infrastructure. In<sup>16</sup> authors use a ML model based upon K-nearest neighbors for relay selection in the context of VANETs. The purpose is for a vehicle to select the best RSU in range or, if none are nearby, to select the best candidate (i.e. another vehicle) to forward data to a relay. Though the use of several features and multi-hop data forwarding makes the model adapted to VANETs, a dedicated architecture of RSUs is required. Furthermore, only one class of mobility (motorized vehicles) is considered. Authors in<sup>17</sup> propose a Fuzzy-based Q-Learning Routing Protocol to find the best next hop based on the node's residual energy, movement and buffer space. The proposed solution is tested in different scenarios with several tens of nodes with velocities up to 1.5 m/s. However, in the context of IoV there can be up to hundreds of users and they can move significantly faster. Authors in<sup>18</sup> use deep learning for the selection of a relay in the case of a node having bad radio conditions due to shadowing. There are several metrics used to determine the best relay. However, this solution only considers a user can connect to a base station through one relay (one hop).

## 2.3 | Electing a relay through cluster head selection

The solution proposed in<sup>19</sup> exploits the proximity of buses at intersections to forward data. Bus density, road connectivity and the path of bus lines are used to determine on which bus the data is forwarded. This solution relies on the presence of a bus network with a minimum density to efficiently forward data. Also, the delivery delay from simulations ranges from 7 to 10 seconds. These make the proposed solution inadequate in areas without bus lines or for applications needing a real-time latency. Al-Kharasani et al.<sup>20</sup> propose a clustering solution to connect nodes in a VANET. Several metrics like bandwidth, connectivity, velocity and distance are used to create a cluster of vehicles. However, the proposed solution is tested in scenarios using constant speed and only motorized vehicles (i.e. high-mobility users) are considered. In<sup>21</sup>, the authors propose a scheme where vehicles with similar motion are regrouped in clusters. The cluster head is also connected to the infrastructure using a cellular link. The clustering allows to regroup vehicles with a similar trajectory together so only the head need to be connected to the infrastructure or to other clusters' head. However, in the context of urban mobility several different types of users are present and many intersections will cause users to travel apart from one another with a high probability. In<sup>22</sup>, authors tackle the problem of joint relay assignment and energy-efficiency maximization to assign a relay to clusters of users. To do so, authors propose to formulate the problem considering users' constraints as a maximization problem, solve the problem for each pair of cluster/relay, and finally assign the relay. The assignment is achieved with a linear complexity. However, the performance evaluation focuses on relatively static users and only 12 of them are considered. This is non-adapted to the IoV where hundreds of devices with very different mobility patterns are present at the same time. In<sup>23</sup>, authors propose a scheme to select relays based on network and trust metrics in VANETs to select trustworthy relays. However, the proposed solution makes use of dedicated RSUs and only one class of mobility is considered.

All of these solutions either lack support for several mobility types typically found in cities (pedestrians, e-scooters, bicycles and motorized vehicles), require a dedicated architecture to operate or they consider a limited amount of hops to connect devices to base stations. In our vision, a more general approach with a greater flexibility is mandatory in improving system performance. Thus, we propose a solution that supports different mobility types, can adapt to devices' needs in real-time and only requires one cellular antenna to work. Few devices with cellular connectivity are needed to serve as relays.



**FIGURE 1** The proposed architecture made up of the *Fog* part and the *Cloud* part. Here, 4 users have access to Internet through the cellular antenna.

### 3 | DYNAMIC SELECTION OF RELAYS

In this section, we present the architecture and an overview of the proposed solution (sec. 3.1), we discuss the MR selection algorithm in details (sec. 3.2), we analyze selected algorithms (sec. 3.3) and we explain the novel metric EPX (sec. 3.4).

#### 3.1 | Architecture and overview

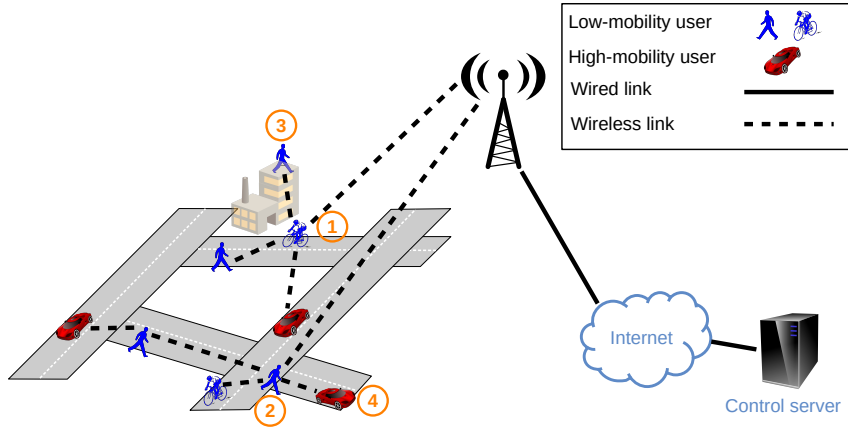
The proposed architecture, inspired from Software Defined Networking (SDN)<sup>24</sup>, is depicted on figure 1 and it is made up of two parts : the *Fog* part and the *Cloud* part. Users with Internet connectivity (pedestrians with a smartphone, connected cars and so on) and the cellular antenna (all circled in green on fig. 1) compose the *Fog* part of the network. On figure 1, fog users have a wireless link to the antenna represented as a black-dashed line. Internet and the Control Server are the *Cloud* part of the architecture (in light blue on fig. 1). This architecture allows a higher degree of flexibility because a dedicated infrastructure is not required, except for one cellular antenna. Even without an antenna users can still communicate locally. Given we assume all of them can be mobile at any time, such an architecture is well adapted compared to a static one where devices can potentially leave the range of the antenna.

The algorithm will run either using a two step mode (*initialization mode*) or a four step mode (*on-going mode*). The *initialization mode* is as follows :

1. Users that have Internet access connect to the controller to send their mobility data (such as velocity, acceleration, type of road the device is on, etc.) as depicted on figure 1;
2. The controller uses the data to determine the mobility profile of the users. The ones with the appropriate profile are selected as relays.

