Effective Prediction of V2I Link Lifetime and Vehicle's Next Cell for Software Defined Vehicular Networks: A Machine Learning Approach

Soufian Toufga*, Slim Abdellatif†, Philippe Owezarski*, Thierry Villemur‡, Doria Relizani*
* LAAS-CNRS, Universite de Toulouse, CNRS, Toulouse, France
† LAAS-CNRS, Universite de Toulouse, CNRS, INSA, Toulouse, France
‡ LAAS-CNRS, Universite de Toulouse, CNRS, UT2J, Toulouse, France
{*toufga, slim, owe, villemur, relizani*}@laas.fr

***Abstract*—Predicting, in the one hand, the time duration that a vehicle remains associated to a cell i.e. Network Attachment Point (NAP) and, on the other hand, the next cell can help anticipating network control decisions to provide services with stringent requirements despite vehicle mobility. In this paper, we propose a machine learning based approach for Software Defined Vehicular Networks that allows a cell to estimate the attachment duration of each newly associated vehicle at the association request time, as well as, a prediction of the upcoming cell, performed at the SDN controller that controls the cells. Our proposed models have been evaluated on a large dataset, which we have generated based on a real mobility trace from the city of Luxembourg, and the evaluation shows promising results in terms of prediction accuracy.**

***Index Terms*—V2I communication, link lifetime estimation, Supervised Machine Learning, Software Defined Vehicular Networks**

I. Introduction

Under the umbrella of cooperative Intelligent Transportation Systems (ITS) and automated driving, a variety of emerging services are envisioned for the near future with diverse performance requirements on V2V (Vehicle-to-Vehicle) and V2I (Vehicle-to-Infrastructure) communications in terms of transfer delay, reliability and bandwidth requirements. The firmness of some of these requirements makes current wireless technologies unsuitable, and one possible research direction is to consider a Software Defined Network (SDN) based hybrid (LTE based and DSRC, etc.) vehicular Network as the access network infrastructure to support these emerging services [1] [2]. Indeed, (1) the ability given to vehicles to associate, during their trip, either simultaneously or consecutively, to multiple network attachment Points (Base Station (BS), Road Side Unit (RSU), etc.), and (2) the "logical" centralized control based on a thorough visibility of the network, combined with the fine-grained and programmable selection and forwarding treatments of flows, inherent to SDN, can bring a noticeable boost to the emergence of these services.

In this paper, we are interested in estimating the cell attachment duration of vehicles (or V2I wireless link lifetime) as well as identifying the next Most Probable Cell (MPC) to which the vehicle is supposed to handover using machine learning based techniques with, amongst, the following expected benefits: First, having an estimate of the cell attachment duration of each associated vehicle as well as a view of upcoming vehicles helps each cell to effectively use its network resources by potentially anticipating the arrival of traffic with strict performance requirements (e.g. [3]). Similarly, for delay sensitive services in a fog/edge computing architecture, some network functionalities and application services can be placed close to the users at the cell premises to provide short latencies. Anticipating the vehicle handover helps triggering a proactive service migration to the identified next cell. Last, network topology discovery is a core function of an SDN controller since it builds at the controller the overall vehicular network topology with V2V and V2I wireless links [4]. This view is then exposed to network control applications. Obviously, very short-live wireless links should not be reported to the SDN controller in order to avoid transient and inconsistent network decisions. The topology discovery service can leverage on the wireless link lifetime estimation to that end.
A last contribution of this work is the creation of a large dataset using simulators used in the automotive industry combining real vehicle mobility traces collected in a large European city (e.g. Luxemburg) and network related information at cells from the network infrastructure of a real operator.

The remainder of the paper is organised as follows. Section II gives a description of related works. The Dataset generation and collected features are explained in section III. Section IV presents the proposed approach, while Section V presents the performance evaluation. Section VI discusses the obtained results. Finally, Section VII concludes this paper.

II. Related Work

During the last decade, wireless link duration estimation and characterization has been researched in the context of wireless multi-hop mobile networks and considered as a crucial piece towards an effective routing in Mobile Ad-hoc NETwork (MANET). Most assumed a predefined mobility model and simplified radio propagation models that, in fact, do not accurately match the reality especially in urban or in-door environments. For V2V wireless link prediction in

Vehicular Ad-hoc NETwork (VANET), some work assumed that the speed of vehicles remains constant (as in [5]), or that vehicles moves along a straight highway [6], [7] assuming some predefined probability distribution for vehicle speed (as in [6]–[10]). Based on similar assumptions (i.e. trains moving along a straight line at constant speed), V2I link prediction was also researched as part of the routing algorithm for railway-specific Software defined LTE-based high speed vehicular networks [11].

Machine learning based techniques have recently emerged as an alternative to these model-based wireless link prediction techniques alleviating the need to resort to simplifying assumptions on vehicle mobility. The work in [12] relies on the use of alert messages sent by each vehicle periodically to convey information used by surrounding vehicles to feed a neuronal network in charge of predicting the expected average speed of the vehicle, from which it derives the V2V wireless link duration Similarly, the work in [13] relies on regular message exchanges between vehicles to collect various V2V wireless link metrics that are then transmitted to the infrastructure to feed a set of predictors that are combined using the adaboost algorithm [14] to build a more accurate prediction of V2V wireless link duration.

