Enhanced Urban Clustering in VANETs Using Online Machine Learning

Ghada H. Alsuhli[1], Ahmed Khattab[1], Yasmine A. Fahmy[1], Yehia Massoud[2]
[1]Electronics and Electrical Communications Engineering Department – Cairo University, Giza, Egypt
[2]School of Systems and Enterprises – Stevens Institute of Technology, Hoboken, NJ, USA
gada.sy@hotmail.com, akhattab@ieee.org, yfahmy@cws-cufe.org, ymassoud@stevens.edu

***Abstract*—Clustering in Vehicular Ad-Hoc Networks (VANETs) is essential to mitigate different challenges and meet the required quality of communications. However, most of the available clustering protocols were designed for highways, and thus become unstable in realistic urban environments with many intersections. In this paper, a Clustering Adaptation Near Intersection (CANI) approach is proposed to ensure clustering stability at intersections. This approach exploits Online Sequential Extreme Learning Machine (OS-ELM) to predict the behavior of the vehicles near an intersection and adapt the clusters accordingly. The main advantage of the developed OS-ELM prediction model is its ability to continuously learn and update in real time. After being validated, the proposed adaptation approach is included in a highway clustering scheme. The resultant clustering protocol is compared to other schemes in a realistic urban environment, and shows significant stability and efficiency performance improvement.**

***Index Terms*—VANET; clustering algorithm; urban environment; machine learning**

I. Introduction

Vehicular Ad-Hoc Networks (VANETs) are the core of Intelligent Transportation Systems (ITS) that aim to provide road users with safety and infotainment. These goals cannot be achieved unless a stable VANET communication is guaranteed. However, the highly dynamic and rapidly changing topology of VANET, in addition to the unbounded number of nodes in such networks, compromise this stability and impede the roll-out of its applications. Efficient and stable clustering presents itself as a promising solution of such problems. Grouping the vehicles in clusters with one representative, called the cluster head (CH), allows the transformation of a flat network structure into a hierarchical one, and hence overcoming the scalability issue. Furthermore, when the vehicles with relatively same mobility characteristics are clustered together, the effect of dynamic topology is mitigated.

Unfortunately, the performance of clustering in VANET is highly affected by the environment and structure of the road [1]. For instance, a stable clustering scheme designed for highways loses its stability in urban scenarios due to the presence of intersections [2]. At an intersection, as the cluster members (CMs) have different turning directions, a few members stay connected to the CH, while others become unclustered [3]. Re-clustering after crossing the intersections considerably increases the number of clusters in the system in addition to the clustering packet overhead [4]. Consequentially, more stable and efficient clustering can be achieved if the clustering protocol has the ability to adapt its behavior near the intersection and prepare the cluster for such crossing to mitigate the re-clustering effects.

In this paper, we propose a machine learning-based approach to adapt clustering near intersections, called Clustering Adaptation Near Intersection (CANI). CANI allows a highway clustering protocol to effectively suit urban environment. This approach exploits the speed and efficiency of Online Sequential Extreme Learning Machines (OS-ELM) to predict the behavior of a vehicle at the upcoming intersection and adapt the algorithm accordingly. The proposed OS-ELM has the ability to learn incrementally in real time using different features extracted from vehicle mobility and road structure. After validating the OS-ELM model, the proposed adaptation approach was included in a recent highway clustering scheme. Then, the improvement in stability and efficiency of the resulting clustering scheme were verified by comparing its performance to other clustering approaches on realistic urban environment.

The remainder of this paper is organized as follows. In Section II, we review the related literature. The proposed CANI approach is presented in Section III. The performance evaluation and results analysis are discussed in Section IV. Finally, Section V concludes the paper.

II. Related Work

During the past decade, many clustering protocols for VANETs were proposed. The existing clustering schemes in the literature usually target either urban or highway environments. As clustering in highways is less complicated, most of the proposed clustering algorithms were designed and evaluated for simple multi-lanes highway scenarios [5]–[8].

Clustering in urban scenarios is much more challenging as the cluster will be scattered once passing the first intersection. To increase the stability of the clusters and mitigate the intersections effects, a few techniques were suggested. The work in [9] used road ID matching as a main metric to produce stable clusters. This choice mainly prevents the clusters coming from different roads from merging temporarily at the intersection and thus prolongs the CH lifetime. For the same purpose, the direction angle was used to distinguish the clusters coming from differer roads near the intersection by grouping only vehicles with acute direction angle [10]. Although the false merge at intersection was mitigated using

978-1-7281-3473-4/19/$31.00 ©2019 IEEE

these two approaches, the problem of unclustered vehicles resulting after crossing the intersection was not handled.