This mode is executed when the network is yet to be built. This is similar to our previous work<sup>10</sup>. Once the network is built, the selection mechanism can be further optimized by running the *on-going mode* :

1. Users send to the controller their mobility data (such as velocity, acceleration, type of road the user is on, etc.) and their network metrics (such as the average delay of their packets and the average amount of transmitted packets), along with their location as depicted on figure 1;
2. The controller uses the mobility data to determine the mobility profile of users. Users with the appropriate profile are pre-selected and tagged as potential relays;
3. The controller uses the network metrics to find the most critical locations. A critical location is one where the values of metrics are considered as too high;



**FIGURE 2** The established topology after a run of the proposed solution.

4. Relays are elected on priority in critical locations. Then, as long as there are locations considered as critical other relays are selected as well.

Figure 2 is an example of the topology once the algorithm has been run and relays have been elected. Relays are directly connected to the antenna. We remind the reader that in this case there were four users with Internet access (see figure 1), labeled from 1 to 4 on figure 2. Here, bicycle user 1 and pedestrian user 2 are the elected MRs. Pedestrian user 3 was not elected because her radio conditions are too bad (she is located inside a building). Car user 4 was not elected as a MR because he is a high-mobility user, not suitable for the stability of the topology. The “appropriate profile” is determined according to local needs. In our case, we consider soft-mobility users (i.e. pedestrians) have the best profile because of their low mobility. Indeed, if they are elected as relays the resulting topology is expected to be more stable compared to electing faster users like connected cars.

The addition of localized metrics such as average delay and EPX (see subsection 3.4) allows to find locations where the density of users and devices is higher. This allows to select relays at these locations which minimizes the number of hops needed by users and devices to reach the relay. The probability of re-transmissions is lower and the end-to-end delay is likely to decrease. This in turn increases QoS. Limiting the number of elected relays minimizes interference between them. Please note that the localized metrics are not part of the classification. They are used after the running of the ML algorithm to enhance the selection mechanism by providing the critical locations.

In our case, the features to classify the mobility-type of users are the position, the velocity and the maximum acceleration. As explained earlier in this section, this data is sent to the controller. However, only the position through time is required (velocity and acceleration can be deduced from it). We suppose the data is measured on a device each 100 ms and sent every second to the controller. This can be further optimized by sending only one position value as long as it does not change. One  $x, y, z$  position can fit in three 4-octets variables (so  $10 \cdot 12$  for 1 second). This data needs to be sent once every tens of seconds in practice because the selection process happens at about this frequency, so the overhead is not very high (a few packets per minute).

### 3.2 | Mobility classification and optimal relay selection

The proposed solution is made up of two principal parts :

1. The ML algorithm classifies users according to their mobility profile and
2. Their network metrics allow to determine which ones from the above step are the most appropriate to act as relays.

We define a network composed of  $U$  users. Each user  $u$  possess one vector of mobility-related metrics  $M$  :

$$M_u = [m_1, m_2, \dots, m_\mu], \quad (1)$$

where  $\mu$  is the number of mobility-related metrics. These metrics can be the current or maximum velocity of the user, his location (i.e. the type of road he is on) and so on. The user periodically records these metrics. Each user  $u$  also stores one vector

of network-related metrics  $N$  :

$$N_u = [n_1, n_2, \dots, n_\nu], \quad (2)$$

where  $\nu$  is the number of network-related metrics the user is experiencing (such as delay, throughput, ETX and so on). The metrics are periodically recorded as well.

Mobility-related metrics (eq. 1) are used by the ML algorithm as features for the classification. The accuracy in determining the mobility type of the user (e.g. pedestrian, bicycle, car, etc.) depends on the choice of features and specific algorithm used. We thus introduce the vector of features for each user  $u$  :

$$F_u = [f_1, f_2, \dots, f_\phi] \quad \text{s.t.} \quad |F_u| > 0 \quad \forall u \quad \text{and} \quad \phi \leq \mu. \quad (3)$$

That is, the features in equation 3 are a subset of the available mobility-related metrics from equation 1. The same  $\phi$  features are used from each user and at least one feature must be used. These features are recorded for a length of time equals to  $\tau$ . The purpose of  $\tau$  is to choose the amount of time during which mobility-related metrics are gathered. Indeed, gathering metrics for a longer amount of time tends to increase accuracy, but it means waiting more before establishing the profile of the user. This can lower QoS, especially in time-sensitive applications so a compromise must be found between accuracy and the delay before predicting the class of a user. The mobility class predicted by the algorithm for each user is thus :

$$c_u^p = L(F_u, \tau). \quad (4)$$

In equation 4,  $L$  is the selected machine learning algorithm,  $F_u$  are the features of user  $u$  (see eq. 3) and  $c_u^p \in C$  is the predicted class of user  $u$ . Given the prediction is not always correct, we introduce the real class of user  $u$ ,  $c_u^r \in C$ , such that :

$$c_u^p = \begin{cases} c_u^r & \text{if the prediction is correct.} \\ c^w \in (C \setminus c_u^r) & \text{if the prediction is wrong.} \end{cases} \quad (5)$$

If the prediction is correct, the predicted and real class of user  $u$  are the same. On the other hand, a failed prediction will classify the user in the wrong class  $c^w$  which is any other class than the real one. The best mobility class  $C_b$  is determined as the most appropriate class to serve as relay. For instance, a slow-mobility class (e.g. pedestrian) can be desirable to increase topology stability.

Once the mobility class of users is determined, the network-related metrics (from eq. 2) are used to find the best location for electing a relay. First, we divide the studied area into  $X$  per  $Y$  tiles. The dimensions of each tile can be as small as to have one user maximum per tile (e.g. 1m x 1m) or as big as to have only one tile for the whole area and for all users :

$$T_{Y,X} = \begin{bmatrix} t_{1,1} & t_{1,2} & \dots & t_{1,X} \\ t_{2,1} & t_{2,2} & \dots & t_{2,X} \\ \vdots & \vdots & \ddots & \vdots \\ t_{Y,1} & t_{Y,2} & \dots & t_{Y,X} \end{bmatrix}, \quad (6)$$

where each  $t_{y,x}$  is one tile. We now define a scoring function  $S$  to compute the score of each tile given the values of network-related metrics of users :

$$t_{y,x} = S(N_u) \quad \forall u \in t_{y,x}. \quad (7)$$

The purpose of tiles is to identify the most critical locations (e.g. tiles with high delay, high ETX, low throughput, etc.) to elect one or more relays inside them. The vector of the most critical locations is introduced :

$$T^{crit} \subseteq T_{Y,X} \quad \text{s.t.} \quad |T^{crit}| > 0. \quad (8)$$

That is, the vector of the most critical locations (eq. 8) is a subset of the matrix of scores (eq. 6). A location can be considered critical when QoS requirements are not met, for instance. Finally, relays  $R$  can be elected once these locations are determined :

$$u \in R \quad \text{iff} \quad (u \in t_{y,x}) \wedge (u \in I) \wedge (u \in C_b) \mid t_{y,x} \in T^{crit}. \quad (9)$$

That is, a user  $u$  will serve as relay if it is located in any critical cell (see eq. 8), has Internet access (belongs to  $I$ ) and is of the correct (best) mobility class (eq. 4 and 5). Note that Internet access is essential for a node to be a potential relay because it must be able to forward traffic back and forth from/to Internet. That is, a node without Internet access is not even a sub-optimal relay compared to one with Internet access; it can not be a relay at all. Hence, these nodes are not considered by the ML algorithm.