This work is rather focused on V2I wireless links in an infrastructure based vehicular network where typically a vehicle get attached to multiple RSUs/BS during its trip. Our prediction of cell (RSU/BS) attachment duration doesn't preclude any predefined assumption on vehicle mobility and, in comparison to the above cited work, it incurs very limited message transmission overhead since the prediction is triggered only at RSU/BS association request.

Next-cell prediction has been researched mainly in the context of LTE cellular networks as a way to improve handover delays. Similarly to link lifetime predictions, many proposals relied on stochastic models, mostly Markov chain based decision processes that take as input a probability transition matrix that describes the potential transitions between adjacent cells. One of the main challenges of these approaches is how to set the probability transition matrix to make it work regardless of any cause that may impact the mobility of vehicles, e.g. the day, the time of day, traffic jam, or any unusual event.

The work in [15] and [16] follow a different approach based on a user mobility database that records, for each vehicle, its mobility during its past history. This database is maintained by the network thanks to position updates sent regularly by each vehicle during its trip. Next cell prediction is based on vehicles' history and the history of neighbouring vehicles flowing in the same direction. Both approaches work well for regular drivers with quite steady itinerary habits but requires tremendous computing/storage at the base station as well as transmission resources.

[17] employs machine learning from the CSI (Channel State Information) observed by the vehicle while passing through the current cell (in addition to the previous cell from which it originated). Since the CSI is regularly conveyed by LTE protocols to the base station, no extra transmission is needed to report the inputs required by the prediction algorithm. Also, the prediction algorithm is continuously updated during system operation, by renewing the training with all users traversing the cell, which increases the prediction accuracy. The main limitation of the proposed approach is the size of the CSI sequence that is needed to get an accurate prediction. The performance evaluations of [17] show that this may take 70% of the path, which could be prohibitive for small cells or fast cars, with not enough time left to proactively trigger some handover procedures.

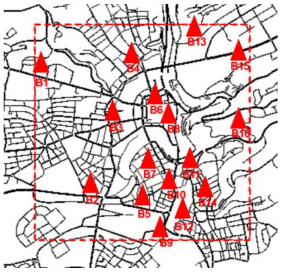
Our proposed next cell prediction scheme is also based on a supervised machine learning scheme. As for V2I link lifetime prediction, it incurs very little overhead, since the information required by the predictor is piggybacked with the association request message. In comparison to [17], our prediction is computed and made available at association time, while still achieving accurate predictions in line with the performance of [17].

III. Input Dataset

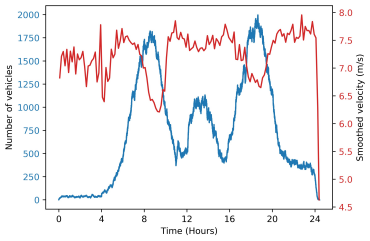
Below, we present the dataset that we created. we first describe the data collection process, then the collected features.

A. Data Collection

The dataset employed in this paper was generated using the VEINS framework [18], which is an open source framework for running vehicular network simulations. It is based on two well-established simulators: OMNeT++ [19], an event-based network simulator, and SUMO [20], a microscopic road traffic simulator. OMNET with the SimuLTE-based LTE extension is responsible for simulating the LTE protocol stack (signal strength, handover, connectivity), while SUMO is responsible for vehicle mobility. The global framework provides a realistic simulation of LTE connectivity for vehicles. Our setup consists of two main parts: the first one concerns the LTE network setup while the second one is relevant to the vehicle mobility setup. We used the Luxembourg SUMO Traffic (LuST) Scenario by Codeca et al. [21]. It is generated using SUMO and is realised in Luxembourg city. The trace reproduces the mobility behavior of almost 300 000 vehicles composed of different types of vehicles (personal vehicles, public transport vehicles, etc.) in an area of 156 Km^2 during 24h. In our study, we focused mainly on the urban scenario; for instance, we selected an area of 2.5*2.5 km in the city-center composed of residential and arterial roads. For the LTE network, we used the eNodeB locations of a Luxembourg mobile network operator introduced by the project LuST-LTE [22]. We selected 16 eNodeBs as shown in Figure 1a. For the LTE network simulation settings, we used those provided by default by the SimuLTE project [23], e.g. the handover procedures implemented by SimuLTE [23], described in [22], is based on the Signal-to-Interference-and-Noise-Ratio (SINR) instead of RSRP (Reference Signal Received Power) and RSRQ (Reference Signal Received Quality) as is usually the case in LTE.



