Studies on Vehicular Communications in Cellular Systems Employing Machine Learnings

機械学習を用いたセルラーシステムにおける車用通信に関する研究

February 2021

Takashi KOSHIMIZU

コシミズ　タカシ

Studies on Vehicular Communications in Cellular Systems Employing Machine Learnings

機械学習を用いたセルラーシステムにおける車用通信に関する研究

February 2021

Waseda University

Graduate School of Fundamental Science and Engineering,

Department of Computer Science and Communications Engineering, Research on Wireless Access

Takashi KOSHIMIZU

コシミズ　タカシ

Abstract

The impacts and needs of cellular communication have been increasing more than ever with the recent success of 5G system launches. Specifically, the application demands for vehicular communications can be recognized as the cross-industry needs. Various automotive-related industries have started to collaborate with the cellular industries with a target to make driving system much safer, much smarter toward the final goal of achieving machine-maneuvered autonomous driving. This thesis titled "Studies on Vehicular Communications in Cellular Systems Employing Machine Learnings" proposes an improved and smarter integrated communication system that target to harmonize the operation between cellular and vehicular networks. Cotemporary vehicles have begun to be equipped with various advanced sensors and monitors are now evolving into Smart Internet of Vehicles (SIV) with always-on connectivity to the Internet. These highly capable sensors monitoring visual and radio images are poised to replace humans in making decisions on driving maneuvers. Distributed computing over virtualized network systems gives a strong momentum to the realization of the above targets by providing spreading computational resources over the network as additional resources to vehicles in leveraging the virtualized network's flexibility. As researchers in the related fields have also stated, without applying the power of Machine Learning (ML), the lofty target would not be realized in various aspects.

With an aim to contribute to achieving such objectives, this work provides a set of studies on sophisticated Vehicular Ad Hoc Networks (VANET) clustering and dynamic network resource acquisition schemes employing the power of ML. A 3GPP-specified vehicular communication system called Cellular-V2X (C-V2X) is an additional powerful vehicular communication platform, which provides both short-ranged communication capability and long-ranged cellular communication capability simultaneously in a single communication system. We apply this C-V2X platform in our study and propose an improved time-critical VANET clustering scheme by leveraging these capabilities.

The next prime contribution of this thesis is to propose a sophisticated scheme for computational resource identification for nomadic mobile vehicles. However, the resources and status of network also dynamically change as resources are consumed and allocated for other requests as well. Therefore, how to find an appropriate resource acquisition is a complex question as it is dynamic and not static. In order to answer this question, especially in view of the dynamic status of network resources, we employed the power of ML through this work.

Having these objectives, Chapter 2 reviews related fundamentals of vehicular clustering schemes and ML theories. As a sophisticated object clustering algorithm, we introduce Affinity Propagation Clustering (APC) which bases on the factor-graph theory, of which concept was originally shown in *Nature*. We applied and enhanced the tool in our VANET clustering scheme. Gazis-Herman-Rothery (GHR) car following model is an additional theory we reviewed in the chapter, as we obtained real traffic data from an observation. The scheme has been quite useful for preparing a set of organized data for simulation analysis. We also conducted thoughtful review of variation of ML algorithms in this part. Specifically, Support Vector Machine Classification (SVMC), Gradient Boosting Machine Classification (GBMC) and eXtream Gradient Boosting Machine Classification (XGBC) are carefully explained in the chapter for foundation of understanding forthcoming proposal part.

Chapter 3 introduces our first field of study, which focuses on an enhanced clustering proposal leveraging APC theory to VANET clustering. Our proposed scheme extends the original concept of APC and is designed to leverage the power of C-V2X wireless communication platform. This is a challenging task, as the cluster has to be formed while vehicles are in dynamic motion and need to meet the traffic characteristics. We carefully designed a key mechanism called *similarity function* in the APC process. This chapter includes extensive sets of simulation-analysis conducted to evaluate the proposed scheme.

Chapter 4 extends the above study for finding an ideal VANET clustering *granularity* and autonomous formulation applying the power of ML. Finding an ideal clustering granularity has been an essential task and was the only remaining task in the previous study. We focused on the fundamental evaluation criteria, which are identified as three essential terms: traffic density, communication volume, and network congestion status. These selected criteria are used for the evaluation for ML decision making. We also designed a message sequence and procedure that fit into the C-V2X communication platform, targeting the ease of implementation. Our proposed scheme perfectly works in application-level enhancements and in a distributed controlled manner. This study applied Support Vector Machine Classification algorithm as the ML classification. We carefully analyzed the fundamental mechanism and examined the parameters with tuning in the simulation. Extensive clustering results were produced and reviewed during the simulation process. For the performance evaluation, ML prediction accuracy as well as contribution of 5G access latency were also displayed in the simulation.

Chapter 5 shows the study results on dynamic server selection employing ML for 5G VANET.

The volume and size of vehicular computation has been increasing with the decision made by ML through the sensing data obtained from high-defined sensors. In order to replace human operation, various data have to be analyzed thorough ML enable decision-making on behalf of humans. As the computation power of a single vehicle is limited, having a distributed computation aid has become an essential approach. Therefore, we have introduced a study of distributed computation on the vehicular communication. There are a number of criteria that have to be evaluated, specifically in time critical dynamic environments. Hence, we applied the power of ML in this study as well. We selected an essential set of criteria for ML decision with a target to minimize computation load and improve prediction accuracies. We also proposed a mechanism that adapts to time criticality. ML needs a learning time to increase its prediction accuracy, their sampling volume and processing time are linearly related. In order to solve this dilemma, we proposed an isolation of the learning process from the execution process in the study. The work also proposed placing a ML agent on the network edge, which priorly acquires components of a necessary task vector in a distributed manner. Therefore, the ML execution process can be isolated from the lengthy learning process. With the simulation results, we examined the prediction performance of ML. We also examined a missing data scenario case in the ML prediction, as all sets of necessary data cannot be realistic, specifically in consideration of time-critical vehicular application scenarios. In this study, newly introduced ML algorithms GBMC and XGBC are examined in addition to SVMC. We carefully reviewed the mechanism and evaluated the performance over these algorithms. Latency contribution in our scheme and three server model were also evaluated in the simulation process.

Finally, Chapter 6 concludes these studies by reflecting the observations from the outcomes. Throughout this thesis we learned quite a large number of identifications in theorical aspect as well as simulation aspects. We presume the results can be applicable to a wide span of moving machine type communication objects. This chapter also provides potential future work and on-going activities of application of ML in 5GS and related communication fields.

Contents

List of Figures

List of Tables

Chapter 1

1 Introduction

The era of the 5th Generation cellular System (5GS) has opened, reflecting the significant efforts in the 3GPP standardizations [01]-[04] as well as the huge effort of in commercial implementations. As of October 2020, 5G was operated in more than 100 global commercial services and reached more than 5% of global population [05]. Having the highly demanding requirements [01], the 5GS has started to provide much higher speed, much lower latency and a much larger number of connections than human beings have ever made on cellular systems. Having various advanced features embedded, 5GS applications are spreading to various industries and extending the range of services on a scale never seen before. Some examples are as follows. Enhanced Mobile Broadband (eMBB) opens a door for displaying a hologram and 3D imaging on a high-resolution display. Ultra-Reliable and Low-Latency Communications (URLLC) will open up new services requiring time-critical and low-error communication, expected to support such applications as medical surgery by remotely located doctors. Massive Machine Type Communications (mMTC) enables large number of connecting objects such as those of MTC and Smart Internet of Things (SIoT) via the cellular system. Conventionally, a majority of cellular communications have been made through handsets operated by human beings. However, in the 5G era and onwards such machine objects will be additional main users.

Vehicular communication is one of the typical and predictable MTC examples which typically require all eMBB, URLLC and eMTC capabilities of 5GS. Level 5 autonomous driving allows complete machine operation of vehicular maneuver with spontaneous critical decision making in time-limited events in replacement of human beings who have ever maneuvered. This is a challenging target; however, it is not a dream but a strong industrial requirement. These messages can be seen from intensive, industry-wide collaborations such as 5GAA (5G Automobile Industry Association) [06]-[07]. This is a cross-industry consortia that typically takes a leading role in driving and solving industry-wide common objectives. Autonomous vehicle control will largely change our way of life as well as a way of product logistics, once it is realized. In addition, the applicability of our studies to vehicles is just an example. Another example of the forthcoming mobile MTC will be robots which can replace humans in carrying out labor-intensive work on roads and in cities. Advanced drones will be another mobile MTC application, which transport objects and it is fair to imagine they will change the future of logistics via sky transportations.

Intelligent decision making is an essential capability targeted at autonomous driving and those SIVs. As our conventional work has been shifted to the always-on style with the emergence of the Internet. Cloud services, web meetings, remote works are all realized by on-line connectivity to the Internet. Specifically, SIVs make their decision through communicating various remote entities located on the Internet. SIVs are equipped with a series of sensors for the sake of conducting autonomous maneuver. Quasi-Zenith Satellites [08] provide much finer precision of vehicular GPS while, Cellular-V2X PC5 [09]-[11] direct connection provides V2V high-speed connection. Laser imaging Detection Ranging (LIDAR) [12] senses proximity images in 3D environments with laser sensors as if human beings were sensing the visional information via visible light waves. Once information is sensed, they are processed and decision making are performed by ML intelligence, which tends to require large computational resources.