Some urban clustering schemes relied on the assumption that each vehicle is totally aware of its trajectory to the destination, as in [11]–[14]. This awareness was used to cluster the vehicles according to the short-term or long-term similarity of these trajectories. This assumption was justified by the availability of a digital map [13], GPS system [11], or based on the driver's interest [12]. Despite the considerable stability improvement using this technique, it assumes that the turn direction at the next intersection is definitely known without errors, which is not always a realistic situation. Moreover, the knowledge of vehicles trajectories was mainly used as a metric to form the clusters which is initially done. However, as this metric was not considered in the cluster maintenance phase, the problem of cluster scattering at coming intersections over time still exists.

A few clustering protocols contained a kind of prediction of vehicle behavior before intersections to suit urban scenarios [15], [16]. This prediction was basically done by considering the lane number that a vehicle occupy. These works proposed that a vehicle on the left/right lane is more probably to turn left/right at the intersection, whereas vehicles in the middle lanes continue straight. Then, the CH is selected from a direction with the largest number of vehicles to increase the cluster life. In our work, a more sophisticated prediction is adopted, using machine learning, based on various features in addition to the lane number. Furthermore, this prediction affects all clustering stages, not only the CH selection in our approach.

Using machine learning for prior prediction of vehicle state at roads intersection is popular in the literature [17], [18]. This kind of prediction was used to enhance some VANET protocols and applications as in [19], [20]. The authors in [19] used k-means clustering to enable a vehicle to predict its turn direction at the next intersection. The predicted direction was then used to look up destinations in the proposed routing scheme. The same goal was achieved in [20] using decision tree to predict the driver behavior and vehicle state which is necessary to estimate the reliable communication time between vehicles in the Internet of Vehicles (IoV). In the same direction, a machine learning model is developed and used in this paper to predict the moving direction of a vehicle at intersections, which is utilized to increase the lifespan of clusters and clustering efficiency in urban scenarios.

III. Proposed CANI Approach

Vehicle-to-Vehicle (V2V) VANET environment is considered in our clustering adaptation near intersection approach. Each vehicle is assumed to be equipped with a wireless transceiver unit to communicate with other vehicles, a Global Positioning System (GPS) to determine the vehicle position, speed and direction, and a digital map to extract some features related to the coming intersection.

Although the proposed CANI approach may be integrated with any VANET clustering protocol, it will be explained and evaluated in this paper in the context of the Double Head clustering (DHC) scheme [21], without loss of generality. In this section, we review the DHC clustering protocol to ease the understanding of how the clustering is adapted in response to an intersection proximity. Then, an overview of the adopted OS-ELM turning prediction model is introduced. Finally, the different phases of CANI approach will be explained in detail.

A. DHC Clustering Protocol

DHC is a recently developed general purpose clustering algorithm. The clustering process of DHC has a number of procedural steps. First, it begins with exploring the neighbors and collecting their information. During this step, the vehicle eligibility is calculated and broadcasted. This eligibility is estimated based on different mobility and link quality metrics. Eligibility values are then updated and used in the next clustering steps, such as CH selection and cluster maintenance. During the cluster maintenance step, DHC proposed several schemes to enhance the cluster stability after formation, such as a cluster replacement scheme. Cluster replacement allows the CM to optionally replace the current cluster when the link with the corresponding CH is threatened, and there is a CH in the range with better interplay link life time. For more details about DHC, one can see [21].

B. Proposed OS-ELM Turning Prediction Model

In this section, some basic concepts of OS-ELM is briefly introduced to facilitate the understanding of OS-ELM sequential training. Then, the adopted OS-ELM turning prediction model is explained.

1) OS-ELM Overview: OS-ELM was originally proposed in [22] as a development of Extreme Learning Machine (ELM). ELM is a feed-forward neural network with a single hidden layer (SLFN), shown in Fig. 1, which is recognized by its good performance with a very high learning speed. In ELM, batch learning is used, i.e. all data samples are available for training at once given the dataset $\{(\boldsymbol{x}_i, \boldsymbol{y}_i) \;:\; \boldsymbol{x}_i \in \mathbb{R}^n,\; \boldsymbol{y}_i \in \mathbb{R}^m\}_{i=1}^{N}$ where $\boldsymbol{x}_i$ is the vector of features and $\boldsymbol{y}_i$ is the target vector. These N samples can be approximated by SLFN with d hidden layers such that:

$$f_d(\boldsymbol{x}_i) = \sum_{j=1}^{d} \boldsymbol{\beta}_j \boldsymbol{G}(\boldsymbol{\alpha}_j, b_j, \boldsymbol{x}_i),\; i = 1, 2, ..., N, \tag{1}$$

where $\boldsymbol{\alpha}_j$, b_j and $\boldsymbol{\beta}_j$ are the input weight, bias and output weight of the hidden nodes, respectively, and G is an activation function.