(a) eNodeB locations



(b) Mobility information

Fig. 1: Considered urban scenario details

TABLE I: Collected (c) and Generated (g) features

Feature Name	Feature Description
(c) Vehicle Id	a unique identifier per vehicle.
(c) Vehicle position	coordinates (x,y), which can be converted to GPS coordinates
(c) Speed (m/s)	vehicle speed in mps
(c) Serving Cell id	serving cell id for a vehicle
(c) Serving Cell position	coordinates (x,y), which can be converted to GPS coordinates
(c) Timestamp of association (s)	timestamp when a vehicle joins a Cell
(c) Timestamp of dissociation (s)	timestamp when a vehicle detach from a Cell
(c) Road_Id	road identifier
(c) Line_Id	line identifier
(g) Distance to serving Cell (m)	distance between the serving cell and vehicle in meters
(g) Link duration (s)	time spent by a vehicle under the coverage of a Cell
(g) Cell load	number of vehicles that are under the coverage of a Cell at the same time
(g) Previous cell	previous cell to which it was connected,
(g) Next cell	next cell to which it will be connected
(g) Theta	the angle between the BS location and vehicle location

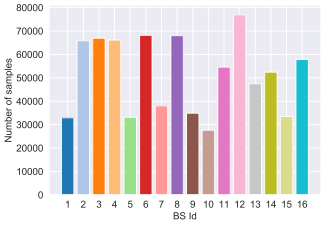
B. Data Description

Table 1 describes All the features considered in our dataset (those we collected, and those we generated). In our proposed algorithms, we use some of these features that we present in Section IV. They cover both vehicles' mobility and network handover decisions. We perform measurements for all vehicles during 24 hours, the total number of vehicles for the selected area is 147 554. Figure 1b shows the evolution of the number of vehicles and their average speed over a day, presented in blue and red lines respectively. The collected dataset consists of 824774 observations, spread over the different BSs, as presented in Figure 2b. The portion of observations related to each BS depends mainly on its location and the traffic density in each area.

The link lifetimes vary from one BS to another, as shown in Figure 3. They mainly depend on the BS coverage (presented in Figure 2a), and the traffic demand in each area. For example, on average, BS3 and BS2 recorded higher values compared to BS9 and BS13. The main reason is that they cover a larger area.

In summary, our created dataset contains V2I links with different lifetimes enriched with mobility information recorded from a mobile network with various kinds of BS and mobility patterns. The complete dataset is available in [24].

(a) BS coverage



(b) Data Proportion by BS

Fig. 2: BS coverage and data proportion by BS

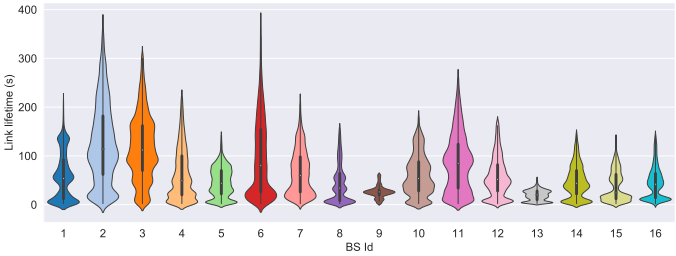


Fig. 3: violin plot of link lifetime per BS (without outliers)

IV. Proposed Approach

In order to estimate the V2I links lifetime, we propose a Machine Learning based approach that allows each BS to learn the link lifetime variation. In particular, we propose a lightweight approach in which the information collected are only the vehicle's location and its speed during the association request, as shown in Figure 4. These information are coupled with others in order to allow the BS to estimate the link lifetime. The model design is detailed in the next section. The second part of our work consists in estimating the next cell to which a vehicle will associate. We adopt a centralized approach with the intention of taking advantage of the global view available at the controller. Furthermore, the ML model is executed by the SDN Controller, which uses mainly some historical data of each vehicle (e.g. previous cell) so that it is possible to infer the next most probable cell.

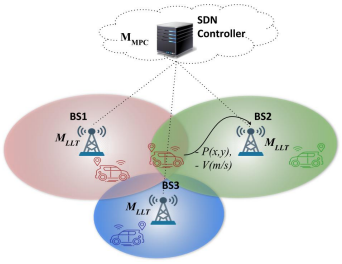


Fig. 4: Conceptual view of the proposed approach

Figure 4 outlines the key elements of the proposed approach. We assume that all vehicles are equipped with a GPS module and send additional information (location and speed) during

the association request. In the case of LTE networks, this information can be sent using Measurement Reports (sent by a UE using the UL-DCCH messages).

In addition, we also assume that each BS sends vehicle information (and the estimated LLT) to the SDN controller to estimate its next cell.

A. Supervised Machine Learning

In our study, we consider the Supervised Machine Learning where the training process is done with labeled data. In other words, the model learns from a set of data with both input and output information. Given a set of data D defined by $D\{(x_1, y_1), ...(x_n, y_n)\}$, the goal of the training process is to establish a relation between input X and output $y, y = M(X)$, so that, for the new input data X_n with unknown outputs, the model can predict the corresponding output $\widehat{y_n} = M(X_n)$ with a good accuracy. We distinguish two types of supervised problem: regression problem, when the value to be predicted is a real continuous number, $y \subset R$ and classification problem, when y belongs to a finite set $C = \{1, 2...c\}$ called classes. Several techniques have been proposed in the literature. Each technique has its pros and cons (sensitivity to noise in data (anomalies), training time, resource consumption, etc.). A comparative study is presented in [25] [26]. Furthermore, we consider the so-called ensemble learning techniques [27], which are one of the most popular and powerful supervised algorithms that provide a more generalized model and avoid overfitting. We focus mainly on the Random Forest algorithm [28] both for regression and classification problems.

B. Model Design

As presented above, we first try to estimate link lifetime between a vehicle and its NAP (e.g eNodeB in case of LTE network) then the most probable next cell of a vehicle. Our problem consists of two sub-problems: the first one can be modeled as a regression problem where the target variable is the link lifetime. And, the second as a classification problem where cell_Ids represent the Classes. We detail in the next sections the techniques and features that we considered to design our model.