Note that in the case where a critical tile has no user that can serve as relay, these users will eventually be connected to a neighboring tile. This neighboring tile will become critical and relays will be elected to address the issue.

---

**Algorithm 1** The selection relay algorithm.

---

```

1:  $maxRelays \leftarrow arg[0]$ 
2:  $potentialRelays \leftarrow NULL$ 
3:  $relays \leftarrow NULL$ 
4: for all users do
5:    $users.gatherMetrics()$ 
6: end for
7: for all users do
8:   if  $A(user) == optimal$  then
9:      $potentialRelays.append(user)$ 
10:  end if
11: end for
12: for  $i = 0$  to  $potentialRelays.size()$  do
13:   if  $computeScore(potentialRelay_i) > criticalScore$  then
14:      $relays.append(potentialRelay_i)$ 
15:   end if
16:   if  $relays.size() \geq maxRelays$  then
17:     break
18:   end if
19: end for
20: return  $relays$ 

```

---

Algorithm 1 shows the relay selection process. It is run whenever the maximum number of relays is not reached, for instance when a MR is disconnecting due to low battery life or because it does not need to be connected to the network any longer. The parameter of the algorithm ( $arg[0]$ ) is the maximum number of relays. Lines 1 to 3 are the initialization steps. The gathering of users' metrics is done in lines 4 to 6 (eq. 1 and 2). In lines 7 to 11, the classification algorithm is run for each users to determine if its mobility profile is suitable to serve as a MR (eq. 4 and 5). If so, it is added to the set of potential relays (line 9). Then, in lines 12 through 19, the score of the potential relays is computed to find the ones that will best increase performance of the network (eq. 8). The score is computed in line 13 (eq. 7) and, if the user is satisfying, it is added to the set of relays in line 14 (eq. 9). The adding of relays can be interrupted if enough relays are present (line 16). Finally, the set of relays is returned in line 20. Note that a user must be in the correct location to become a potential relay. An important advantage of using ML is the fact it can select a relay amongst non-optimal candidates (for instance a car) if there are no optimal candidates (pedestrian) available. This can yield to good performance if the car has a temporary low mobility, perhaps due to a traffic jam. This allows more flexibility compared to an exhaustive search solution as we look at the current features (velocity, acceleration) of nodes rather than their overall mobility profile.

### 3.3 | Benchmark of different ML classification algorithms

We tested several ML algorithms to classify the mobility-type of devices. The results are shown in table 1. The training set was generated with the Simulator of Urban MObility (SUMO, section 4.1). The size of the testing set is 40% of the total data<sup>1</sup>.

The accuracy of the algorithms depends on the delay allowed to gather data. When the delay is very short (2 s for instance), the error is rather high because the user was not active long enough. For example, a car just starting might stop at a red light right away and it is not possible to differentiate it from a bicycle. However, although a longer delay allows for better accuracy, it also means a longer time before selecting a new relay. During this time, no connectivity will degrade QoS and might not be the right choice. This is why a compromise must be found between accuracy and delay to yield the best QoS possible.

Table 1 shows various values of accuracy for three different algorithms: K-nearest neighbors, Support Vector Machine (SVM) and decision tree. For each algorithm, two lines show the accuracy using a 10-cross-validation in classifying users as pedestrian, bicycle or car for the training and testing sets. Decision tree yields the best accuracy with 99% for the training set and ranging

---

<sup>1</sup>We trained the algorithms with different ratios for testing/training sizes and the difference is marginal.



**TABLE 1** The precision in classifying the correct type of mobility against the allowed delay. The table shows the values for the training and testing sets using 10-cross-validation.

|                      |          | Classification delay (in s) |    |     |    |    |
|----------------------|----------|-----------------------------|----|-----|----|----|
|                      |          | 2                           | 4  | 6   | 10 | 14 |
| K-nearest (in %)     | Training | 97                          | 97 | 97  | 97 | 97 |
|                      | Testing  | 85                          | 82 | 80  | 87 | 90 |
| SVM (in %)           | Training | 93                          | 93 | 93  | 93 | 93 |
|                      | Testing  | 85                          | 77 | 73  | 79 | 78 |
| Decision tree (in %) | Training | 99                          | 99 | 99  | 99 | 99 |
|                      | Testing  | 90                          | 98 | 100 | 98 | 98 |

**TABLE 2** The precision, recall, f-score and the sample size when using the decision tree algorithm.

|                   |            | Classification delay (in s) |     |     |     |     |
|-------------------|------------|-----------------------------|-----|-----|-----|-----|
|                   |            | 2                           | 4   | 6   | 10  | 14  |
| Precision (in %)  | Pedestrian | 22                          | 42  | 70  | 72  | 73  |
|                   | Bicycle    | 100                         | 100 | 100 | 100 | 100 |
|                   | Car        | 100                         | 100 | 100 | 100 | 100 |
| Recall (in %)     | Pedestrian | 100                         | 100 | 100 | 100 | 100 |
|                   | Bicycle    | 100                         | 40  | 100 | 78  | 100 |
|                   | Car        | 65                          | 88  | 93  | 95  | 97  |
| F-score (in %)    | Pedestrian | 36                          | 59  | 82  | 84  | 84  |
|                   | Bicycle    | 100                         | 57  | 100 | 88  | 100 |
|                   | Car        | 79                          | 94  | 96  | 98  | 96  |
| Number of samples | Pedestrian | 2                           | 5   | 7   | 13  | 19  |
|                   | Bicycle    | 1                           | 5   | 5   | 9   | 9   |
|                   | Car        | 20                          | 33  | 40  | 66  | 95  |