The training procedure of OS-ELM is done using two stages: (i) initialization stage, and (ii) sequential learning stage. In the initialization stage, N_0 samples of the training data is used to train the initial model, $\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^{N_0}$, such that $N_0 \geq d$. In this stage, the input weight $\boldsymbol{\alpha}_j$ and the bias b_j are randomly assigned. Then, the initial hidden layer output matrix $\boldsymbol{H}_0$ is calculated:

$$\boldsymbol{H}_0 = \begin{bmatrix} G(\boldsymbol{\alpha}_1, b_1, \boldsymbol{x}_1) & ... & G(\boldsymbol{\alpha}_d, b_d, \boldsymbol{x}_1) \\ \vdots & \ddots & \vdots \\ G(\boldsymbol{\alpha}_1, b_1, \boldsymbol{x}_{N_0}) & ... & G(\boldsymbol{\alpha}_d, b_d, \boldsymbol{x}_{N_0}) \end{bmatrix} \tag{2}$$

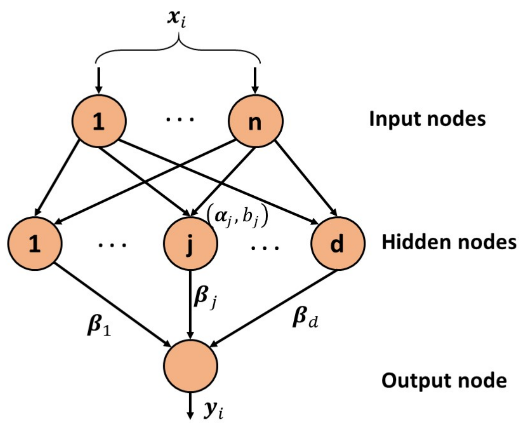

Fig. 1. ELM architecture

Therefore, the problem is equivalent to the minimization of $||\boldsymbol{H}_0\boldsymbol{\beta} - \boldsymbol{Y}_0||$, where $\boldsymbol{Y}_0 = [\boldsymbol{y}_1, ..., \boldsymbol{y}_{N_0}]$. According to [22], the optimum $\boldsymbol{\beta}^{(0)}$ is estimated:

$$\boldsymbol{\beta}^{(0)} = \boldsymbol{M}_0\boldsymbol{H}_0^T\boldsymbol{Y}_0 \tag{3}$$

where $\boldsymbol{M}_0 = (\boldsymbol{H}_0^T\boldsymbol{H}_0)^{-1}$.

In the sequential learning stage, the samples arrive chunk by chunk. The k^{th} chunk is represented by $\left\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\right\}_{i=(\sum_{n=0}^{k-1} N_n)+1}^{\sum_{n=0}^{k} N_n}$, where N_n is the number of samples in the n^{th} chunk. During the sequential learning stage, the hidden layer output matrix $\boldsymbol{H}_k$ is computed first.

$$\boldsymbol{H}_k = \begin{bmatrix} G(\boldsymbol{\alpha}_1, b_1, \boldsymbol{x}_{(\sum_{n=0}^{k-1} N_n)+1}) & \dots & G(\boldsymbol{\alpha}_d, b_d, \boldsymbol{x}_{(\sum_{n=0}^{k-1} N_n)+1}) \\ \vdots & \ddots & \vdots \\ G(\boldsymbol{\alpha}_1, b_1, \boldsymbol{x}_{\sum_{n=0}^{k} N_n}) & \dots & G(\boldsymbol{\alpha}_d, b_d, \boldsymbol{x}_{\sum_{n=0}^{k} N_n}) \end{bmatrix} \tag{4}$$

Second, the output weight $\boldsymbol{\beta}^{(k)}$ is calculated.