1) LLT Model (Link Lifetime):

It's trivial that the link lifetime depends on the communication range of a cell. In 4/5G mobile networks, we mainly distinguish between two types of cells: micro and macro cells. These cells are deployed and tuned based on traffic demand and coverage conditions in a given area. Figure 2a shows that in the considered scenario, we have small cells (e.g. BS_10, BS_5) deployed in dense areas and are generally characterized by shorter link lifetime (median around 50s) compared to those with a larger communication range (e.g. BS_3, BS2), with a median link lifetime around 100s (see Figure 3). One of the criteria that also impacts the link lifetime is the distance traveled by a vehicle within the coverage of a given cell as well as its speed, and this depends on the characteristics of the trajectory: size (m), type (arterial, residential ...), as shown in Figure 2a, and depicted in Figure 5. A vehicle can spend more time than another under the coverage of the same cell depending on the shape of the route taken by each vehicle. We use this trajectory type property as a learning variable (feature) to help the model differentiate roads and therefore estimate the link lifetime more accurately. To do that, each vehicle sends its location with the association request. The cell then calculates the distance D (1) and the angle θ (2) used as input in the LLT model, as shown in Figure 5.

$$D = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{1}$$

$$\theta = \arctan\left(\frac{y_2 - y_1}{x_2 - x_1}\right) \tag{2}$$

with (x_1, y_1) and (x_2, y_2) the position coordinates of vehicle and base station

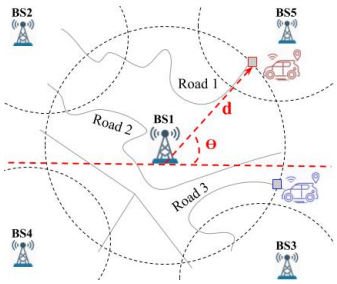


Fig. 5: Road identification as a learning variable

The third aspect that can impact the link lifetime is the road traffic. When roads are more overloaded (e.g. rush hours), the vehicles move more slowly. Consequently, they spend more time under the coverage of a given cell. In order to integrate this aspect in our model, the cell calculates its load based on the number of associated vehicles at a given time. This may help the model to infer the roads occupancy, and therefore, enhance the quality of links' lifetime estimations.

Given that the identified characteristics vary from one cell to another, we decided that each cell has its own model.

Algorithm 1: Link lifetime estimation

Input: *vehicle location* (x_1, y_1), *vehicle speed* v, *Serving_cell location* (x_2, y_2), *historial information* (number of vehicle served by a cell), *Sliding window* T, M_{LLT} (Model obtained from offline training)

Output: $\widehat{llt}$: predicted link lifetime for vehicle v_i

$D = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$

$\theta = atan(\frac{y_2 - y_1}{x_2 - x_1})$

for t in *sliding window* T

    compute cl $(number\ of\ associated\ vehicles)$

endfor

$\widehat{llt}$ = M_{LLT} (D, θ, v, cl)

return $\widehat{llt}$

2) MPC Model (Most Probable Cell):

As for the LLT estimation, the next most probable cell depends mainly on the trajectory traveled by a given vehicle, as shown

in Figure 5, the vehicles passing through the cell1 via roads 1 (e.g. red car) and 2 will have BS2 as the next cell, while vehicles using road 3 will have the BS3 as a next cell.

We consider the same road identification method explained in the LLT model, so we use distance and angle (D, θ) as variables in the MPC model.

One of the points we considered in the design of the MPC model is the last cell to which the vehicle was attached. This will allow the model to distinguish the recurring trips, as shown in Figure 5. Vehicles coming from the area covered by cell 2 and passing through cell 1 tend to go to an area covered by cell 5, while those coming from cell 4 tend to go to an area covered by cell 3. So, we consider the previous cell as a learning variable in our model. To design a model able to make relevant decisions, we also consider the connectivity time with the cell. Given the case schematized in Figure 6, the car covered currently by cell 1 may have three potential next cells (BS 2, 3 and 4). By using only the features presented above the model may not take the relevant decision. For that, we propose to consider the connectivity time with the cell as a learning variable. By doing so, the model may dismiss the BS2 choice as a next cell, if the vehicle is still connected more than T1 (s) and decline the BS3 choice if the connectivity duration is greater than T2.

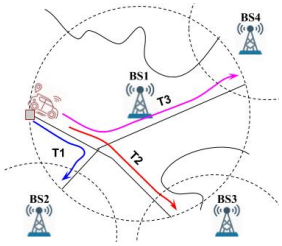


Fig. 6: Previous_cell as a learning variable for MPC Model

Given that one of the main factors impacting the next cell is the trajectory taken by the vehicles, the model infers the trends using the features mentioned above. However these trends may vary depending on the time of day and / or the day of the week, according to the most requested places (e.g. business centers, shopping centers). Therefore, the time of day and day of the week may be integrated into the model in order to deduce the potential variations and improve the prediction accuracy.