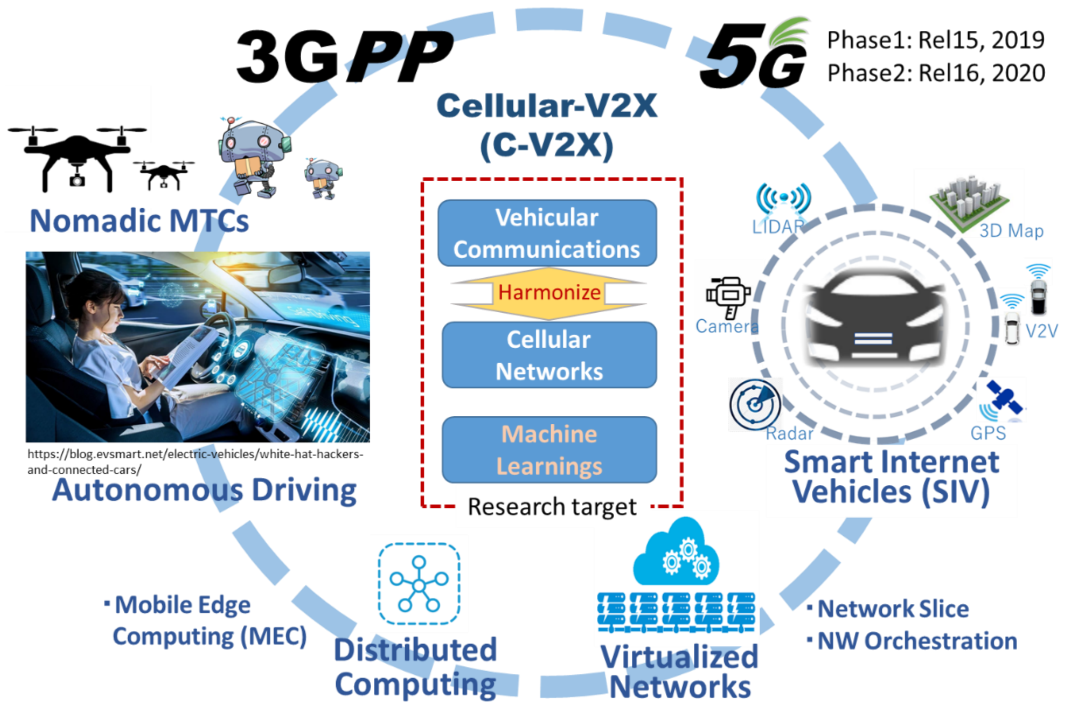


Fig.1.1 Research Backgrounds and Related Technologies

Distributed computing much flexibly allows such intensive computation spreading over the network and allows the required computation to be used in the best place. A platform of virtualized network opens the access to resources and also minimizes the operational cost. Figure 1.1 displays the overall image of backgrounds and technologies related to this thesis. We explain each part in the following section.

1.1 Research Background

1.1.1 5G Mobile Communication System

The first standardization sets for 5th Generation cellular system have been published from 3GPP in 2019 as Release 15. These sets are called 5G System (5GS) Phase 1 specification according to the report of GSMA. The fundamental mechanism of 5GS has been designed to meet industry demanded requirements with a target to support enhanced mobile broad band that provides large data rates, ultra-reliable low-latency communication that provides much lower error rates and massive machine type communication that allows a much larger number of connections compared to the previous generation cellular systems.

1.1.2 Cellular V2X Communication System

There have been various vehicular communication technologies studied. Cotemporary wireless technology for vehicular communication has been specified in 3GPP called Cellular-V2X [09]-[11]. C-V2X represents performances that provide improved connectivity from vehicle to anything, including vehicles-to-vehicles (V2V), connectivity to infrastructures (V2I), as well as interworking to pedestrians (V2P). Therefore, C-V2X has communication capability not only for short-ranged wireless communication as the conventional IEEE 8.2.11p-based Dedicated Short Ranged Communication (DSRC) but also for long-ranged communication to a cellular network.

The demand of improved vehicle radio technology has been increasing from a wide range of industrial needs, as the work in a single industry only will not be able to realize a complete set of implementable distributed vehicular communication system. Various sensing technologies, smart vehicle navigation systems and improved V2X communication technologies are needed as a set of a system. C-V2X is designed along with such a wide range of demands and requirements.

eMBB, URLLC and eMTC are the key design targets of 5GS and advanced vehicular communication exactly needs their enhanced performance. Image processing and 3D Map analysis need a much wider data channel. Low latency capability is critical for real-time application in nomadic vehicular communication. mMTC represents a much large number of moving MTC objects in the future. Therefore, 5GS and C-V2X can be considered to be an ideal combination and how to integrate their advanced use-cases should be further considered. We presumed that vehicle clustering and vehicle distributed computation in 5GS are best study subjects so, we set them as our study scope and pursued in this thesis.

1.1.3 Vehicular Ad hoc Network and Clustering

Clustered vehicles form a network called a vehicular ad hoc network (VANET), which connects its members through proximity wireless links. Such short-ranged inter vehicular wireless communication supports the formation of VANETs and the function has several advantages. First advantage is the isolation of local communication from global communication enabled by VANET inside communication via local short-range radio links. It largely contributes to improving system stability by segmenting local communications from global cellular communications. Next advantage is the creation of advanced services from inter-VANET short-range communications. Targeting to autonomous and advanced assisted driving, the driving speed, precise location, and vehicles' maneuver information are essential data need to be exchanged for safer and accident-free autonomous driving control.

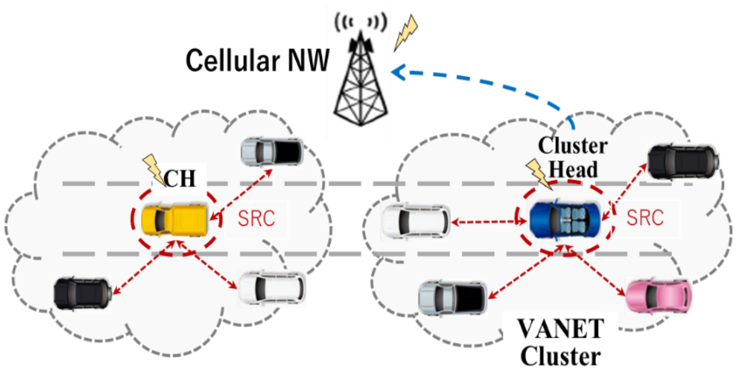


Fig.1.2 Vehicular Ad Hoc Networks (VANET) and Global communications

1.1.4 Virtualized Cloud and Distributed System

Network virtualization is a typical trend which can be seen in cotemporally network system-design and in cellular network function as well. The main motivation for network virtualization is to maximize network resource utilization and best resource allocation among various users through virtualized systems. This functional example can be typically observed via the network slice capability in 5GS. As the network slice concept offers desired network capabilities in a virtualized platform, it is not necessary to prepare an individual hardware system-set for each of the specific purposes, while conventional deployments have to do so. For example, an eMBB network can be prepared based on the requirements and the needs on demand. This is also applicable to URLLC as well as mMTC networks, which are formed on demand based on the needs. Their functional capability is made based on the request and can also be reformed once the need arises. From a cost perspective, this on-demand capability substantially reduces the operational cost of implementors.

Operation and Maintenance (O&M) in cellular systems has greatly advanced by cooperating with power of virtualized system design. Network orchestration is designed to achieve zero-touch operation, which eliminates complex labor-intensive, human-manipulated network maintenances. This is realized with various collaborations such as ETSI Network Functions Virtualization (NFV) project as well as open-source activity in Open Network Automation Platform (ONAP).

Distributed computing is a scheme of computation designed to integrate discrete computing powers into a single computational occasion through a unified process over the network entity via coordination message exchange [13]. In advanced vehicular communications such as SIV, this capability enables various new service offerings by performing a large-scale computation by utilizing additional resources. The URLLC capability in 5G largely contributes to achieving the above target as if the computational resources were located in the next hop. ML, object recognition, and natural language comprehension are examples of applications necessitating large-volume computational resources. Therefore, distributed learning mechanism and distributed computing capability are essential functionalities to achieve objectives of advanced vehicular services mentioned so far.

Mobile Edge Computing (MEC) is a typical realization of distributed computing in cotemporally cellular systems. In MEC, required functions are located on the network edge with the intension to minimize access latency and provide additional resources to the mobile entities. In our proposed scheme, we also utilized this capability in placing ML agent onto the network edge.

1.2 Challenges and Motivations

In realizing vehicular communications in 5GS applying smart clustering and ML schemes, we have the following challenges with associated motivations.

1.2.1 VANET Clustering Applying Affinity Propagation

The first challenge is to propose a scalable and sophisticated VANET cluster formation in conjunction with C-V2X. The first motivation for this challenge is to apply a sophisticated VANET clustering algorithm. While conventional studies have been proposing case-by-case clustering schemes and self-suitable approaches, we apply a mathematically well-stable and therefore extendable clustering approach, which is rooted in the Affinity Propagation Clustering (APC) concept.