$$\boldsymbol{\beta}^{(k)} = \boldsymbol{\beta}^{(k-1)} + \boldsymbol{M}_k\boldsymbol{H}_k^T(\boldsymbol{Y}_k - \boldsymbol{H}_k\boldsymbol{\beta}^{(k-1)}) \tag{5}$$

$$\boldsymbol{M}_k = \boldsymbol{M}_{k-1} - \boldsymbol{M}_{k-1}\boldsymbol{H}_k^T(\boldsymbol{I} + \boldsymbol{H}_k\boldsymbol{M}_{k-1}\boldsymbol{H}_k^T)^{-1}\boldsymbol{H}_k\boldsymbol{M}_{k-1} \tag{6}$$

$$\boldsymbol{Y}_k = [\boldsymbol{y}_{(\sum_{n=0}^{k-1} N_n)+1}, ..., \boldsymbol{y}_{\sum_{n=0}^{k} N_n}] \tag{7}$$

The procedure of sequential learning is repeated every time a new chunk arrives. We observe in Equation (5) that $\boldsymbol{\beta}^{(k)}$ is computed based on the matrices calculated for the previous chunk in addition to the current chunk samples $\boldsymbol{Y}_k$. Thus, OS-ELM has the ability to learn the data samples chunk-by-chunk (or even one-by-one). These chunks of data may have the same or different sizes (as in our case). Moreover, the previously trained data observations can be discarded once the training is done. This reduces the burden of storing the overall training dataset. These advantages, in addition to its frequent use in solving several recent engineering issues [23], [24], make OS-ELM a good choice for our prediction model requirements.

2) OS-ELM for Turning Prediction: The adopted OS-ELM model considers predicting the movement of a vehicle at the next intersection as a multi-class classification problem. We consider that a vehicle can take one of a three actions at the next intersection: turning left, turning right or going straight. To predict the action of a vehicle using OS-ELM, six features are adopted: (1) lane number, (2) distance to intersection, (3) vehicle speed, (4) traffic light existence on the intersection, (5) legal turning directions at current intersection branch, and (6) turning indicators' signals (blinkers). The first five features are commonly used in similar prediction issues [18], while the sixth is an original contribution of this paper. In our approach, the vehicles are supposed to be capable of collecting these features using a localization system (features (1) and (2)), a digital map (features (2), (4) and (5)), or a proper sensor to sense the binary signal of the blinkers in the current vehicle.

C. CANI Approach Phases

Clustering adaptation in response to a vehicle vicinity to an intersection passes through three main phases in the proposed CANI approach. The first phase is the offline initialization phase, which results in the initially trained OS-ELM model. Then during clustering process, the second phase takes place to prepare the clusters for passing the intersection when the vehicle draws near an intersection. The final phase happens after the vehicle leaves the intersection, then the OS-ELM model is updated.

1) Offline Initialization Phase: This phase takes place before the clustering procedure starts. In this phase, the first stage of OS-ELM is executed to build up the initial model of turning prediction. Thereafter, this model is exported to all vehicles in the system. The used data samples for the initial training can be realistic or collected via simulation, as this model will be sequentially trained later in the targeted environment. After starting the clustering process, each vehicle calculates the distance from the coming intersection during its journey in the road. When this distance becomes less than a predefined distance R (see Fig. 2), the clustering preparation phase begins.

2) Clustering Preparation Phase: Near the intersection, the vehicle begins to periodically acquire different features and predict its turning direction (donated as the vehicle state) at this intersection using OS-ELM up-to-date model. This state is exchanged within the hello message between all vehicles in this phase. Next, the required action is only a conformation of the available clusters since the re-clustering here compromises rather than benefits the stability of the formed clusters. This conformation: (1) avoids joining a cluster with different state, (2) rejects any CM joining request if it has different state at the next intersection, (3) encourages the CM that will leave its cluster after the intersection to replace it with another cluster with identical state (if any), (4) ensures that the clusters with different states never merge. This conformation is simply done

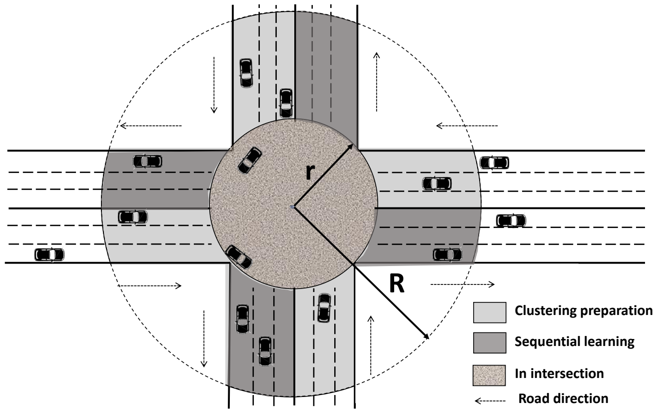

Fig. 2. CANI phases mapped to an intersection layout

for DHC protocol by dropping the received eligibility of a nieghbor with different state to ’0’. This is because merging, cluster replacement and joining a cluster all are affected by the eligibility value of the candidate vehicle.