V. Performance evaluation

The training is done in two main steps, the first consists of tuning the hyperparameters of our models, while the second aims at learning the model parameters (affected weights to each selected feature). Random Forest has several hyperparameters that can be tuned in order to optimize the prediction accuracy (e.g. the number of trees, the minimum number of samples required to be at a leaf node, etc.). To that end, we used the CV-GridSearch technique, which consists of running the CV K-fold [29] several iterations, each time with different model parameters (specified as input). At the end of this process, the model with best accuracy is used for the second step to train the model in order to learn the feature's parameters. For both models (M_{LLT} and M_{MPC}), we used 75% of dataset for training, the remaining 25% are used for testing, which is a commonly used split ratio. Then, for each entry X_i in the test set X, we compute the output $\widehat{y_i} = M(X_i)$ by using the resulting model M. Then, we compare it with the real value y_i. We thus calculate the prediction accuracy of each model using the performance metrics presented in the following section.

Algorithm 2: Most Probable Cell prediction

Input: $vehicle\ position(x_1, y_1)$, $Serving\_cell\ s\_c$, $Serving\_cell\ location\ (x_2, y_2)$, $historical\ information$ (previous handover decisions of each vehicle), llt estimated link lifetime, M_{MPC} (Model obtained from offline training)

Output: $\widehat{mpc}$: most probable next cell for vehicle v_i

$D = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$

$\theta = atan(\frac{y_2 - y_1}{x_2 - x_1})$

for vehicle v_i

    get the $previous\ cell\ p\_c$

endfor

$\widehat{mpc}$ = M_{MPC} (s_c, D, θ, p_c, llt)

return $\widehat{mpc}$

A. LLT Model

1) Performance metrics and baseline: Two metrics are considered in our evaluation :

- Mean Absolute Error (MAE) represents an average of absolute differences between the predicted and observed link lifetime values. The smaller the value, the better the prediction. It is calculated as follows :

$$\frac{1}{N}\sum_{i=1}^{N}|y_i - \widehat{y_i}| \tag{3}$$

  where $\widehat{y_i}$ is the predicted value.
- Coefficient of determination R^2 represents the percentage of how much our model is better than a simple baseline (prediction values is the mean value of link lifetime). A value closer to 1 means a good model. it is calculated as follows :

$$1 - \frac{\sum_{i=1}^{N}(y_i - \widehat{y_i})^2}{\sum_{i=1}^{N}(y_i - \overline{y_i})^2} \tag{4}$$

  where $\overline{y_i}$ is the mean of link lifetime values, $\widehat{y_i}$ is the predicted value.

2) Results: The LLT model learns primarily from the main features (D, theta and speed). In order to evaluate the impact of the additional feature "cell_load", we evaluate two models: the first one (M_{LLT1}) with only the main features, and the second one (M_{LLT2}) with cell load as an additional feature (calculated using a window T =100 s).

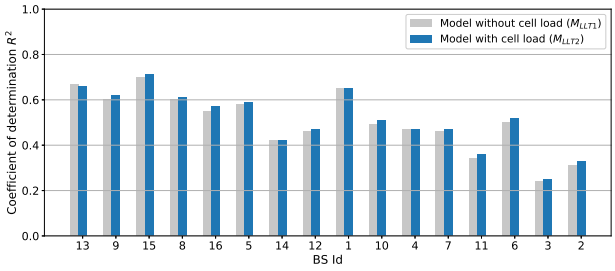


Fig. 7: LLT Model performance (R^2)

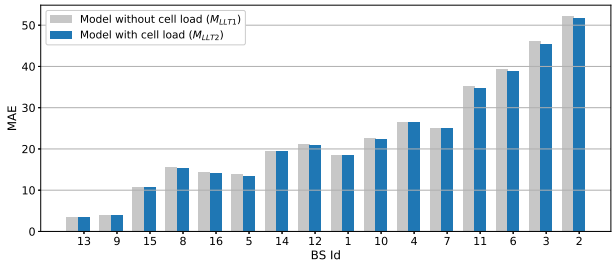


Fig. 8: LLT Model performance (MAE)

Figures 7 and 8 show the performance results of the proposed model. They respectively show the Coefficient of determination R^2 and the Mean Absolute Error.

The x-axis of Figures 7 and 8 represents the BSs sorted in the ascending order (from left to right) of the average link lifetime. Cell 13 has the smallest average of 19.13 s, while cell 2 has the highest average, which is equal to 125.66 s. We can notice that R^2 is mostly between 0.4 and 0.6, with a maximum, around 0.7, for cells with small link lifetimes (i.e. 13, 15), and a minimum, around 0.3, for cells with high link lifetimes (i.e. 3, 2). We can also notice that MAE increases with the average link lifetime. Cell 13 has the minimum MAE of 3.52 s while cell 2 has the maximum MAE, which is equal to 50 s. We can notice that our proposed model performs very well for cells with small (e.g. 13, 9) and medium (e.g. 1, 10) link lifetimes compared to cells with high link lifetimes (e.g. 2, 3). This can be explained by the fact that cells with high link lifetime values generally have a large coverage (as shown in Figure 2a). Consequently, the vehicle may change frequently the path identified at association request.

One possible improvement would be to recompute the LLT prediction using the new vehicle's position after a time interval T (specified according to the service using the prediction). An adaptive interval T is more suitable for cells with high link lifetimes in order to make a trade-off between network overhead and prediction accuracy. This adaptation can be guided by the average observed errors of each location (road). Thus, the model can define a specific interval based on the initial location from where the vehicle is connected to the cell.