The second motivation is to propose a distributed and scalable system through the set of the proposals. As the motion of vehicles is nomadic and spontaneous, we designed and proposed the scheme that not oriented toward a centralized control approach, but in the way able to work according to a distributed and self-organized controlled principle. Additional advantage given from clustering formation is system scalability. One-to-many connectivity to a cellular system causes control congestion, as the numbers of connected MTC objects increases on the demand. Although the communication volume from such MTC has been known to be small however, the communication frequency tends to be large. Therefore, a VANET cluster largely contributes to system stability by using a Cluster Head (CH) as a network proxy.

The third motivation is to design the proposed scheme adaptable to time critical and nomadic vehicular applications. Therefore, the proposed system needs to be carefully tuned for minimizing the communication latency and the communication frequency.

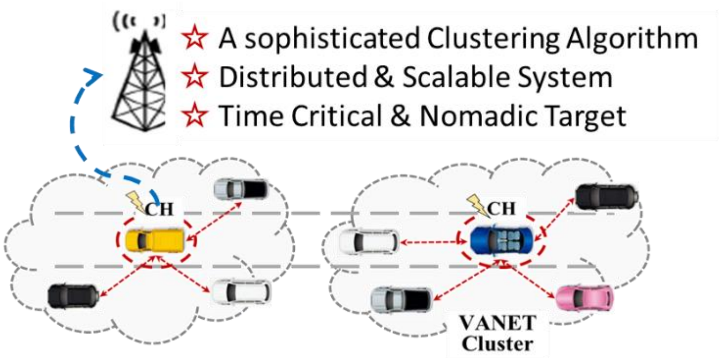


Fig.1.3 Challenges on VANET Clustering

1.2.2 Autonomous VANET Clustering Applying Machine Learning

The next challenge is autonomous VANET cluster formulation applying ML. So far, our study was not reached to find an ideal VANET granularity, the first motivation is to derive an ideal clustering size knowing vehicular dynamics. The second motivation in this challenge is targeted to design a scheme having the capability of autonomous clustering with the existing C-V2X procedure integrated to the maximum extent. This motivation widens the chance of implementing the proposed scheme and, the use of C-V2X procedure will make its realization quite stable. The third motivation is high adaptability and extendibility of the proposed scheme to other purpose, once we find further improvements and extensions. Remarkably, this can easily conduct as we are using a ML approach. It simply updates the algorithm and decision criteria, then we can achieve another application without not re-constructing entire mechanism. This reflects the ease of extendibility of the proposed approach.

Fig.1.4 Challenges on Autonomous VANET Clustering

1.2.3 Dynamic Resource Acquisition Applying Machine Learning

The third challenge we explore is dynamic network resource acquisition from mobile vehicles considering the network dynamics. As a network resource, we tried to find an appropriate computational resource *i.e.*, server from alternatives. Conventionally, these server selections have been done in static circumstances, however these resources over the network are constantly changing as they are continuously consumed for various users. Therefore, selecting specific resources in a fixed scenario is unrealistic. As such resource identification must be a dynamic scenario in real a use case, and we leveraged the power of ML to solve the issue.

This capability allows and eases future application of distributed vehicular computation. This is an essential demand as the computational power of vehicle communication will be extremely large in the future as we are motivating autonomous driving aiming to replace the current human maneuver. For realizing such a lofty goal, distributed vehicular computation is indispensable functionality in forthcoming vehicular networks.

In providing such capability even aided by ML, time criticality is another consideration point as the target vehicles are dynamically mobile. Therefore, taking a specific approach must be essential. We considered an additional time minimization scheme in the proposal while applying ML.

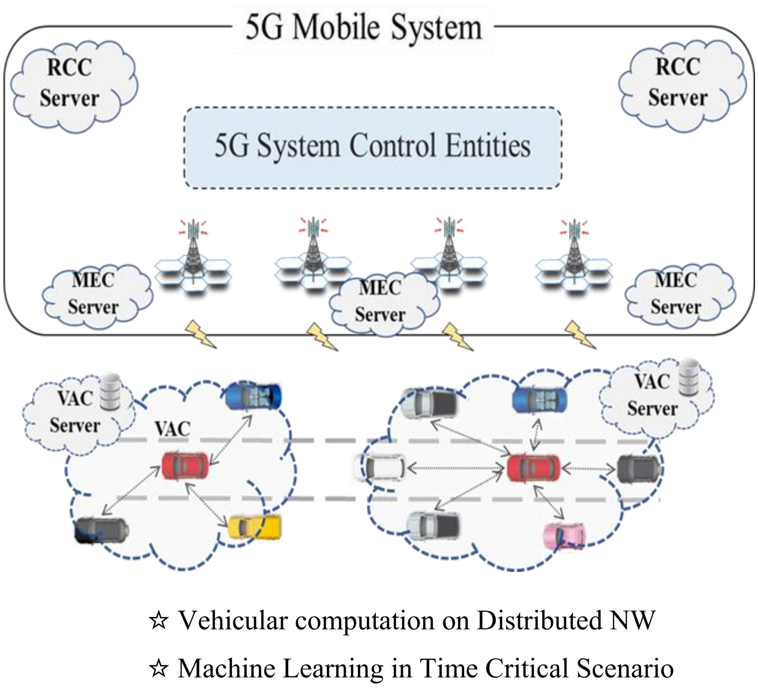


Fig.1.5 Challenges on Dynamic Cloud Server Selection

1.3 Contributions

The key contributions of this thesis comprise three key components. The first contribution is to elaborate on a sophisticated VANET clustering applying the APC theory. The second contribution is to study autonomous VANET cluster formation applying a ML scheme. The third contribution is to study the discovery of desirable computational resources knowing network dynamics applying a ML scheme. Each of them is described in the following part.

Chapter 3 studies VANET clustering applying APC. This study targets at designing a distributed clustering scheme and obtaining system scalability thorough the proxy and concentration effect via identified cluster heads. The proposed function has been designed to be applicable to time critical and monadic mobile targets. The main contributions on this part can be summarized as follows;

- The proposed mechanism is designed easy to implement yet adaptable to vehicles' motion dynamism as they are nomadic in principle.

- The proposed scheme leverages the power of C-V2X communication property for improved performance in message exchanging frequency among the members.
- The scheme based on a reliable and mathematically well-defined procedure called affinity preparation in enhancing the similarity-function.
- It designed to fit distributed control in consideration of nomadic behavior of the targets and for avoiding a single point of failure.

Chapter 4 studies autonomous VANET cluster formation applying ML. This study extends the previous work in applying ML for deducing ideal cluster granularity and the formation from the selected criteria. The scheme and procedure largely rely on the fundamental mechanism of C-V2X. This makes the scheme highly adaptable to C-V2X application and greatly eases its implementation. The advantage of applying ML is its expendability in changing components of the task vector and the evaluation criteria where this mechanism can be easily extended to another needs. The main contributions on this part can be listed as follows:

- The proposed scheme called multi-dimensional affinity propagation enables autonomous formation of VANET clusters adaptable to vehicular motion dynamics.
- Specifically, we employed a scheme of ML for deducing clustering granularities based on the minimum essential criteria.
- For the evaluation criteria, we specifically incorporated the traffic density with the congestion status in Node-B in the decision flow of ML.
- Not only mathematical ideas but we proposed a fully functionable sequence and procedure in extending those of C-V2X in a distributed manner.

Chapter 5 studies to find desirable computational resources from alternatives knowing network dynamics by applying ML. This study intends to realize various and extensive needs of vehicular computation in a distributed manner. Conventional server selection has been performed in static consideration in heuristic approach. In contrast to these static decision makings, this study can be applicable to a situation involving a changing real-time network status. The main contributions on this part can be summarized as follows:

- This work proposes a sophisticated server selection scheme employing ML taking into account the network dynamics with limited evaluation criteria.
- Variety of ML algorithms are applied to analyze the fundamental mechanism for extending the original idea.

- For applying time critical targets, the learning process is designed in a way isolated from the execution process that directly impacts the computation length independent of the sampling numbers.
- Missing data handling procedures are also considered. This capability greatly improves the applicability of the proposed scheme as full data sets are not always obtained practically.

1.4 Thesis Structure

The rest of the thesis consists of the following structure. Chapter 2 explains related fundamentals of the key scheme that we apply, which includes the concept of affinity propagation, car following theory, ML and algorithms applied to understand the comprehensive context of the proposed section. Chapter 3 introduces the context of normalized multi-dimensional parameter based-APC in C-V2X. Chapter 4 proposes and explains an extension of the previous work specifically, by deducing an autonomous cluster and granularity via ML. Chapter 5 introduces a dynamic server selection scheme applying ML for 5G VANET. This specifically examines distributed VANET computation knowing system dynamics in consideration of ML evaluation criteria. Chapter 6 concludes the entire study then, proposes findings and touches upon potential future research scopes.

Chapter 2

2 Fundamentals of Vehicular Clustering and Machine Learning Theory

2.1 Vehicular Clustering

A vehicle cluster consists several members in the group and uses a cluster head as a communication proxy to an external network. The overall taxonomy of clustering techniques for various VANET applications in each category is provided in [14]. An extensive survey conducted on clustering techniques can also be seen in [15], which analyses a variety of VANET clustering techniques aiming to solve problems such as cluster head selection, cluster affiliation, and cluster management.