This phase continuous until the distance from intersection is less than r.

3) Sequential Learning Phase: After leaving an intersection, the vehicle knows the actual direction that it has already turned, by measuring the turning angle.

Hence, the feature samples that have been collected near the intersection in the clustering preparation phase are now labeled. This chunk of data is used to revise and update the OS-ELM model, for more prudent prediction in the future. Consequentially, the more the crossed intersections by a vehicle, the better the accuracy of the prediction becomes. Moreover, using this approach each vehicle builds up a special model that commensurate with the nature of the usually crossed intersections.

It is worth mentioning that, if the prediction was wrong, the vehicle continues as a member of the current cluster until the link with the inconsistent CH is lost, or until another consistent CH is found.

IV. Performance Evaluation

In this section, the proposed OS-ELM prediction model is discussed and validated. For this purpose, a dataset was built to train and test this model. After validation, the turning prediction model was exported to DHC. Hence, DHC becomes suitable for urban as well as highway environment, donated UH-DHC. The performance of UH-DHC is then evaluated and compared with other clustering protocols.

A. OS-ELM Parameter Selection and Validating

In order to validate the ability of the proposed model to predict the movement of a vehicle at the next intersection, a simulation-based dataset was built. The reason behind building the dataset instead of using an existing real one, is that none of the available dataset contains the blinker signals, which is proposed to be used as a turning prediction feature in this paper.

For the aim of building the dataset, different intersections with various shapes were selected from United States urban area using Open Street Map (OSM). Realistic vehicle mobility was inserted in these intersections and simulated using Simulation of Urban MObility (SUMO). Some of these intersections are illustrated in Fig. 3. During SUMO simulation, the considered features are recorded using TRAffic Control Interface (TRACI) and a dataset of 3600 samples was built.

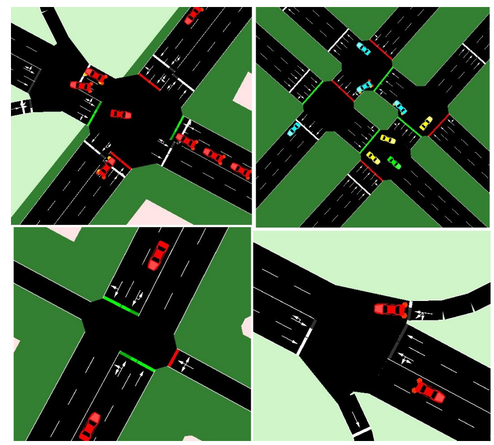

Fig. 3. Some of the simulated intersections to build the dataset

Two factors mainly affect the performance of OS-ELM, the activation function and the number of hidden nodes. Hence, a 10-fold cross-validation method was used to select these parameters and validate the model. Nine folds used for initialization stage and the tenth used for testing and then learning sequentially. Figure 4 shows the testing accuracy when four different activation functions (Sigmoid, Sin, RBF and Hardlim), and different number of hidden nodes are used. We observe that the RBF activation function has the worst performance under the extracted data. In addition, when the number of hidden nodes is less than 400, the Sigmoid function outperforms the other functions. Whereas, the Hardlim and Sin functions perform better with higher number of hidden nodes. Considering that increasing the number of hidden nodes considerably increases the training time, the Sigmoid function and 350 hidden nodes are adopted. Then, the initial prediction accuracy is about 80% for the developed model. For the same selected parameters, this accuracy drops to 75% when Batch ELM instead of OS-ELM is used.

B. UH-DHC Performance Evaluation

To show the effectiveness of our approach, UH-DHC is compared with other recent clustering schemes, on a realistic urban environment, using different evaluation metrics.

1) Simulation Environment: In order to assess the performance of the compared algorithms in a realistic VANET environment, a real 1 km$\times$2 km urban area was exported from Manhattan (USA) map using OSM, Fig. 5. This area was edited by SUMO and a realistic vehicle traffics were inserted randomly all over the available roads.