We can also notice that both models (M_{LLT1}) and (M_{LLT2}) have slightly the same performance. The use of cell load as a learning feature doesn't greatly improve the prediction accuracy.

B. MPC Model

1) Performance metrics and baseline: We consider the classical metrics used in the literature to evaluate the performance of a classification model. The goal is to analyze the ability of the proposed model to predict vehicle's next cell using the identified features. Firstly, we compute the *Accuracy* which represents the ratio of correct predictions with regard to the total number of input data. It measures the prediction accuracy of the model for all the classes (BS_id). A high value means that the model generally predicts well the next cell for all the classes (BSs). Secondly, we analyze the prediction quality of each class using ROC curves, which represents a plot between the true positive rate TPR (*Sensitivity*) and false positive rate FPR (*1 - Specificity*) of each class. This allows us to identify the classes in which the model performs better (high TPR and low FPR, curve in the top left corner of the plot), and the classes where the model takes more incorrect decisions (lower TPR and higher FPR). Finally, we compute the *Precision* metric of each class in order to analyze the true positive rate with regard to the false positive, which represents the ability of the model to capture the correct cases and do not confuse other items with a given class (e.g. BS1, the model decides that some items belong to class BS1, while, in fact, they're not). Table 2 defines the above-cited metrics.

TABLE II: Performance metrics.

Metrcis		**Formula**
Accuracy		$\frac{TP+TN}{TP+TN+FP+FN}$
Precision		$\frac{TP}{TP+FP}$
ROC	Sensitivity	$\frac{TP}{TP+FN}$ (True Positive Rate)
	Specifity	$\frac{TN}{TN+FP}$ (True Negative Rate) = 1 - FPR (False Positive Rate)

Where TP: True Positives, FN: False Negatives, FP: False Positives, TN: True Negatives

We compare our model to a simple baseline that considers the frequency of going from one cell to another as a metric to predict the next cell. Figure 9 represents the computed frequency for each cell and its neighboring cells. This frequency is calculated using all the observations of the dataset. For example, the frequency of going from cell 1 to cell 2 represents the ratio of the number of samples where the current cell equals to 1 and next cell equals to 2 with respect to the total number of samples where current cell equals to 1. From there, the most probable cell is straightly derived by choosing the one that represents the maximum frequency compared to neighboring cells. For example, the prediction of the next cell for the vehicle connected to cell 1 is cell 3 and for those connected to cell 4 is cell 13.

2) Results: The MPC model learns primarily from the main features (current cell, distance, theta and previous cell). In order to evaluate the impact of the additional feature "LLT", we evaluate two models: the first one (M_{MPC1}) with only the main features, and the second one (M_{MPC2}) with LLT as an additional feature (the evaluations are based on the observed

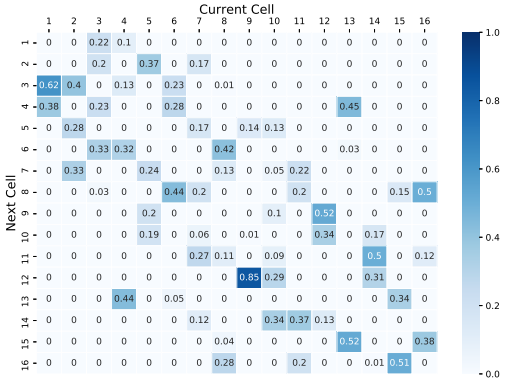


Fig. 9: Neighborhood map with calculated frequencies

LLT values). Figure 10 shows the accuracy of the proposed model compared to the baseline. The model (M_{MPC1}), has a precision of almost 80% compared to the baseline presented above with an accuracy of 40%. In addition, the model reached 86% of accuracy (M_{MPC2}) when we include the additional feature about LLT, presented in Figure 6.

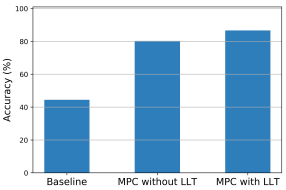


Fig. 10: Prediction accuracy of the proposed model

In order to examine the obtained accuracy, we first analyze the TPR and FPR of each class for each model using the ROC curves, and then the Precision as a second step. Figures 11 (a), 11(b) and 11(c) show respectively the ROC curves for baseline model, M_{MPC1} and M_{MPC2} . The dotted line represents the random model. The ROC curve of a class closer to the upper left corner of the plot (high TPR and low FPR) means that the model correctly predicts the next cell for the vast majority of vehicles heading towards that cell.