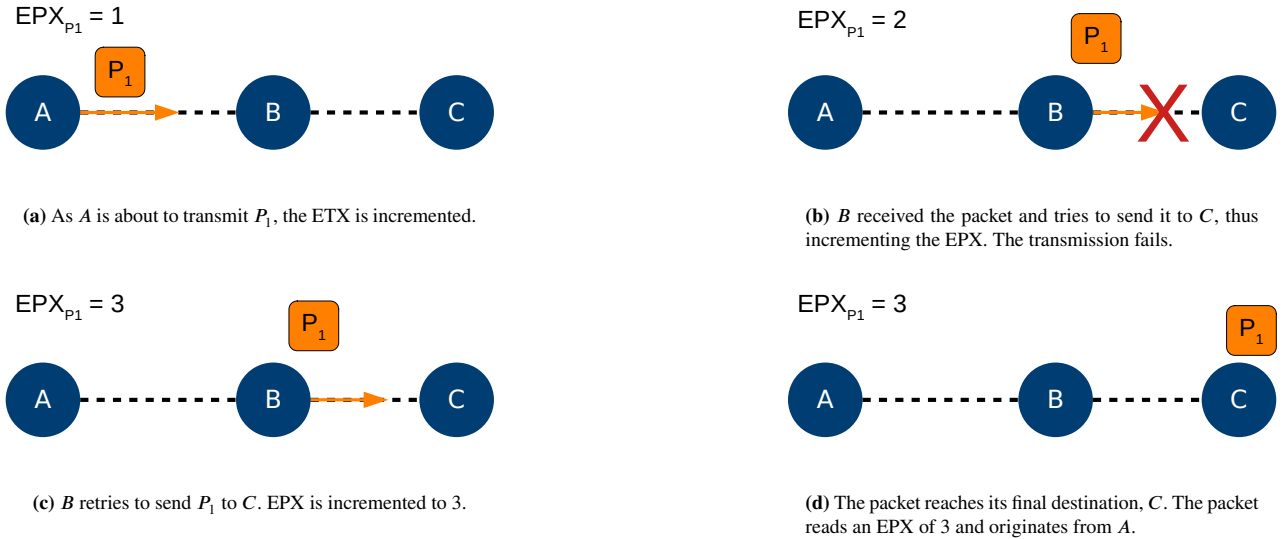
from 90% to 100% during testing. K-nearest is the second most accurate with a training accuracy of 97% and a testing accuracy from 80% (6 s) to 90% (14 s). SVM is the least accurate with correct classification in 93% of cases for the training set and between 73% and 85% when testing.

The longest delays do not significantly improve the accuracy of the classification. This means a very long delay is not necessarily better. This is why we use decision tree with delays up to 6 s in the evaluation of performance (sec. 4). In terms of complexity, decision tree is restricted to a depth of 2 to avoid over-fitting and there are 3 labels as output (see section 4).

Table 2 depicts the precision, recall and the resulting f-score for the decision tree algorithm according to the type of mobility and delay allowed for the classification. The number of pedestrians, bicycles and cars (“Number of samples”) used during the evaluation is also in the table. The proportion of bicycles is lower compared to the two others (less significant). Cars are easier to identify because of the potential high velocity they can reach, although when the allowed delay is very short (2 s), they do not have time to reach such speed. This explains the relative low recall (65%) compared to a classification delay of 14 s (recall is then 97%). The precision when identifying pedestrians is also quite low when the delay is 2 s (22%). This is coherent with results from section 4, where using a delay of 4 s or 6 s yields better results (the classification is more relevant).

### 3.4 | New metric : the Expected Packet Count (EPX)

The EPX is different from Expected Transmission Count (ETX)<sup>25</sup>. It is the total amount of (re-)transmissions needed for packets sent from a location to reach their destination during a certain time frame. Each hop in the path of these packets counts towards EPX including re-transmissions. It is the "density of packets" of a given location. A high EPX is the consequence of long paths to reach a relay, bad radio conditions (many re-transmissions) and/or simply high traffic.



**FIGURE 3** An example of the computation of EPX: packet  $P_1$  is emitted from A and has destination sink C. The EPX, stored in  $P_1$ , concerns the area where A is located.

Figure 3 shows a simple example of the computation of EPX. One packet is emitted from node A and its destination is the sink node C. First, when the packet is about to be sent after its creation, the value of EPX is incremented to 1 (fig. 3a). The location of the packet's source is kept in the packet. When the packet is received on B, its EPX value is incremented as it is about to be forwarded to C (fig. 3b). The transmission fails due to interference and is not acknowledged by C. B increments the EPX value again and resends the packet (fig. 3c) to C. The value of EPX is now 3. Finally, C successfully receives the packet (fig. 3d). Note that the value of EPX from  $P_1$  concerns the area where the source (here, A) is located, not the source itself. Thus, if a sink receives multiple packets from a same location, the total EPX is the sum of all EPX values from all packets from this area.

## 4 | PERFORMANCE EVALUATION

The evaluation of performance is done by comparing four different algorithms. Two scenarios are studied and two parameters vary (thus, there are 4 series of graphs). The two first graphs of each series (figures a and b in section 4.3) represent the scenario with a low density of users (column *Low density scenario* in table 3). The third and fourth graphs of each series (figures c and d in section 4.3) refer to the scenario with a high density of users (column *High density scenario* in table 3). The line charts depict the variation of traffic generated by users, whereas the bar charts represent the change of selection delay (the time during which network related metrics are gathered for the purpose of classification of the mobility type of users). The studied algorithms are :

1. *Without classification* does not use any classification. Relays are selected randomly. In line charts it is represented by red circles. In bar charts it is represented as red-crossed bars with a selection delay of 0 s;
2. *With classification* uses classification of mobility profile to select a new relay<sup>10</sup>. Relays with low mobility are favored. In graphs, this algorithm is represented by blue diamonds (line) and blue horizontally-lined bars (bar charts);
3. *With classification and metrics* is the proposed solution. It uses mobility classification (as *With classification*) and a score function (see below). It is represented in line charts as green squares and in bar charts as green-filled bars;
4. *Static RSUs* does not use classification (selection delay of 0 s). RSUs are placed on the studied as static units. In graphs, this algorithm is represented by purple crosses (line charts) and purple vertically-lined bars (bar charts).

Given the classification requires some time to gather data to determine the mobility profile of users, the selection delay of the algorithms without classification is always 0 s (no delay). The score function (see eq. 9) for the solution using classification with metrics combines the delay of tiles with the EPX.