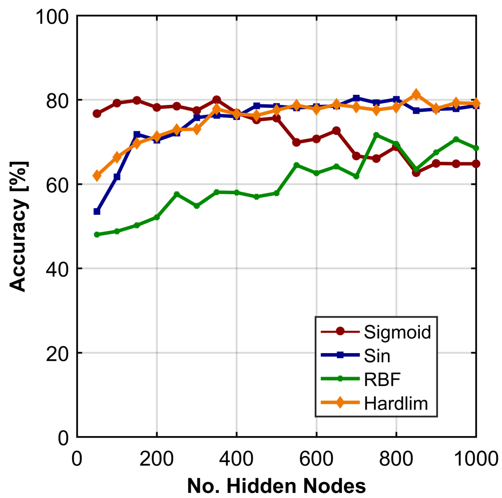

Fig. 4. Testing accuracy with different activation functions and number of hidden nodes

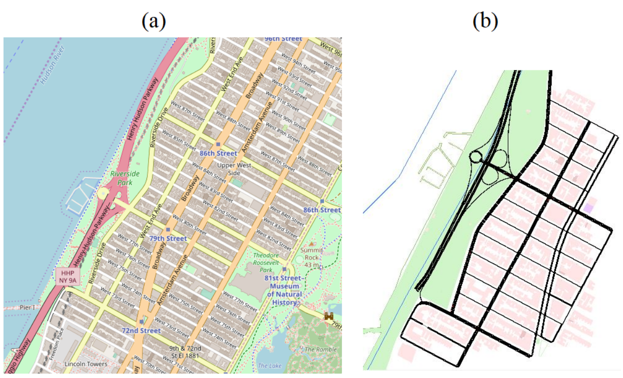

Fig. 5. Simulated Manhattan urban area. (a) In OSM. (b) In SUMO

The compared algorithms were implemented on Network Simulator NS3.26 with the parameters listed in Table I.

2) Evaluation Metrics: Five evaluation metrics are used to evaluate the compared algorithms: Cluster Head Lifetime (CHL), CM Life-Time (CML), Control Packet Overhead (CPO), number of clusters generated in the network during simulation, and state transitions per vehicle. More informations about these metrics are available in [21], .

TABLE I
NS3 SIMULATION PARAMETERS

Parameter	Value
NS3 version	3.26
Simulation time	100 s
Tx range	200 m
MAC protocol	IEEE 802.11p
Propagation Model	Two-Ray Ground
No. vehicles	25, 50, 75, 100
R	25 m
r	12 m

3) Performance Comparison Results: The performance of UH-DHC algorithm is compared with two recent clustering approaches: the original clustering protocol DHC, and a Unified framework for clustering (UFC) [25] in terms of the previously mentioned metrics. For fair compassion, all common parameters were set to the same values, which are equivalent to DHC default parameters.

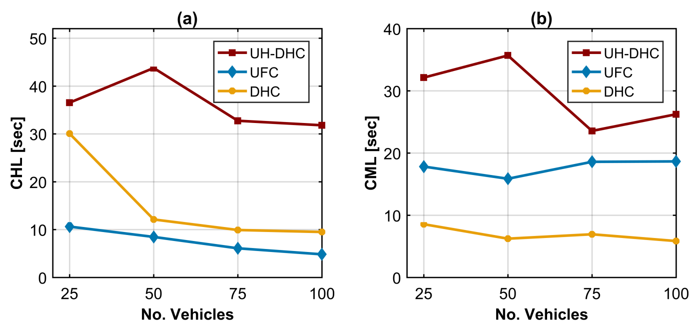

Fig. 6. CHL and CML comparison between algorithms. (a) CH lifetime. (b) CM lifetime

Figure 6(a) illustrates the compassion results of the CHL when the three algorithms are used, with different vehicle densities. We observe that UH-DHC has the highest CHL for all vehicle densities and thus better clustering stability. The improvement of stability becomes clearer when the number of vehicles in the simulation increases. This is because splitting larger clusters on the intersections significantly affects the stability in absence of clustering adaptation.

In Fig. 6(b), the CML in UH-DHC is significantly improved compared to DHC. This can be interpreted by the fact that a vehicle will not join a cluster with different state on the next intersection. At the same time, the cluster replacement near the intersection allows th CM to join a cluster that is more probably to connect longer.