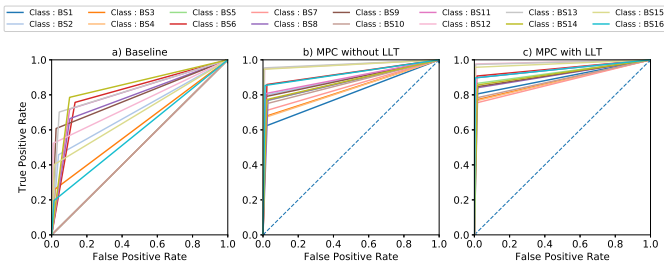


Fig. 11: MPC model performance (ROC metrcis)

We can notice that in our proposed model M_{MPC1} (without LLT), the majority of the classes has a TPR of almost 80%, some reach a TPR around of 95% (as shown in the Figure 11 (b)). This is the case of cells 12, 13, 15, which are surrounded by a reduced number of neighbors (3 neighboring cells, as shown on Figure 9) and they have the specificity of being connected to their neighbors by mostly arterial roads with few intersections (as shown in the Figure 2a). All contribute to reduce the chance that a vehicle leaves the path identified by the model at the association request, hence, helping the model to correctly estimate the next cell. Conversely, cell 1 (ROC blue curve) surrounded by neighboring cells that have a large coverage area with residential roads and many intersections has the lowest TPR (around 62%). In such case, accurate prediction is harder.

The prediction can be however improved by including the LLT metric. Indeed, with M_{MPC2} (as shown in Figure 11 (c)) the majority of the cells have a TPR greater than 83% (some reache 97%). Notably, Cell 1's TPR is improved by 18%. This can be explained by the fact that cell 1 is surrounded by 3 cells with a wide coverage (as shown in Figure 2a), which means that vehicles spend generally more time in neighboring cells to get to cell 1, this helps the model to take the right decisions. The same trends are observed for the precision metric (shown in Figure 12) with a precision of around 80% for the majority of the cells and a high precision for cells 12,13,15 (around 92%) and a precision around 70% for the cells 1, 4, 5. These latter cells are the neighboring cells of the cells 2 and 3 which means that the model confuses the choice of the next cell for vehicles served by cells 2 and 3, by choosing cell 1 as the next cell instead of cells 4 and 5. The precision of the model is also improved by including llt (M_{MPC2}), reaching more than 80% for the majority of cells.

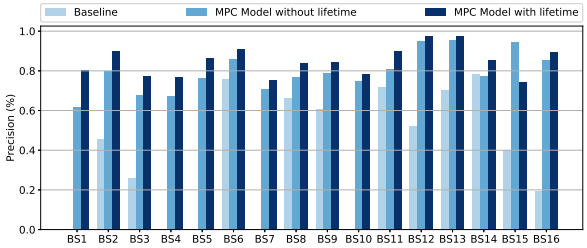


Fig. 12: MPC model performance (Precision metric)

VI. Discussion

Estimating the V2I link lifetime coupled with vehicle's next cell opens the way towards an intelligent and efficient network control. In our proposed approach, we mainly exploited road identification and recurrent trips as the main learning variables in our models, and we made the choice that this happens during the association request with a given cell. Indeed, it does not represent any overload for the network. Performance tests have shown very good results in the vast majority of cases.
The models are trained offline using the data collected during a day. However, the trends captured by the models during the training can vary slightly from one day to another. For example, the roads and places solicited during the business day are different from the weekend. This may influence the learning variables, for example for the same previous cell (p_c)

and given road (d, theta). We then will have a trend to go to a given cell during the business day, and to another cell during the weekend. A training with traces collected over a long duration (e.g. week) including the day of the week as learning variable allows the model to capture these variations. On the other hand, the ISP provider may change the network parameters in order to optimize its network (modifying cell coverage, add or remove a cell). This may impact the performance of the models. A retraining can occur if the prediction error exceeds a given threshold (fixed according to the service using the prediction outputs) with the new data collected and obviously taking into account the new conditions.

VII. Conclusion and future work

In this paper, we have proposed a supervised machine learning method based on random forests to estimate the time duration that a vehicle remains connected to a cell. The main peculiarity of our method is that the prediction is done at association request and only requires the position and speed from the vehicle (both piggybacked with the association request message) and the cell load from the NAP. Based on a dataset derived from real mobility traces from the Luxembourg city that we complemented with network related information derived for the infrastructure of a real cellular network operator in Luxembourg, the performance results show a minimum MAE (Mean Absolute Error) of 3.52 s for cell with narrow coverage and short lifetime values and a maximum MAE of 50 s for cell with wide coverage and higher link lifetime values. We have also proposed a supervised machine learning method to predict the next cell that a vehicle is expected to handover by considering an additional feature: the cell that the vehicle is leaving. Performance results from the above-cited dataset show an accuracy at the level of 80%.

The main perspectives to this work are: Firstly, to investigate new features in order to enhance the accuracy of the proposed models. For example, hour of day and day of the week may help the MPC model to infer the trends of recurring trajectories during the day and the week. Secondly, we plan to test our models performance in different scenario, with various road topology and mobility patterns and finally, to validate our models using a real experimental dataset.

Acknowledgment

This work is funded by Continental Digital Service France (CDSF) in the framework of the eHorizon project.

References

[1] X. Ge, Z. Li, and S. Li, "5g software defined vehicular networks," *IEEE Communications Magazine*, vol. 55, no. 7, pp. 87–93, July 2017.

[2] S. TOUFGA, P. Owezarski, S. Abdellatif, and T. Villemur, "An SDN hybrid architecture for vehicular networks: Application to Intelligent Transport System," in *9th European Congress on Embedded Real Time Software And Systems (ERTS)*, Toulouse, France, Jan. 2018, p. 8p.

[3] C. Campolo, A. Molinaro, A. Iera, and F. Menichella, "5g network slicing for vehicle-to-everything services," *IEEE Wireless Communications*, vol. 24, no. 6, pp. 38–45, Dec 2017.