In centralized clustering schemes, cluster formation is performed via a NB or Rode Side Unit (RSU) with periodical message exchanges to the target vehicles. Qi *et al*. [16] proposed an SDN-based centralized clustering scheme by exploiting a social pattern, *i.e*., knowing the vehicles' traffic pattern to deduce an expected route, where the metrics of inter-vehicular distance, relative speed and vehicle attributes are used. In distributed clustering schemes, cluster formation and CH selection are performed only between the vehicles through message exchanges without involving the management of central node in a fully distributed manner. The centralized clustering approach is advantageous in that it has more computational resources and the managements available from a central entity. On the other hand, from the standpoint of nomadicity of vehicular mobility, the distributed clustering approach has advantages that it can avoid signaling concentration and a single point of failure; specifically, it allows autonomous decision making among widely spanned vehicular targets.

2.1.1 Affinity Propagation Clustering

In contrast to those conventional clustering schemes, an advanced, fully distributed VANET clustering scheme was proposed [17] by Hassanabadi *et al*. [18] and by Shahwani *et al*. [19] based on a mathematically well-proven concept called Affinity Propagation, which was originally published in *Science* by Frey and Dueck [20].

The study applies AP to VANET clustering by assuming that each vehicle is communicating and exchanging messages via inter-vehicle dedicated wireless links. This mutual communication property is an essential condition for applying APC, because the scheme identifies clusters by iteratively passing messages among the nodes, *i.e.*, vehicles. Fundamental idea of APC stems from the concept of factor graph. It tries to find a best set of clusters by recursively exchanging specific messages until a good set of CH (*i.e.*, exemplar) is identified along with the members by measuring the similarities in a specific similarity function. The following outlines the fundamental process of how this scheme works.

In AP, the similarity measured between a pair of data points *k* and *l*, denoted as *s(k, l)*, is evaluated to identify exemplars which will emerge. The process allows any data point to be a possible exemplar that will exchange messages with other exemplars until another set of higher-quality exemplars emerges. In this process, two kinds of messages are exchanged between vehicles *k* and *l*. First, the responsibility message *r(k,l)* is designed to bet sent from vehicle *k* to vehicle *l* as a potential centering point, *i.e.*, an exemplar, which represents a degree of adequateness of vehicle *l* as an exemplar relative to other potential candidate exemplars. The message called responsibility *r(k,l)* and the self-responsibly message *r(l,l)* in each iteration can be shown as

$$r(k,l) \leftarrow s(k,l) - \max_{l' \neq l} \left\{ a\left(k,l'\right) + s\left(k,l'\right) \right\} \quad \forall\, k,l \tag{2.1}$$

$$r(l,l) \leftarrow s(l,l) - \max_{l' \neq l} \left\{ s\left(l,l'\right) \right\} \tag{2.2}$$

Next message called *availability a(k,l)* is instead reversely sent from the candidate exemplar *l* back to source point *k*, showing *k's* suitability of becoming an exemplar for *k*, considering the response from other candidate exemplars. The message of availability *a(k,l)* and the *self-availability* message *a(l,l)* in each iterations can be provided by

$$a(k,l) \leftarrow \min\left\{0, r(l,l) + \sum_{l' \notin \{k,l\}} \max\left\{0, r(k',l)\right\}\right\} \quad \forall\, k,l \tag{2.3}$$

$$a(l,l) \leftarrow \sum_{k' \neq l} \max\left\{0, r(k',l)\right\} \tag{2.4}$$

During the iteration process, these messages are tuned to avoid computational oscillations which would potentially impede convergence. This can be adjusted with the introduction of a tuning factor shown as

$$message_{new} = \lambda \cdot message_{old} + (1-\lambda) \cdot message_{new} \tag{2.5}$$

where λ is a damping factor whose assigned value is 0 ~ 1, while in our simulation, $\lambda = 0.5$ is used as it is a default value [21]. When the message converges, the AP process comes to terminate its iteration. Eventually, a specific data point is identified as a CH within the member, and this is identified when the following condition can be met in the *self-responsibility* and *self-availability* messages as given by

- In a condition k if $r(k, k) + a(k, k) > 0$, where point k is an exemplar, or
- In a condition l if $r(l, l) + a(l, l) > 0$, where point l is an exemplar.

From the results obtained in these responsibilities and availabilities iteration process, exemplars are identified. Upon convergence, each node k's CH is identified by

$$CH_i = \arg\max_{l}\{a(k,l) + r(k,l)\} \tag{2.6}$$

Eventually, the identified CH acts as a gateway for other members in the cluster to the cellular network.

2.1.2 Gazis-Herman-Rothery (GHR) car following model

Review of traffic models is another field we tried to understand the behavior of real vehicle traffic dynamics. These studies have been performed extensively such as in civil engineering and traffic science for analyzing and designing safe and efficient traffic systems. We identified a well-studied and sufficiently proofed model called the GHR vehicle acceleration model, also known as the General Motor's car following model [22]. We used a simulation scheme by referring to the mathematical explanation in [23]. The above-mentioned model is particularly useful as it enables us to obtain vehicles' motion dynamics such as acceleration, velocity and position information. As data acquisition of dynamically moving targets is sometimes limited, we applied the aforementioned GHR car following model. A well-organized reference [23] explains how the GHR model predicts the follower's acceleration *a* using the equation below:

$$a_{n+1}^{t} = \left[\frac{\alpha_{l,m}(v_{n+1}^{t})^{m}}{(x_{n}^{t}-x_{n-1}^{t})^{l}}\right][v_{n}^{t} - v_{n+1}^{t}] \quad (2.7)$$

for the n^{th} vehicle (n = 1, 2, ⋯, N) $N \in \mathbb{N}$ at discreet time t, where l represents a headway exponent, m represents a speed exponent and $\alpha_{l,m}$ is called a sensitivity coefficient of the GHR. This equation is the base of our traffic simulation model. From the derived acceleration, both v_n^t and x_n^t are identified from the Newton's law of motion. According to [22] and [23], we specifically applied the parameters: l = 1.0, m = 0 and $\alpha_{l,m}$= 18 in the GHR model in order to perform simulations.

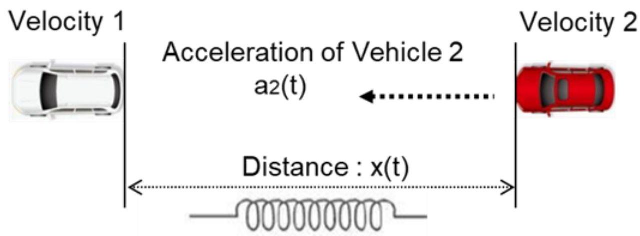


Fig.2.1 GHR Car Following Model

2.2 Machine Learnings

ML enables computers to identify hidden insights through iterative learning of given data sets. Table 2.1 summarizes the variations of ML in terms of categories, objectives, algorithms and application examples in communication fields [24]. In a large perspective, each ML application is classified as either a *supervised* or *unsupervised* learning scheme. In this study, we chose a supervised scheme because the training data are fairly obtained, and it enables us to evaluate the prediction results by comparing with those given from the decision tree. In comparison with SVM, Neural Network (NN) is generally considered more applicable when the *dimension* of the explanatory variable is large. In addition, SVM is known to enables the tuning of prediction accuracy by the limited number of parameters, which are γ and C parameters. Reflecting these properties, we decided to employ SVM to identify the clustering granularity in this study. In general, the process of supervised ML consists of two stages: training and testing. In the training stage, a model is *learned* based on a set of prepared training data. Once the function is educated by the training data, the trained function is applied for providing *predictions* in the testing stage.

The next sub-section explains the details of the task vector and the decision tree we propose, both of which are the essential elements of SVM.

TABLE 2.1: Machine Learning Variation and Application Examples

Category	Objective	Algorithm	Application Ex.
Supervised	Classification	SVMC, GBMC, Neural Network,	Anomaly detection, Server selection
	Regression	Logistic Regression, SVR	Throughput prediction, channel parameter regression
Unsupervised	Clustering	K-means, Spectrum clustering	Congestion control, hierarchical routing
	Dimension Reduction	Manifold learning	Data aggregation

2.2.1 Support Vector Machine Classification

This section explains fundamentals of the SMV Classification scheme.