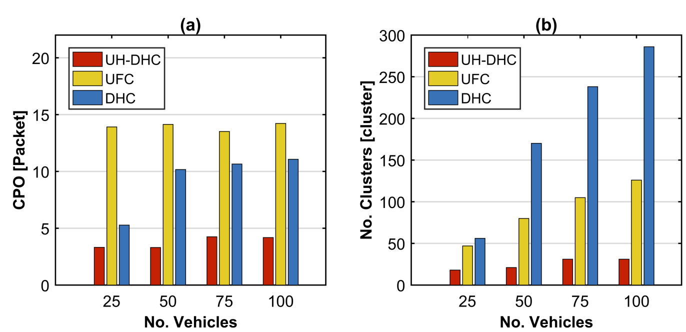

Fig. 7. CPO and number of clusters comparison between algorithms. (a) Control packet overhead. (b) No. clusters

The clustering packet overheads caused by the three algorithms are shown in Fig. 7(a). The CANI approach used in UH-DHC noticeably mitigates the clustering overhead caused by the mandatory re-clustering near intersections. Indeed, UH-DHC reduces the CPO by 73% and 61%, on average, compared to UFC and DHC, respectively.

Figure 7(b) shows the number of clusters generated during simulation when different number of vehicles are simulated. Since the splitting and re-clustering after the intersection causes more clusters to be generated, DHC and UFC, suffers from generating large number of clusters, in contrast to UH-DHC. For example, in presence of 100 vehicles in the simulation, UH-DHC reduces the number of clusters from 286 in DHC and 126 in UFC to only 31 clusters.

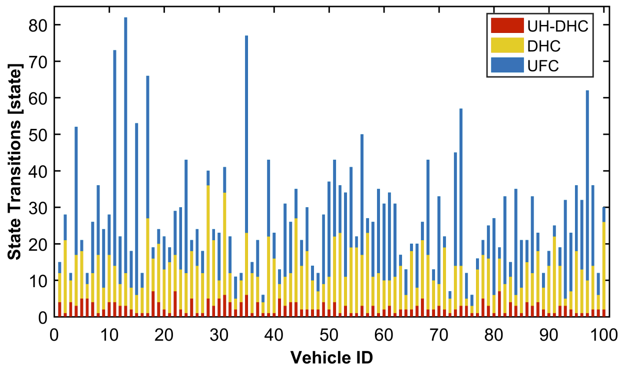

Fig. 8. Number of state transitions comparison between algorithms

To show the ability of our approach to stabilize the clustering, Fig. 8 depicts the state transitions of 100 vehicles during simulation time. The lowest number of transitions is caused by UH-DHC then DHC and finally UFC. For instance, vehicle 50 transits to 9 and 26 states during simulation for DHC and UFC, respectively. This is compared with only 2 transitions caused by UH-DHC.

V. Conclusion

In this paper, we have presented a machine learning-based CANI approach to maintain the clustering stability by adapting the clustering behavior at next intersection. An OS-ELM machine learning model is developed to provide fast and continuous learning and thus allow the vehicle to accurately predict its behavior at the upcoming intersection. The proposed CANI approach is merged with the DHC clustering scheme to got a UH-DHC protocol that suits both urban and highway environments. The experimental analysis in realistic urban shows that UH-DHC enhances urban clustering performance in terms of stability and efficiency compared to other recent clustering approaches.

References

[1] C. Cooper, D. Franklin, M. Ros, F. Safaei, and M. Abolhasan, "A comparative survey of VANET clustering techniques," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 1, pp. 657–681, 2017.

[2] Y. Huo, Y. Liu, L. Ma, X. Cheng, and T. Jing, "An enhanced low overhead and stable clustering scheme for crossroads in VANETs," *EURASIP Journal on Wireless Communications and Networking*, vol. 2016, no. 1, p. 74, 2016.

[3] Y. Huo, Y. Liu, X. Xing, X. Cheng, L. Ma, and T. Jing, "A low overhead and stable clustering scheme for crossroads in VANETs," in *Wireless Algorithms, Systems, and Applications*, K. Xu and H. Zhu, Eds. Cham: Springer International Publishing, 2015, pp. 232–242.

[4] Y. Huo, W. Dong, J. Qian, and T. Jing, "Coalition game-based clustering strategy for crossroads in vanets," in *International Conference on Identification, Information and Knowledge in the Internet of Things (IIKI)*, 2016.

[5] M. Zhang, C. Li, T. Guo, and Y. Fu, "Cluster-based content download and forwarding scheme for highway VANETs," *China Communications*, vol. 15, no. 4, pp. 110–120, 2018.

[6] X. Cheng and B. Huang, "A center-based secure and stable clustering algorithm for VANETs on highways," *Wireless Communications and Mobile Computing*, vol. 2019, 2019.