**TABLE 3** Values for the most relevant parameters. †The variable parameter is either traffic density (from 10 to 15 packets/s) or the relay selection delay (from 2 to 6 s). The solutions without classification always use 0 s).

| Parameter Value                | Low density scenario | High density scenario |
|--------------------------------|----------------------|-----------------------|
| Playground size                | 1400 m · 1400 m      | 1400 m · 1400 m       |
| User generation period         | ≈ 3 s                | ≈ 0.75 s              |
| Num. of users, total           | ≈ 330                | ≈ 1250                |
| Num. of users, inst.           | 30-50                | 100-150               |
| Ratio of relays                | max. 5 %             | max. 5 %              |
| Car:bike:pedestrian ratio      | 3 : 1 : 1.25         | 2.75 : 1 : 1.5        |
| Simulation duration            | 1000 s               | 1000 s                |
| Repetitions                    | 10                   | 10                    |
| Traffic model                  | VBR                  | VBR                   |
| <b>Traffic density</b> †       | 10 to 15 packets/s   | 10 to 15 packets/s    |
| <b>Relay selection delay</b> † | [0, 2, 4, 6] s       | [0, 2, 4, 6] s        |
| Data packet size               | 1280 bytes           | 1280 bytes            |
| Max. user throughput           | 10 Mbps              | 10 Mbps               |

## 4.1 | Simulation setup

Simulations were done using the simulator Omnet++ 5.6.1. Users' trip were generated using SUMO (Simulation of Urban MObility) 1.3.1. The studied area is the location of the University Gustave-Eiffel (France) and was downloaded from OpenStreetMap. SUMO allows to randomly generate trips for different kinds of users (we used cars, bicycles and pedestrians) in a realistic way. Users follow roads from the downloaded OpenStreetMap area (sidewalks in the case of pedestrians) and obey to traffic rules. Once the trips of users have been created in SUMO, they are exported to a XML file. Then, the Omnet simulation will create one "module" (or node) for each user that was generated by SUMO. To train the ML algorithm doing the classification, we ran another simulation on a different location, allowing us to test the generalization of the algorithm. The delay of the prediction allows to gather more or less data from users. A lower delay is less accurate (about 70% accuracy for 2 s) because fewer data from a user's position, acceleration, speed and so on were accumulated. On the other hand, waiting a longer time to make the prediction means a higher precision (about 94% for 6 s) but users are not connected during that period.

The list of the most important parameters is presented in table 3. The size of the area is about 1.96 km<sup>2</sup>. Users are created on the network each 3 s (low density case) or 0.75 s (high density case). The total and instantaneous number of users represent, respectively, the amount of users created on one run and how much of them there are at any time. About 5% of the instantaneous number of users are elected as relays. Most of the users are in cars and the smallest part of users are bicycles. Simulations were 1000 s in length with 10 repetitions to increase results accuracy. All users except relays generate data following a Variable Bit Rate (VBR). Data packets are set to the minimum size of an IPv6 packet<sup>26</sup> and users have a maximum throughput of about 10 Mbps<sup>27</sup>. The maximum range of users is about 105-110 m with a path loss based on a  $-log_{10}(d)$  ( $d$  is the distance) model<sup>28 29</sup>. The amount of static RSUs (fourth algorithm) is set to 5. This is about the same amount compared to the others (5% of relays), though it is purposely constant to reflect the fact the infrastructure is static.

## 4.2 | Studied performance indices

This section provides explanations concerning the Performance Indices (PIs) used for the performance evaluation. We use the term "PI" to differentiate results shown in this section from metrics used to build the topology (as explained in section 3). PIs are presented in the same order as the next section (sec. 4.3) for the sake of clarity.

### 4.2.1 | Packet delivery ratio

The packet delivery ratio (PDR) represents the percentage of generated data packets that reach their final destination. Some factors such as frequent disconnections of nodes from their neighbors or very bad radio conditions causing the amount of re-transmissions exceeding the maximum number of retries will cause PDR to decrease. Note that a packet being re-transmitted many times on each hop can still yield to a high PDR, though other PIs such as delay will increase. The higher the PDR the better.

## 4.2.2 | End delay

The end delay is the time taken for a packet to reach its final destination. It depends on how good radio conditions are (many re-transmissions will increase delay) and on the amount of congestion (buffer occupancy). A low throughput means nodes require more time to send packets, increasing delay. The amount of nodes a packet has to go through on its path (hop count) does not necessarily increase delay, as a long path composed of nodes with high throughput and low congestion will perform better than a short congested path. The lower the delay the better.

## 4.2.3 | Energy consumption

The energy consumption per node represents the amount of energy spent by each node on transmitting and receiving packets, including signaling and data packets. A high number of transmissions per unit time implies increased duty-cycling. Bad radio conditions resulting in many re-transmissions and topology instability causing more signaling are examples of causes of increase energy consumption. Energy consumption can be very important, especially for devices that can not have their battery recharged. The lower the consumption the better.

## 4.2.4 | Throughput usage

The throughput usage represents the percentage of the total throughput a node can achieve that is in use. For instance, if a node can achieve a maximum throughput of 10 Mbps, a throughput usage of 20% means the node sends on average 2 Mbps of packets, including signaling and data. A lower value is achieved when good radio conditions (fewer re-transmissions) are present or if the topology is more stable (less signaling). The lower the throughput usage the better.

## 4.2.5 | Relative amount of packets reaching destination

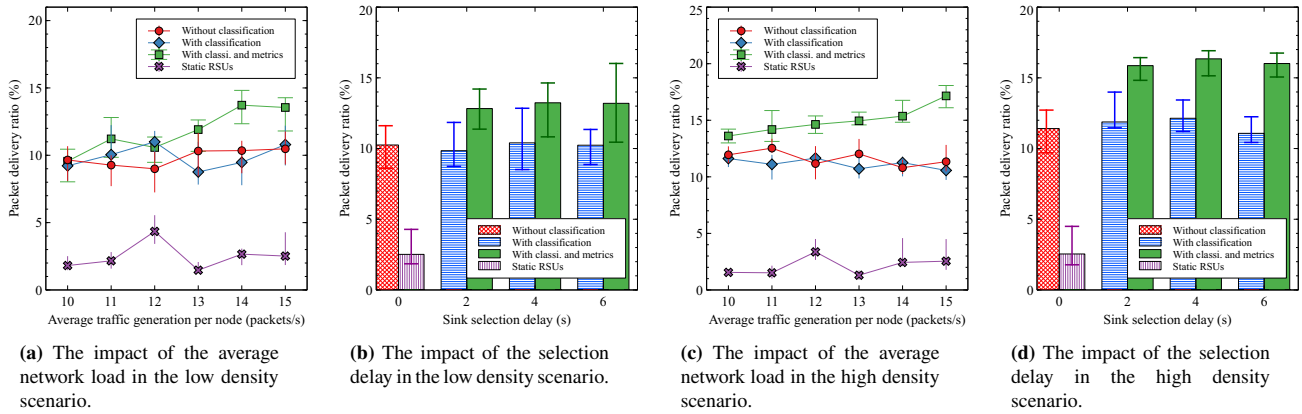
PDR, end delay and throughput usage are not enough to see how well the different solutions perform. Indeed, if a solution generates a very low amount of packets due to lack of connectivity, the PDR of these packets can still be very high, their delay very short and the usage of throughput will be low as well. The relative amount of packets reaching destination is the number of packets that reached the final destination compared to the total amount of packets created. It is the PDR times the total amount of generated packets and represents how well nodes are connected to a relay. The higher the better.