1) Mechanism of soft-margin-based SVM

The fundamental concept of SVM originates from a neural model applying a linear-threshold logical unit, which eventually provides class discrimination. When a set of training data is given in $(\boldsymbol{X}_i, y_i), \cdots, (\boldsymbol{X}_M, y_M),\ 1 \leq i \leq M,\ \boldsymbol{X} \in R^d, y \in \{\pm 1\},\ M \in \mathbb{N}$, with the corresponding correct class-labels y_M, the linear discriminate function can be denoted as

$$y = f(\boldsymbol{X}) = sign[(\boldsymbol{W}^T \boldsymbol{X} - h)] \tag{2.8}$$

$\boldsymbol{X}$ is a task-vector consisting of explanatory variables. $\boldsymbol{W}$ is the corresponding synapse weight-factor, and h represent a threshold value. This model provides an output $y \in \{\pm 1\}$. It returns +1 when the inner product of the task vector $\boldsymbol{X}$ and weight-factor $\boldsymbol{W}^T$ exceeds the threshold, and it returns -1 when it is less than the threshold. Geometrically, this concept is applicable as a classification system of input data, which segments data into two fields as depicted in Fig. 2.2(a). The figure shows two types of class-regions R_1 and R_2. They are distinguished by the corresponding hyper planes $H_1\colon \boldsymbol{W}^T \boldsymbol{X} - h = 1$ and $H_2\colon \boldsymbol{W}^T \boldsymbol{X} - h = -1$, which are formed by a limited number of double-circled support vectors. Notably, no data exists within the margin-area formed by the two hyper planes. An optimal hyperplane H^* can be identified by maximizing the margin between the hyperplanes, which can be provided in minimizing $1/\| \boldsymbol{W} \|$, and which is an equal to maximizing the part of $\| \boldsymbol{W} \|^2$. Therefore, this can be re-written to an optimized question given by

$$\min\ L(W) = \frac{1}{2} \| \boldsymbol{W} \|^2$$
$$s.t. \quad y_i(\boldsymbol{W}^T \boldsymbol{X}_i - h) \geq 1 \quad (i = 1, 2, \cdots, M) \tag{2.9}$$

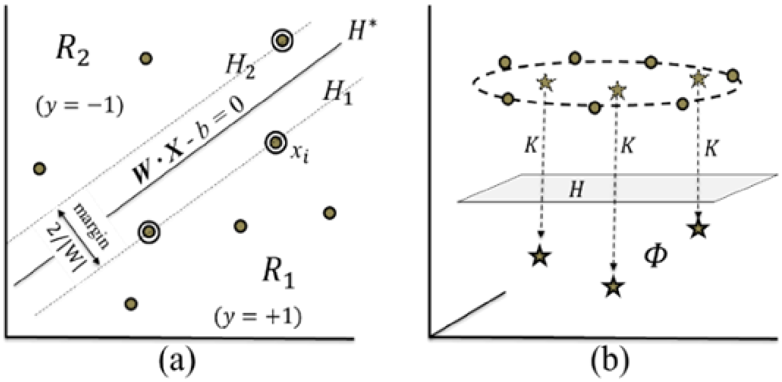


Fig.2.2 Support Vector Machine Operations

Although we assume the data are linearly separable, in practice it is often not the case. Furthermore, a too strict policy results in an excessive margin aimed at avoiding missing classification of some points. To cope with this problem, a concept of slack-variable ξ_i has been introduced to allow some errors instead of having a too wide margin. This is given by replacing the above inequality constraints on to

$$y_i(\boldsymbol{W}^T \boldsymbol{X}_i - h) \geq 1 - \xi_i \quad (i = 1, 2, \cdots, M) \tag{2.10}$$

where slack-variable ξ_i allows those data to be in a margin when $0 \leq \xi_i \leq 1$ and misclassification can be given where holds $1 < \xi_i$. As input training data that tends to misclassify in the slack value is more than 1.0, $\Sigma_i \xi_i$ represents a boundary number of the data in misclassification. The targeted maximization of the margin is equal to the minimization of $1/2\| \boldsymbol{W} \|^2$, which can be shown as an argument to penalize the misclassification and margin error given by a term of $C\Sigma_i \xi_i$. Therefore, now the optimization problem can be shown as

$$\min\ L(\boldsymbol{W}, \xi) = \frac{1}{2} \| \boldsymbol{W} \|^2 + C \sum_{i=1}^{M} \xi_i$$
$$s.t. \quad y_i(\boldsymbol{W}^T \boldsymbol{X}_i - h) \geq 1 - \xi_i\,,\ \ \xi_i \geq 0,\ \forall_i \tag{2.11}$$

This formula has been called soft-margin SVM as it has a specific cost-parameter *C*. This parameter introduces additional control capability which enables adjustment of the balance between the amount of margin allocation and the amount of slack allowance [25]. In a graphical representation on Fig. 2.2(a), when a smaller *C* is assigned, the indicated margin becomes wider. It has been known that the solution of this optimization problem can be obtained by the saddle point of the Lagrangian [26]. Introducing two Lagrange multipliers $\alpha_i \geq 0$ and $\beta_i \geq 0$, we reformulate the objective function into

$$L(\boldsymbol{W}, h, \alpha, \beta) = \frac{1}{2} \parallel \boldsymbol{W} \parallel^2 + C\sum_{i=1}^{M} \xi_i$$

$$-\sum_{i=1}^{M} \alpha_i \{y_i(\boldsymbol{W}^T \boldsymbol{X}_i - h) - (1 - \xi_i)\} - \sum_{i=1}^{M} \beta_i \xi_i \tag{2.12}$$

Lagrangian duality enables this primal problem to be transformed to the Wolfe-dual problem, which is given by:

$$\max_{\alpha,\beta} \boldsymbol{W}(\alpha, \beta) = \max_{\alpha,\beta} \left\{ \min_{W,h,\xi} L(\boldsymbol{W}, h, \xi, \alpha, \beta) \right\} \tag{2.13}$$

By minimization with respect to $\boldsymbol{W}$, *h*, and $\xi_i$of the Lagrangian *L*, a partial derivative is taken on each them:

$$\frac{\partial L}{\partial \boldsymbol{W}} = 0 \quad \Rightarrow \quad \boldsymbol{W} = \sum_{i=1}^{M} \alpha_i y_i \boldsymbol{X}_i^T \tag{2.14}$$

$$\frac{\partial L}{\partial h} = 0 \quad \Rightarrow \quad 0 = \sum_{i=1}^{M} \alpha_i y_i \tag{2.15}$$

$$\frac{\partial L}{\partial \xi_i} = 0 \quad \Rightarrow \alpha_i + \beta_i = C \ \Rightarrow \ \alpha_i \leq C \ (0 \leq \beta_i) \tag{2.16}$$

By substituting them onto (2.12), eventually the optimization problem is transformed into the following dual Lagrangian formula [27]:

$$\max_{\alpha} \boldsymbol{W}(\alpha) \ = \ \max_{\alpha} \left[\sum_{i=1}^{M} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{M} \alpha_i \, \alpha_j y_i y_j \boldsymbol{X}_i^T \boldsymbol{X}_j \right] \tag{2.17}$$

$$s.t.\ \begin{cases} \sum_{i=1}^{M} \alpha_i\, y_i = 0 \\ 0 \le \alpha_i \le C \ \ (i,j = 1,\cdots,M) \end{cases} \tag{2.18}$$

Then, the optimal weight vector $\boldsymbol{W}^*$ can be given by:

$$\boldsymbol{W}^* = \sum_{i=1}^{M} \alpha_i^*\, y_i \boldsymbol{X}_i \tag{2.19}$$

Optimal threshold h^*can be given by:

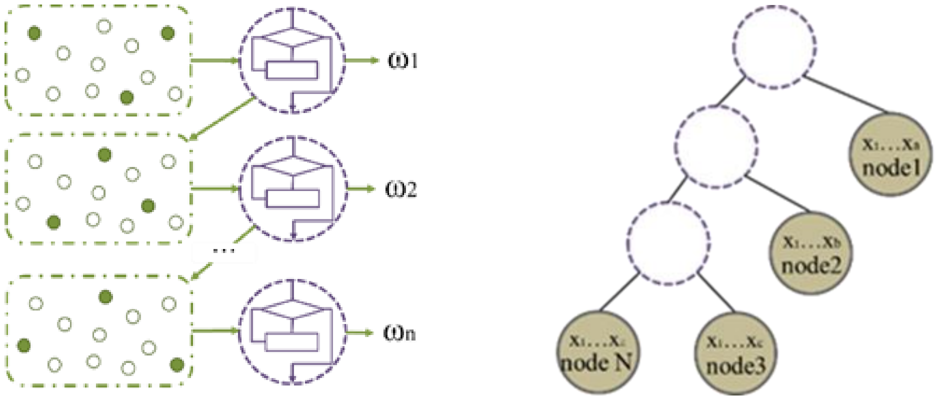
$$h^* = y_j - \sum_{i=1}^{M} \alpha_i^*\, y_i\left(\boldsymbol{X}_i^T \mathbf{X}_j\right) \tag{2.20}$$

Finally, the decision function is provided by:

$$f(\boldsymbol{X}) = sign\left[\sum_{i=1}^{M} \alpha_i^*\, y_i\left(\boldsymbol{X}_i^T \boldsymbol{X}\right) - h^*\right] \tag{2.21}$$

2.2.2 Gradient Boosting Machine Classification

This section explains fundamentals of the Gradient Boosting Machine Classification (GBMC) scheme. Fundamental idea of GBMC stems from a tree ensemble boosting classification [28], [29]. The tree ensemble algorithm starts using a weak tree decision model from a given *n* number of samplings from the entire data space. Then, it sequentially concatenates feedback from those outcomes to build a next improved model by capitalizing miss-classifications applying the weight factor ω as illustrated in Fig.2.3 (a).



(a) Sequential concatenation (b) Tree structure and the nodes

Fig.2.3 Gradient Boosting Machine Operations

The sequentially repeated ensemble learning process is expressed as *boosting* and it greatly improves the classification performance [30], [31]. Eventually, the result forms nodes and the members as in Fig2.3(b), and GBMC finds a best tree structure with the members in minimizing the prediction loss and the tree structure complication.