[4] S. Toufga, S. Abdellatif, P. Owezarski, and T. Villemur, "Openflow based topology discovery service in software defined vehicular networks: limitations and future approaches," in *2018 IEEE Vehicular Networking Conference (VNC)*, Dec 2018, pp. 1–4.

[5] S.-S. Wang and Y.-S. Lin, "Passcar: A passive clustering aided routing protocol for vehicular ad hoc networks," *Computer Communications*, vol. 36, no. 2, pp. 170 – 179, 2013.

[6] M. Hu, Z. Zhong, R. Chen, M. Ni, H. Wu, and C. Chang, "Link duration for infrastructure aided hybrid vehicular ad hoc networks in highway scenarios," in *2014 IEEE Military Communications Conference*, 2014.

[7] T. H. Luan, X. Sherman Shen, and F. Bai, "Integrity-oriented content transmission in highway vehicular ad hoc networks," in *2013 Proceedings IEEE INFOCOM*, April 2013, pp. 2562–2570.

[8] S. Shelly and A. V. Babu, "Analysis of link life time in vehicular ad hoc networks for free-flow traffic state," *Wireless Personal Communications*, vol. 75, no. 1, pp. 81–102, Mar 2014.

[9] X. Wang, C. Wang, G. Cui, and Q. Yang, "Practical link duration prediction model in vehicular ad hoc networks," *Int. J. Distrib. Sen. Netw.*, vol. 2015, pp. 2:2–2:2, Jan. 2015.

[10] X. Wang, C. Wang, G. Cui, Q. Yang, and X. Zhang, "Eldp: Extended link duration prediction model for vehicular networks," *IJDSN*, 2016.

[11] X. Yan, P. Dong, X. Du, T. Zheng, J. Sun, and M. Guizani, "Improving flow delivery with link available time prediction in software-defined high-speed vehicular networks," *Computer Networks*, vol. 145, 2018.

[12] N. Alsharif, K. Aldubaikhy, and X. S. Shen, "Link duration estimation using neural networks based mobility prediction in vehicular networks," in *2016 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*, May 2016, pp. 1–4.

[13] J. Zhang, M. Ren, H. Labiod, and L. Khoukhi, "Link duration prediction in vanets via adaboost," in *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, Dec 2017, pp. 1–6.

[14] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119 – 139, 1997.

[15] H. Ge, X. Wen, W. Zheng, Z. Lu, and B. Wang, "A history-based handover prediction for lte systems," in *2009 International Symposium on Computer Network and Multimedia Technology*, Jan 2009, pp. 1–4.

[16] M. Daoui, A. M'zoughi, M. Lalam, M. Belkadi, and R. Aoudjit, "Mobility prediction based on an ant system," *Computer Communications*, vol. 31, no. 14, pp. 3090 – 3097, 2008.

[17] X. Chen, F. Mériaux, and S. Valentin, "Predicting a user's next cell with supervised learning based on channel states," in *2013 IEEE 14th Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, June 2013, pp. 36–40.

[18] C. Sommer, R. German, and F. Dressler, "Bidirectionally coupled network and road traffic simulation for improved ivc analysis," *IEEE Transactions on Mobile Computing*, vol. 10, no. 1, pp. 3–15, Jan 2011.

[19] A. Varga and R. Hornig, "An overview of the omnet++ simulation environment," in *Proceedings of the 1st International Conference on Simulation Tools and Techniques for Communications, Networks and Systems & Workshops*, 2008, pp. 60:1–60:10.

[20] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker, "Recent development and applications of sumo - simulation of urban mobility," 2012.

[21] L. Codeca, R. Frank, and T. Engel, "Luxembourg sumo traffic (lust) scenario: 24 hours of mobility for vehicular networking research," in *2015 IEEE Vehicular Networking Conference (VNC)*, Dec 2015.

[22] T. Derrmann, S. Faye, R. Frank, and T. Engel, "Poster: Lust-lte: A simulation package for pervasive vehicular connectivity," in *2016 IEEE Vehicular Networking Conference (VNC)*, Dec 2016, pp. 1–2.

[23] A. Virdis, G. Stea, and G. Nardini, "Simulating lte/lte-advanced networks with simulte," in *Simulation and Modeling Methodologies, Technologies and Applications*, 2015, pp. 83–105.

[24] "Llt dataset," accessed: 2019-09-15. [Online]. Available: www.bitbucket.org/soufian_toufga/linklifetimedataset/

[25] S. B. Kotsiantis, "Supervised machine learning: A review of classification techniques," in *Informatica*, 2007, pp. 249–382.

[26] P. V. Klaine, M. A. Imran, O. Onireti, and R. D. Souza, "A survey of machine learning techniques applied to self-organizing cellular networks," *IEEE Communications Surveys Tutorials*, vol. 19, no. 4, pp. 2392–2431, Fourthquarter 2017.

[27] O. Sagi and L. Rokach, "Ensemble learning: A survey," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 8, no. 4, p. e1249, 2018.

[28] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, Oct 2001.

[29] S. Arlot and A. Celisse, "A survey of cross-validation procedures for model selection," *Statist. Surv.*, vol. 4, pp. 40–79, 2010.