## 4.3 | Simulation results

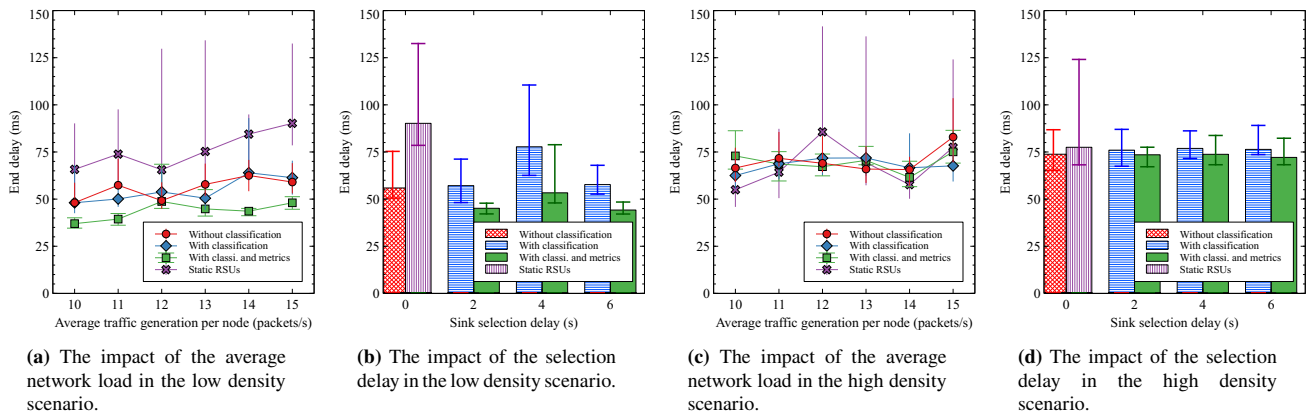
The following results are regrouped by the studied metric. We remind the reader that in line charts (where traffic load varies), the selection delay is 2 s. In bar charts (representing the variation of selection delay), the traffic load is set to 15 packets/s.

### 4.3.1 | Packet delivery ratio

Figure 4 shows the PDR of the different algorithm. When studying the impact of traffic in the low density scenario, (fig. 4a), the proposed solution performs significantly better at higher loads. This is because taking into account network metrics allows it to adapt the location of relays where the need is greater. On the other hand, at lower traffic loads (figs. 4a) all solutions perform similarly, except the *static RSUs*. This is because mobile users eventually travel out of range of RSUs resulting in a very low PDR. In the case of a higher density of users (figs. 4c and 4d), the proposed solution almost always performs significantly better because of more traffic generated in absolute. Indeed, in the case of a higher density of traffic, the use of network metrics allows to select relays in the most critical locations. Algorithm *with classification* has similar results compare to *without classification* on figures 4a and 4c because the low amount of relays (5%, see table 3) and the fact that users within the correct class of mobility are not differentiated from one another makes it harder to select a candidate in a critical location. The same is seen on figures 4b and 4d. The variance of the proposed solution increases with the selection delay in the low density scenario (fig. 4b) because the network metrics used for the scoring function are older (less accurate). This is not the case in the second scenario 4d because the higher density of users greatly increases the amount of network metrics and better locations are available to select a relay.



**FIGURE 4** The packet delivery ratio of the different scenarios. The vertical axis has the same scale for all 4 graphs.



**FIGURE 5** The average end delay results of the different scenarios. The vertical axis also has the same scale on all graphs.

### 4.3.2 | End delay

Figure 5 shows the average end delay of data packets for the different solutions. It is interesting to see that although the PDR was not better in the low density scenario (fig. 4a) for the proposed solution, the end delay is better (fig. 5a). This is because even though there are fewer choices to select a potential relay, the network metrics still help in determining the most critical locations. When studying the selection delay in the low density scenario (see fig. 5b), the proposed solution is better, except when the selection delay is 4 s. As stated at the beginning of section 4.1, this is due to the fact an average selection delay will decrease the accuracy of the network metrics but it will only increase the classification accuracy by a moderate amount. On the other hand, either a short selection delay (accurate network metrics and shorter waiting time) or a longer selection delay (much more accurate classification) are preferable. For the high density scenario (figs. 5c and 5d), the delay is comparable to the other algorithms. In this case, the high amount of users makes the delay and EPX increase in all locations. The other solutions perform similarly (figs. 5a, 5b, 5c and 5d) for the same reasons mentioned before. The variance of *static RSUs* tends to be very high because the location of RSUs varies from one run to another and users tend to travel out of their range. This highlights the relevance of dynamically selecting relays in the context of IoV.