Having s dimension of explanatory variable $\boldsymbol{X}_i$ and objective variable y_i, in n samples $(i = 1, 2, \cdots, N)$, let $\hat{y}_i$ be the prediction value from the set of $\boldsymbol{X}_i$. Now, we express the error term of the first tree, which can be expressed as $\epsilon_i^{(1)} = y_i - \hat{y}_i^{(1)}$. The boosting scheme sets the error as an objective variable for building a second tree. Then, the second prediction can be $\hat{y}_i^{(2)} = \hat{y}_i^{(1)} + \hat{\epsilon}_i^{(1)}$. The summation of error term till the second tree will be $\hat{\epsilon}_i^{(2)} = y_i - \hat{\epsilon}_i^{(1)} = y_i - (\hat{y}_i^{(1)} + \hat{\epsilon}_i^{(1)})$. In a repetitive manner, the third tree is established having $\hat{\epsilon}_i^{(2)}$as the objective variable, which can be shown as $\hat{y}_i^{(3)} = \hat{y}_i^{(2)} + \hat{\epsilon}_i^{(3)} = \hat{y}_i^{(1)} + \hat{\epsilon}_i^{(2)} + \hat{\epsilon}_i^{(3)}$. The boosting algorithm constructs a newer decision tree in reflecting previous decision knowledge in minimizing compiled errors. For a given n samples from a data set $D = \{(\boldsymbol{X_i}, y_i)\}$ $(|D| = n,\ \boldsymbol{X_i} \in \mathbb{R}^s, y_i \in \mathbb{R})$, the ensemble model in K times boosting produces the prediction value in

$$\hat{y}_i^{(K)} = \sum_{k=1}^{K} f_k(\boldsymbol{X_i}) \tag{2.22}$$

The function $f(\cdot)$ can be expressed as a loss function $l(a, b)$ which counts the degree of loss in the prediction value $\hat{y}_i$ and the target value y_i. Then, the question is how to formulate a t-th tree structure knowing the information obtained from the tree structure so far. The best function $f(\cdot)$ will be found in the form of minimizing the loss, then

$$\min_{f_t} \sum_{i=1}^{n} l\left(y_i, \hat{y}_i^{(k)}\right) =$$

$$\sum_{i=1}^{n} l\left(y_i, \{\sum_{k=1}^{t} f_k(\boldsymbol{X}_i)\}\right) =$$

$$\sum_{i=1}^{n} l\left(y_i, \{\sum_{k=1}^{t-1} f_k(\boldsymbol{X}_i) + f_t(\boldsymbol{X}_i)\}\right) =$$

$$\sum_{i=1}^{n} l\left(y_i, \{\, \hat{y}_i^{(t-1)} + f_t(\boldsymbol{X}_i)\}\right)$$

(2.23)

2.2.3 Extreme Gradient Boosting Machine Classification

This section explains fundamentals of XGB Classification scheme. However, the calculation above considers the evaluation loss only and not considering the complication of a given tree structure and future predictions. Therefore, it has been known to lead to an over-fitting problem. For improving these shortcomings, specifically XGBC introduces the following penalty term Ω into the above equation and then re-defines it to minimize the final objective function $\mathcal{L}^{(t)}(f_t)$, which is provided by

$$\mathcal{L}^{(t)}(f_t) = \sum_{i=1}^{n} l\left(y_i, \hat{y}_i^{(t-1)} + f_t(\boldsymbol{X}_i)\right) + \Omega(f_t)$$

$$= \sum_{i=1}^{n} l\left(y_i, \hat{y}_i^{(t-1)} + f_t(\boldsymbol{X}_i)\right) + \gamma T + \frac{1}{2}\lambda \|\omega\|^2$$

(2.24)

This function consists of three terms. The first term represents the prediction loss as shown l-term. The second term T represents the number of nodes of the tree structure with tuning variable γ. The last term ω represents the weight of each node used to calculate the score on a corresponding tree associated with a tuning parameter λ. Hence, the final objective function evaluates the degree of prediction loss, considering the degree of complexity in a given tree structure.

1) Gradient Tree and the optimized structure

As $\hat{y}_i^{(t)}$ is the prediction of i-th instance at t-th iteration, we try to find a $f_t(\cdot)$ to minimize the objective function given in (2.24). For finding an optimal state in limited time, second-order approximation is applied for quickly finding the optimized objective function by taking Tylor-expansion around zero. Then, (2.24) will be

$$\approx \sum_{i=1}^{n}[l\left(y_i, \hat{y}_i^{(t-1)}\right) + g_i f_t(\boldsymbol{X}_i) + \frac{1}{2}h_i {f_t}^2(\boldsymbol{X}_i)] + \Omega(f_t)$$

(2.25)

where

$$g_i = \partial_{\hat{y}_i^{(t-1)}} l\left(y_i, \hat{y}_i^{(t-1)}\right) \text{ and } h_i = \partial^2{}_{\hat{y}_i^{(t-1)}} l\left(y_i, \hat{y}_i^{(t-1)}\right)$$

(2.26)

are the first- and second-order gradients on the loss function. By removing the constant terms, the following simplified objective function is obtained at step t.

$$\tilde{\mathcal{L}}^{(t)}(f_t) = \sum_{i=1}^{n}[g_i f_t(\boldsymbol{X}_i) + \frac{1}{2}h_i {f_t}^2(\boldsymbol{X}_i)] + \Omega(f_t)$$

(2.27)

For obtaining Quadratic form, this is re-written into the following form.

$$= \sum_{i=1}^{n}[g_i f_t(\boldsymbol{X}_i) + \frac{1}{2}h_i {f_t}^2(\boldsymbol{X}_i)] + \gamma T + \frac{1}{2}\lambda\|\omega\|^2$$

$$= \sum_{j=1}^{T}[\sum_{i\in I_j} g_i f_t(\boldsymbol{X}_i) + \frac{1}{2}\sum_{i\in I_j} h_i {f_t}^2(\boldsymbol{X}_i)] + \gamma T + \frac{1}{2}\lambda\sum_{j=1}^{T}\omega_j^2$$

$$= \sum_{j=1}^{T}[(\sum_{i\in I_j} g_i)\omega_j + \frac{1}{2}\sum_{i\in I_j} h_i\omega_j^2] + \gamma T + \frac{1}{2}\lambda\sum_{j=1}^{T}\omega_j^2$$

$$= \sum_{j=1}^{T}[(\sum_{i\in I_j} g_i)\omega_j + \frac{1}{2}(\sum_{i\in I_j} h_i + \lambda)\omega_j^2] + \gamma T$$

(2.28)

Therefore, in a given tree structure $q(\boldsymbol{X})$, the optimal weight ω_j^* of the node j can be derived from $d\tilde{\mathcal{L}}^{(t)}/d\omega_j = 0$ in

$$\omega_j^* = -\frac{\sum_{i\in I_j} g_i}{\sum_{i\in I_j} h_i + \lambda}$$

(2.29)

Then, the corresponding optimal value in a given tree structure can be given by

$$\tilde{\mathcal{L}}^{(t)}(q) = -\frac{1}{2}\frac{\left(\sum_{i\in I_j} g_i\right)^2}{\sum_{i\in I_j} h_i + \lambda} + \gamma T.$$

(2.30)

This measures the scoring of the quality of tree structure at q, and the larger $\left(\sum_{i\in I_j} g_i\right)^2 / \sum_{i\in I_j} h_i + \lambda$ is, it returns the lesser value of objective function $\tilde{\mathcal{L}}^{(t)}$.

Chapter 3

3 Affinity Propagation Clustering for Cellular V2X

3.1 Introduction

This section explains our first contribution on the VANET clustering scheme applying Affinity Propagation Clustering (APC) for C-V2X. The APC clustering based on well mathematically proofed concept hence, the extension is quite flexible as the fundamental idea uses *similarity* function to find homogeneous members in measuring it. This approach differs from conventional one-by-one oriented heuristic approach therefore its application widely expandable to tuning similarity function and the evaluation terms.

3.1.1 Objectives

The fundamental objective of forming vehicle cluster is obtaining system scalability, in considering 5GS is platform of a global communication system. One-to-N connectivity will cause various system un-stability such as signaling congestion, numbers of state information managements, and control and user-plane management in each of connectivity. These overheads have been typically known increased in the case of small but frequent MTC oriented communications. Therefore, clustering mechanism is strongly demanded from operators and those who provides the system. Having the requirements, we apply mathematically well proofed and extensible clustering algorithm called APC targeting to dynamically moving vehicles.

3.1.2 Related Works

This part explains related works on clustering schemes. A highly relevant paper which outlines a mechanism of VANET-UMTS integration is presented in [32]. This paper proposes VANET dynamic clustering mechanism, called Clustering-based Multi-metric adaptive Mobile Gateway Management mechanism (CMGM), using specific metrics for finding minimum number of vehicular gateways, *i.e.*, Cluster Heads (CHs). According to our observation, although this proposes a comprehensive mechanism in a dynamic mobility environment, its implementation is hardly to realize due to the proposed concept uses inclination angle θ toward eNodeB and another inclination angle between vehicles. Beacause the detection of these angles of inclination would associate substantial error components as individual vehicles are dynamically in motion. Another study of hybrid system integration between IEEE 802.11p based VANET clustering with 4G cellular system can be seen in [33] and [34]. In this proposal, Vehicular Multi-hop algorithm for Stable Clustering (VMaSC) scheme is introduced. For a CH selection, this scheme proposes relative mobility metric calculated from the averaged relative speed with respect to the neighboring vehicles with some theoretical analysis. Although, the proposed algorithm and procedure become light weight, it still spends large state-management effort using specific state transition matrix and communication messages. According to our observation, implementing such specific procedure and algorithm onto each vehicle is hardly achievable.