[7] X. Cheng, B. Huang, and W. Cheng, "Stable clustering for VANETs on highways," *IEEE/ACM Symposium on Edge Computing (SEC)*, 2018.

[8] Z. Khan, P. Fan, S. Fang, and F. Abbas, "An unsupervised cluster-based VANET-oriented evolving graph (CVoEG) model and associated reliable routing scheme," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–16, 2019.

[9] M. Hadded, P. Muhlethaler, R. Zagrouba, A. Laouiti, and L. A. Saidane, "Using road IDs to enhance clustering in vehicular ad hoc networks," in *International Wireless Communications and Mobile Computing Conference (IWCMC)*, 2015.

[10] M. Hadded, P. Muhlethaler, A. Laouiti, and L. A. Saidane, "A novel angle-based clustering algorithm for vehicular ad hoc networks," in *Vehicular Ad-Hoc Networks for Smart Cities*. Springer, 2017, pp. 27–38.

[11] M. M. C. Morales, C. S. Hong, and Y.-C. Bang, "An adaptable mobility-aware clustering algorithm in vehicular networks," in *Asia-Pacific Network Operations and Management Symposium*, 2011.

[12] I. Tal and G.-M. Muntean, "User-oriented fuzzy logic-based clustering scheme for vehicular ad-hoc networks," *in Vehicular Technology Conference (VTC Spring)*, 2013.

[13] Y. Zhou, X. Wu, and P. Wang, "A novel vehicle clustering algorithm in intersection scenario," in *International Conference on Advanced Infocomm Technology (ICAIT)*, 2017.

[14] H. Shahwani, T. D. Bui, J. P. Jeong, and J. Shin, "A stable clustering algorithm based on affinity propagation for VANETs," in *International Conference on Advanced Communication Technology (ICACT)*, 2017.

[15] M. S. Almalag and M. C. Weigle, "Using traffic flow for cluster formation in vehicular ad-hoc networks," *IEEE Workshop On User MObility and VEhicular Networks*, 2010.

[16] A. Mehmood, A. Khanan, A. H. H. M. Mohamed, S. Mahfooz, H. Song, and S. Abdullah, "ANTSC: an intelligent naïve bayesian probabilistic estimation practice for traffic flow to form stable clustering in VANET," *IEEE Access*, vol. 6, pp. 4452–4461, 2018.

[17] A. Zyner, S. Worrall, and E. Nebot, "Naturalistic driver intention and path prediction using recurrent neural networks," 2018.

[18] D. J. Phillips, T. A. Wheeler, and M. J. Kochenderfer, "Generalizable intention prediction of human drivers at intersections," in *IEEE Intelligent Vehicles Symposium (IV)*, 2017.

[19] W. K. Lai, M.-T. Lin, and Y.-H. Yang, "A machine learning system for routing decision-making in urban vehicular ad hoc networks," *International Journal of Distributed Sensor Networks*, vol. 2015, 2015.

[20] L. Liu, C. Chen, T. Qiu, M. Zhang, S. Li, and B. Zhou, "A data dissemination scheme based on clustering and probabilistic broadcasting in VANETs," *Vehicular Communications*, vol. 13, pp. 78–88, 2018.

[21] G. H. Alsuhli, A. Khattab, and Y. A. Fahmy, "Double-head clustering for resilient VANETs," *Wireless Communications and Mobile Computing*, vol. 2019, p. 17, 2019.

[22] N. Liang, G. Huang, P. Saratchandran, and N. Sundararajan, "A fast and accurate online sequential learning algorithm for feedforward networks," *IEEE Transactions on Neural Networks*, vol. 17, no. 6, pp. 1411–1423, Nov 2006.

[23] F. Lu, J. Wu, J. Huang, and X. Qiu, "Aircraft engine degradation prognostics based on logistic regression and novel OS-ELM algorithm," *Aerospace Science and Technology*, vol. 84, pp. 661 – 671, 2019.

[24] Y. Li, R. Qiu, and S. Jing, "Intrusion detection system using online sequence extreme learning machine (OS-ELM) in advanced metering infrastructure of smart grid," *PLOS ONE*, vol. 13, no. 2, pp. 1–16, 02 2018.

[25] M. Ren, J. Zhang, L. Khoukhi, H. Labiod, and V. Vèque, "A unified framework of clustering approach in vehicular ad hoc networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 5, pp. 1401–1414, 2018.