### 4.3.3 | Energy consumption

The average energy consumption per user is depicted on figure 6. In all scenarios (figs. 6a, 6b, 6c and 6d), the proposed solution consumes less energy. The higher PDR (fig. 4) and the similar or lower end delay (fig. 5) show that there are less re-transmissions, resulting in a lower consumption of energy. It is worth noting that even though there are more users and packets generated on the high density scenario (see fig. 8), the consumption per user is not necessarily higher. This is due to the fact that more paths are

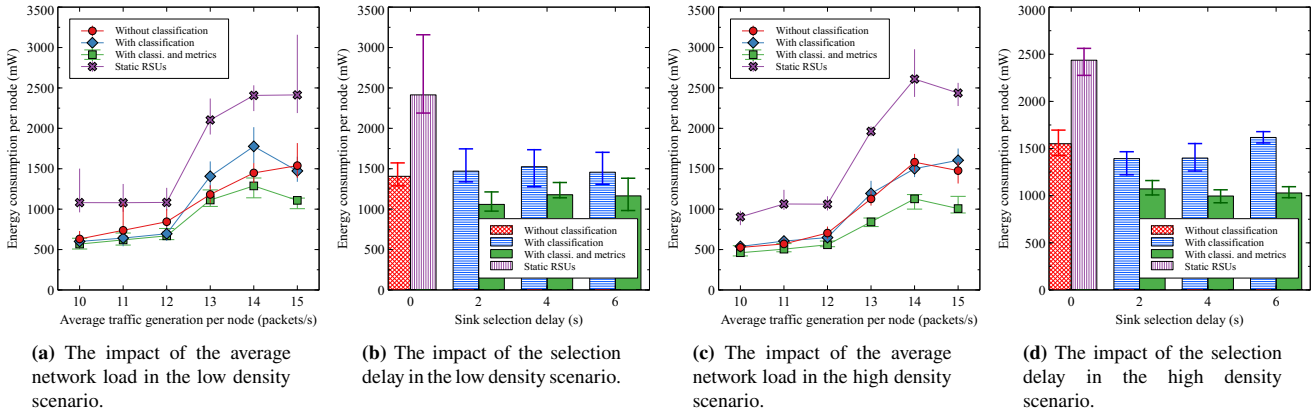


FIGURE 6 The average energy consumption per node in the different scenarios.

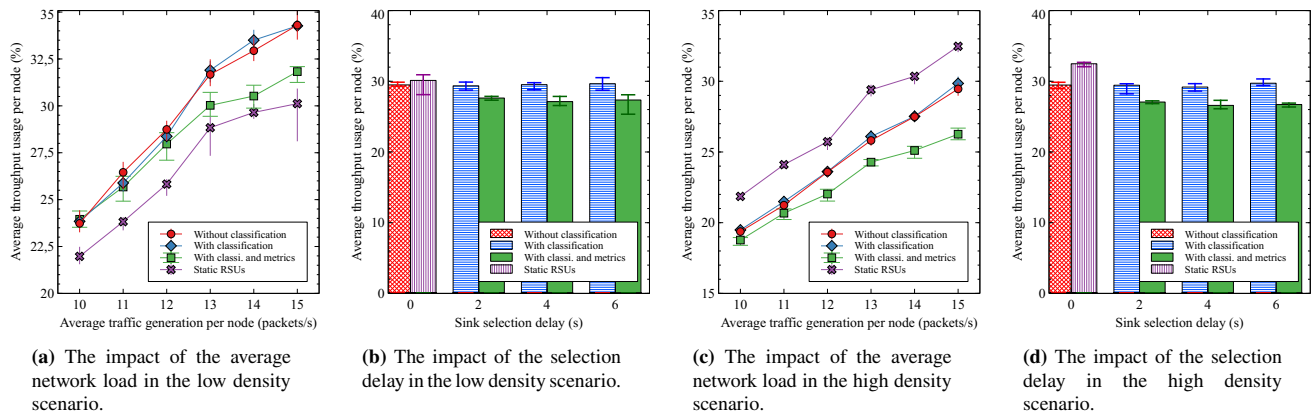


FIGURE 7 The average throughput usage in the different scenarios.

available, so the PDR is higher (as depicted on fig. 4c) and there are less re-transmissions. *Static RSUs* consume more energy because of lower PDR (fig. 4) and higher delays (fig. 5) caused by more re-transmissions and losses.

#### 4.3.4 | Throughput usage

Figure 7 shows the average throughput usage per node of the different solutions. In the low density and high density scenarios (figs. 7a and 7c), the usage increases with load for all algorithms. The usage is higher in the low density case compared to the high density case due to a higher PDR (fig. 4a compared to fig. 4c) and more re-transmissions. *Static RSUs*' PDR is similar in low and high density cases, so the throughput usage is similar. The selection delay (figs. 7b and 7d) does not impact much the throughput usage, though the variance increases slightly for the algorithm *with classification and metrics* in the low density case (fig. 7b) because of the decreased accuracy of the network metrics.

#### 4.3.5 | Relative amount of packets reaching destination

The relative amount of packets reaching their final destination compared to the amount of generated packets is presented in figure 8. The proposed solution shows better results in the low density scenario at higher loads (fig. 8a) and in the high density scenario (fig. 8c). This follows the results on the PDR (fig. 4a and fig. 4c). The selection delay does not impact the relative amount of arrived packets for the proposed solution in both scenarios (figs. 8c and 8d). The low PDR of *Static RSUs* explain why the amount of packets reaching the destination is low.

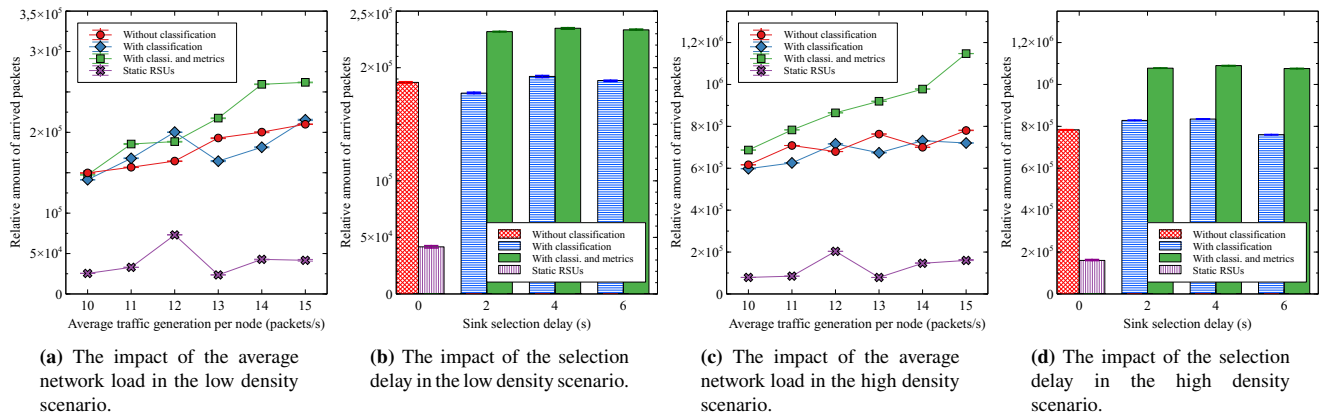


FIGURE 8 The relative amount of packets that reached their final destination.