In contrast to the aforementioned conventional VANET clustering schemes, contemporary VANET clustering come to emerge in [17], [35], and [19] based on mathematically well proofed concept called Affinity Propagation, which is originally published in *Science* [20]. Affinity PROpagation for VEhicular networks (APROVE) is presented in [17], then improved in [35], specifically by means of metrics for the similarity function of AP using vehicles distance and future *prediction* time τf. In the author's simulation, the value of τf = 30sec is used, which means individual position of vehicles are forecasted 30sec later and it is obviously imagined that it associates large prediction error as vehicles change their position during the predicted time. The APROVE also uses a message which exchanges its information every second. In this study, preliminary iterations required around 10 times when the number of neighborhood vehicles is 40, which requires 10 sec for the Clustering Interval (CI). This value is not trivial as vehicles are in dynamic motion. They would change their location more than 200m during the elapsed time in a highway.

3.1.3 Contributions

This part explains our contribution on AP Clustering for C-V2X. This study aims to design a distributed VANET clustering scheme for providing system stability and scalability thorough the ideal CH working for group proxies the communications. The proposed function has been designed to be applicable in time critical and monadic mobile targets. The main contributions on this part can be described following paragraph.

#1 We designed the proposed scheme to be light-weight, simple implementation and

effectively apply to motion dynamics vehicles which constantly changing the motion.
#2 The scheme applied fundamental feature of Cellular V2X PC-5 radio performance and leveraged to fit inter-vehicle communication for the clustering message exchange.
#3 It incorporated fundamental concept of Affinity Propagation (AP) which provides extending ability by modifying the similarity function to decide clustering formation.
#4 The proposed scheme is targeting to provide distributed clustering control and fully leveraged by the C-V2X functionality.

3.2 Proposed VANET Clustering Scheme

This section explains the detail of our VANET clustering proposal employing the concept of APC. The proposed VANET clustering scheme uses the idea of APC, by enhancing its similarity function with Euclidian special distance, *i.e.*, vehicle's position information along with the velocity information by normalizing each part. Through message exchanges between vehicles the similarity function identifies vehicles similar motion dynamics that result in formation of clusters of those vehicles.

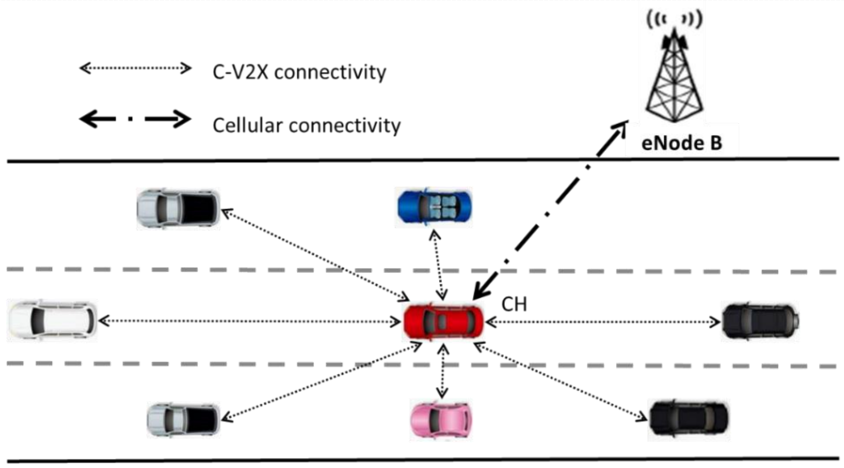


Fig.3.1 C-V2X PC5 Interface for Inter Vehicle Messages

Fig.3.1 shows the use of Cellular V2X technology for exchanging messages between vehicles. The cluster members use Cellular V2X PC5 interface which is direct vehicle to vehicle local connectivity with the neighbor nodes. The figure depicts after a specific cluster is formed from the APC process and specifically shows connectivity to the identified CH. Each of the vehicles has ability to connect and communicate surrounding vehicles as far as the Cellular V2X radio reaches. All the required messages for APC are sent in the application layer over the reliable radio interface. The improved data transmission frequency of C-V2X largely contributes to the performance of APC as it requires frequent message exchange among the nodes. The similarity function of AP can be flexibly designed as far as it represents state of similarity of target object dynamics.

3.2.1 NMDP-APC Clustering Scheme

This section explains key idea on the proposed NMDP-APC Clustering scheme. Reviewing previous works, we identified the most essential metrics which represent VANET's motion dynamics are the vehicles current position and velocity [32], [34]. Hence, we designed our similarity function by associating these key terms into a single formula as shown in (3.1). The similarity function can be obtained by a summation of the negative value of distance vector of the position of target vehicles with the negative difference of each vehicle's current velocity. In order to compose two different terms in a single similarity function, each term has been normalized from 0 to +1, which denoted as "*nor*" in (3.1). We also assigned weighted parameter α and β on the normalized components for a control nob of each term. This approach provides significant extendibility, because it allows multiple dimensional terms into a single similarity function with adjusting the proportion by the weight parameters. According to our observation, no other study has yet proposed this level of flexibility on the similarity function on the APC. Eventually, our enhanced similarity function *s(i,j)* is shown as follows:

$$s(i,j) = -\left(\alpha\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_{nor} + \beta\|\boldsymbol{v}_i - \boldsymbol{v}_j\|_{nor}\right) \tag{3.1}$$

where $\alpha + \beta = 1,\ \alpha, \beta > 0$

$$\boldsymbol{x}_i = \begin{bmatrix} x_i \\ y_i \end{bmatrix}, \quad \boldsymbol{v}_i = \begin{bmatrix} v_{x,i} \\ v_{y,i} \end{bmatrix} \tag{3.2}$$

Where $\boldsymbol{x}_i$ represents current position in x and y coordinate of node i and $\boldsymbol{v}_i$ represents current velocity in x and y coordinate of node i respectively. Note that in contrast to previous work [35], proposed similarity function of the NMDP-APC does not compose any *future* prediction value. This characteristic is a remarkable difference in our proposal comparing to the previous study.

3.3 Simulation Results

3.3.1 Data Transmission Frequency & Velocity Components

In order to confirm the validity of our proposed similarity function (3.1), we conducted this simulation using simple traffic scenario, which merely placed three vehicles in even distance of 50m interval, with and without considering their velocities. The velocity is assigned to vehicle A = 90 km/h, then B and C= 100 km/h. Fig.3.2 (a) displays the original distribution of three vehicles in x and y coordinates. The second row Fig.3.2 (b) displays proposed NMDP-APC clustering result *only* considering current location, by assigning the value of $\alpha = 1.0$ and $\beta = 0$. The third row Fig.3.2 (c) displays NMDP-APC clustering result considering *both* current location and velocity, where $\alpha = 0.8$ and $\beta = 0.2$ are assigned. We also conducted this simulation by assigning different combination of velocities such as, A and B = 90km/h, then C = 100km/h. In addition, we changed the evenly spaced distance from 50m to 200m, in 50m steps. From these simulations, adequate vehicle clusters are formed when considering *both* location and velocity as shown in Fig.3.2 (c). However, in single dimensional clustering case which only taking into account location information did not show expected clustering formation. This simple demonstration indicates valid evidence that our multi-dimensional AP scheme works appropriately.