## 5 | CONCLUSION

Guaranteeing QoS in the IoV is still a challenging issue. The different mobility profile of users (pedestrians, bicycles, e-scooters, cars, buses and so on), their different velocity and their different paths contribute to the instability of the topology. We are convinced our solution to dynamically select relays amongst users by running a machine learning algorithm to classify users according to their mobility profile and the use of network metrics to find critical locations is key in improving QoS in the IoV. Results show the proposed solution increases the PDR by up to 6%, reduces the average energy consumption of users by up to 30% and increase topology stability by reducing re-transmissions. Future work will focus on extending the solution to support more wireless technologies and network metrics.

## References

1. Sakiz F, Sen S. A survey of attacks and detection mechanisms on intelligent transportation systems: VANETs and IoV. *Ad Hoc Networks* 2017; 61: 33–50.
2. Ji B, Zhang X, Mumtaz S, et al. Survey on the Internet of Vehicles: Network Architectures and Applications. *IEEE Communications Standards Magazine* 2020; 4(1): 34–41.
3. Stoyanova M, Nikoloudakis Y, Panagiotakis S, Pallis E, Markakis EK. A Survey on the Internet of Things (IoT) Forensics: Challenges, Approaches and Open Issues. *IEEE Communications Surveys & Tutorials* 2020.
4. Cheng J, Cheng J, Zhou M, Liu F, Gao S, Liu C. Routing in internet of vehicles: A review. *IEEE Transactions on Intelligent Transportation Systems* 2015; 16(5): 2339–2352.
5. Shamganth K, Sibley MJ. A survey on relay selection in cooperative device-to-device (D2D) communication for 5G cellular networks. In: IEEE. ; 2017: 42–46.
6. Rachedi A, Badis H. BadZak: An hybrid architecture based on virtual backbone and software defined network for Internet of vehicles. In: IEEE. ; 2018; Kansas City, MO, USA: 1–7.
7. Gasmri R, Aliouat M, Seba H. A Stable Link Based Zone Routing Protocol (SL-ZRP) for Internet of Vehicles Environment. *Wireless Personal Communications* 2020: 1–16.
8. Fabian P, Rachedi A, Gueguen C. Programmable objective function for data transportation in the Internet of Vehicles. *Transactions on emerging telecommunications technologies* 2020. doi: 10.1002/ett.3882
9. Alexander R, Brandt A, Vasseur J, et al. RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks. RFC 6550; 2012

10. Fabian P, Rachedi A. Dynamic selection of relays based on classification of mobility profile in a highly mobile context. In: IEEE. ; 2020: 1–6.
11. Abada D, Massa A, Boulouz A, Salah MB. An Adaptive Vehicular Relay and Gateway Selection Scheme for Connecting VANETs to Internet via 4G LTE Cellular Network. In: Springer. 2020 (pp. 149–163).
12. Liu D, Wang J, Xu Y, Xu Y, Yang Y, Wu Q. Opportunistic mobility utilization in flying ad-hoc networks: A dynamic matching approach. *IEEE Communications Letters* 2019; 23(4): 728–731.
13. Dao NN, Park M, Kim J, Paek J, Cho S. Resource-aware relay selection for inter-cell interference avoidance in 5G heterogeneous network for Internet of Things systems. *Future Generation Computer Systems* 2019; 93: 877–887.
14. Liao Z, Liang J, Feng C. Mobile relay deployment in multihop relay networks. *Computer Communications* 2017; 112: 14–21.
15. Mchergui A, Moulahi T, Nasri S. Relay Selection Based on Deep Learning for Broadcasting in VANET. In: IEEE. ; 2019: 865–870.
16. Zhou Y, Li H, Shi C, Lu N, Cheng N. A fuzzy-rule based data delivery scheme in VANETs with intelligent speed prediction and relay selection. *Wireless Communications and Mobile Computing* 2018; 2018.
17. Dhurandher SK, Singh J, Obaidat M, Woungang I, Srivastava S, Rodrigues J. Reinforcement Learning-Based Routing Protocol for Opportunistic Networks. In: IEEE. ; 2020: 1–6.
18. Abdelreheem A, Omer OA, Esmail H, Mohamed US. Deep Learning-Based Relay Selection In D2D Millimeter Wave Communications. In: IEEE. ; 2019: 1–5.
19. Zhang Y, Tang X, Xu Y, Chen W. Data Forwarding at Intersections in Urban Bus Ad Hoc Networks. In: IEEE. ; 2020: 1–6.
20. Al-Kharasani NM, Zukarnain ZA, Subramaniam SK, Hanapi ZM. An Adaptive Relay Selection Scheme for Enhancing Network Stability in VANETs. *IEEE Access* 2020.
21. Zhao Y, Jiang L. A scheme of D2D-based delay analysis and vehicle relay algorithm in VANET. In: . 715. IOP Publishing. ; 2020: 012030.
22. Baidas MW. Joint relay assignment and energy-efficiency maximization in energy-harvesting downlink/uplink clustered nonorthogonal multiple-access networks. *Transactions on Emerging Telecommunications Technologies* 2020: e3962.
23. Dahmane S, Kerrache CA, Lagraa N, Lorenz P. WeiSTARS: A weighted trust-aware relay selection scheme for VANET. In: IEEE. ; 2017: 1–6.
24. Taleb T, Samdanis K, Mada B, Flinck H, Dutta S, Sabella D. On multi-access edge computing: A survey of the emerging 5G network edge cloud architecture and orchestration. *IEEE Communications Surveys & Tutorials* 2017; 19(3): 1657–1681.
25. De Couto DSJ. *High-throughput routing for multi-hop wireless networks*. PhD thesis. Massachusetts Institute of Technology, 2004.
26. Deering DSE, Hinden B. Internet Protocol, Version 6 (IPv6) Specification. RFC 8200; 2017
27. Stallings W. *Data and Computer Communications 9th Edition*. 2011.
28. Debnath SK, Saha M, Funabiki N, Kao WC. A throughput estimation model for IEEE 802.11 n MIMO link in wireless local-area networks. In: IEEE. ; 2018; Nagoya, Japan: 327–331.
29. Giordani M, Zanella A, Higuchi T, Altintas O, Zorzi M. On the Feasibility of Integrating mmWave and IEEE 802.11 p for V2V Communications. In: IEEE. ; 2018; Chicago, IL, USA, USA: 1–7.