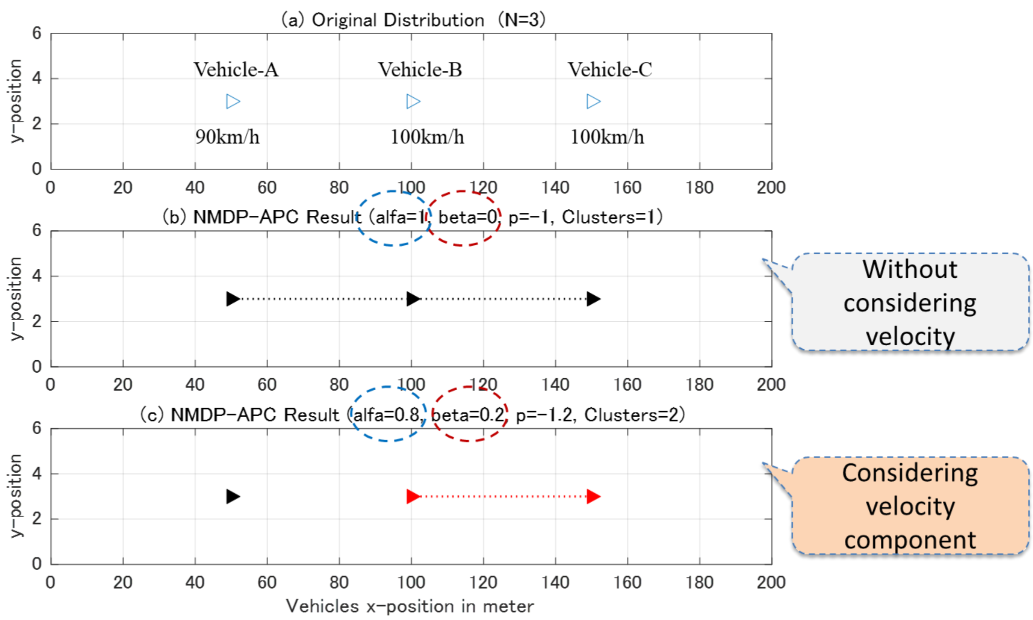


Fig.3.2 NMDP-APC Clustering Result with and without Considering Velocity

3.3.2 NMDP-APC with Real Traffic Data

This part explains the sets of simulation results of NMDP-APC outcomes on real traffics. For the real traffic data in this simulation, we captured one-minute traffic video on a four-lane Inter City highway (I-294) on Aug 8th 2017 in Chicago, Illinois, USA. From the real video traffics we obtained leading vehicles' position and velocity information, then identified following vehicles' traffic information via aforementioned GHR car following model. A leading vehicle is represented with sufficient distance from the vehicle in front, which eventually tends to drive at its own pace. Instead, following vehicle tends to follow leaders driving pace as it has limited headway distance. From the result of above process, every 5sec interval of each vehicle's traffic data in four lanes is traced. The overall sample data length is one minute. Then, the data was fed into our NMDP-APC MatLab code to analyze how the proposed scheme identifies clusters with the CHs in the real traffic data. Fig.3.3 (a) shows original vehicles distribution of the sampled traffic data at t = 20sec. As previously mentioned, we have 12 sets of data with an interval of every 5sec, i.e., from 5 to 60secs. We examined traffic data in each time interval, yet due to the paper space limitation only a single result is presented. The number of vehicles in the observed space is 19, shown as N = 19. Fig.3.3 (b) is the NMDP-APC result with simulation parameter used $\alpha = 0.8$ and $\beta = 0.2$. In each of the cluster, it can be observed that CH is identified in mostly the central location by linked dot line with the member nodes. The *px*-value was -1.3188, where the value enables to control the number of clusters *i.e.*, cluster granularity in the NMDP-APC. In this simulation, the default value is used which is the median value of the similarity function. This is an important parameter in the APC, we examine the clustering granularity by changing the *px*-value in the later section. The last row Fig.3.3(c) shows the Net Similarity value in the APC iteration process. The Net Similarity indicates value of given similarity function during the APC iteration process. From the figure, after 5 iterations the final configuration seems to be formed as the value reaches saturation in a maximum value. In the following simulation, we examine the minimum number of iterations required to identify stable clusters by changing number of target vehicles.

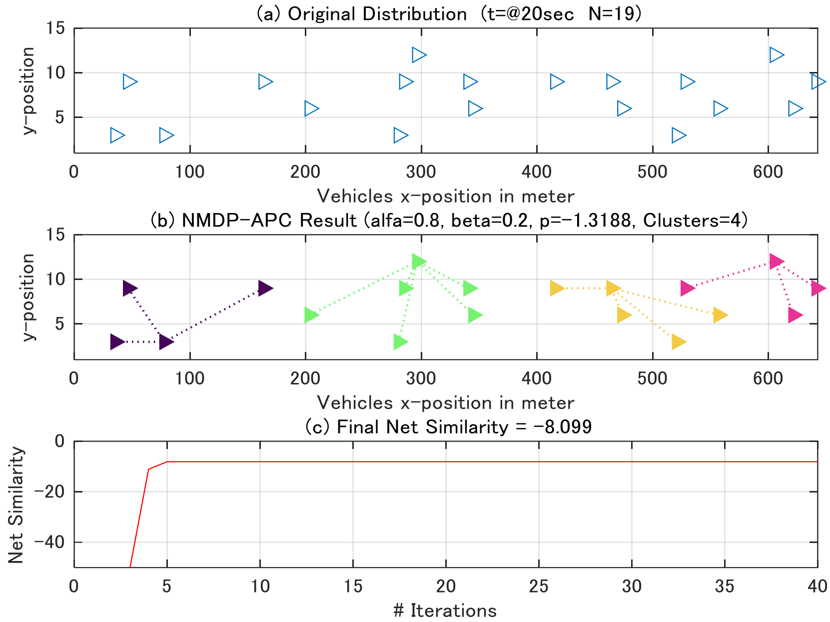


Fig.3.3 NMDP-APC Clustering Result in Real Traffic Data

3.3.3 Minimum Numbers of Required Iterations

As mentioned above, the APC identifies final cluster configuration by reaching saturation of the Net Similarity value. This implies no other best clustering configuration can be identified even the APC tries to find. The required number of iterations tends to increase by increasing the node numbers N. The computational complexity is generally observed in the order of $O(N^2)$ [36]. Having a dynamically moving VANET scenario the processing time is critical, so the shorter it is the better. Yet, insufficient number of iterations will not reach final cluster configuration. Hence here, we conducted the following simulations to identify minimum number of iterations required in aforementioned real traffic data. Fig.3.4 shows minimum number of iterations required to saturate the Net Similarity value in our NMDP-APC scheme. While time elapses, it is observed that more number of vehicles came into the observation space. The number of minimum required iteration is shown with triangle-mark, which is increasing as the number of processed vehicles increases. This tendency is reasonable because the necessary processed data points are increasing. In end, the simulation shows that at least 10 iterations are required for identifying stable clustering in N=60 which is a largest number of vehicles in this simulation. Considering real implementation scenario of APC, this degree of number of iterations is acceptable. If the number of cluster member goes too large, APC process will require more processing time by increasing the computational complexity. In the next section, we examine to control the cluster granularity by adjusting specific parameter called *px*-value.

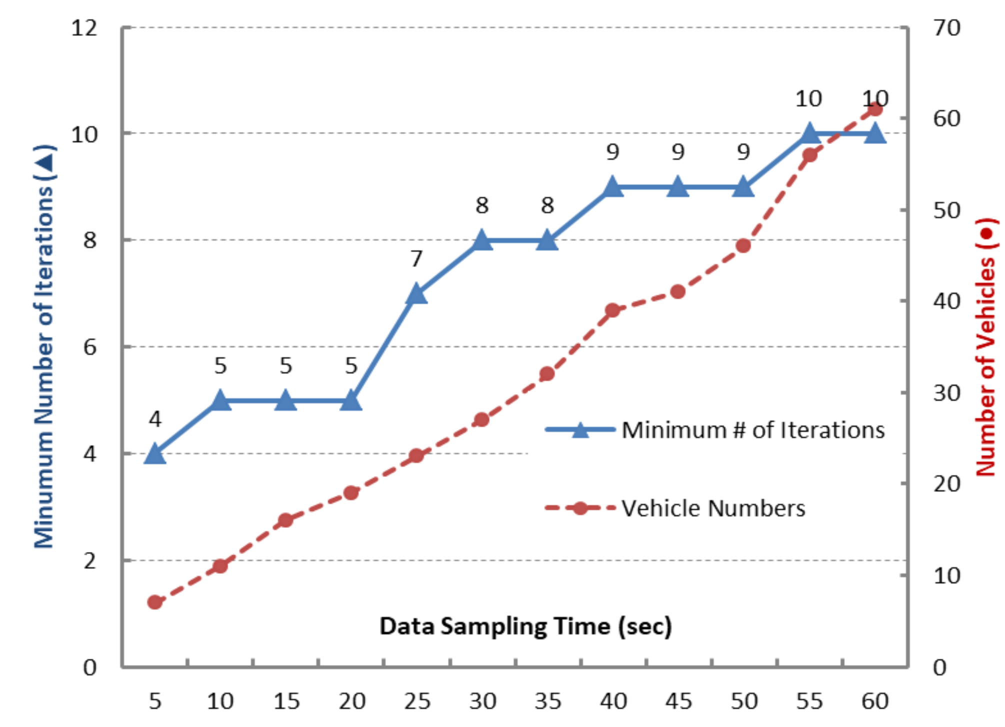


Fig.3.4 NMDP-APC Minimum Number of Required Iteration with Number of Vehicles

3.3.4 Cluster Granularity Changing by *px*-value

This part evaluates the cluster granularity changing by *px*-value in the simulation. The number of clusters, or granularity of each clusters can be controlled by varying the *px*-value in our NMDP-APC. When the cluster members are sending less transmission data, the cluster size could be large because the CH will not be a communication bottle neck. In contrary, when cluster members need to send relatively large amount of data, fewer members contribute to avoid their CH to be a communication bottle neck. Another angle of consideration is vehicles motion dynamics. When vehicles are moving at a faster pace and in a random direction, the cluster size should be small. Having these potential consideration points, we conducted this simulation for adjusting cluster granularity by changing the *px*-value in our obtained traffic data. Fig.3.5 shows a simulation outcome of number of clusters identified when the *px*-value has been changed between the ranges of 0.1 to 2.0 multipliers. The standard *px*-value is median value of the similarity function. We changed the original value by multiplying it with 0.1, 0.2, 0.5, 1.0, 1.5 and 2.0 respectively, and then observed the number of clusters made over the same real traffic data sampled in 5sec intervals as mentioned previously. At each sampling time, the processed numbers of vehicles correspond to the numbers as shown in Fig.3.5. The result clearly indicates that by changing the *px*-value, NMDP-APC scheme clearly enables to control the clustering granularity. Where, smaller *px*-value produces the large number of clusters. In contrarily, larger *px*-value forms the smaller number of clusters.

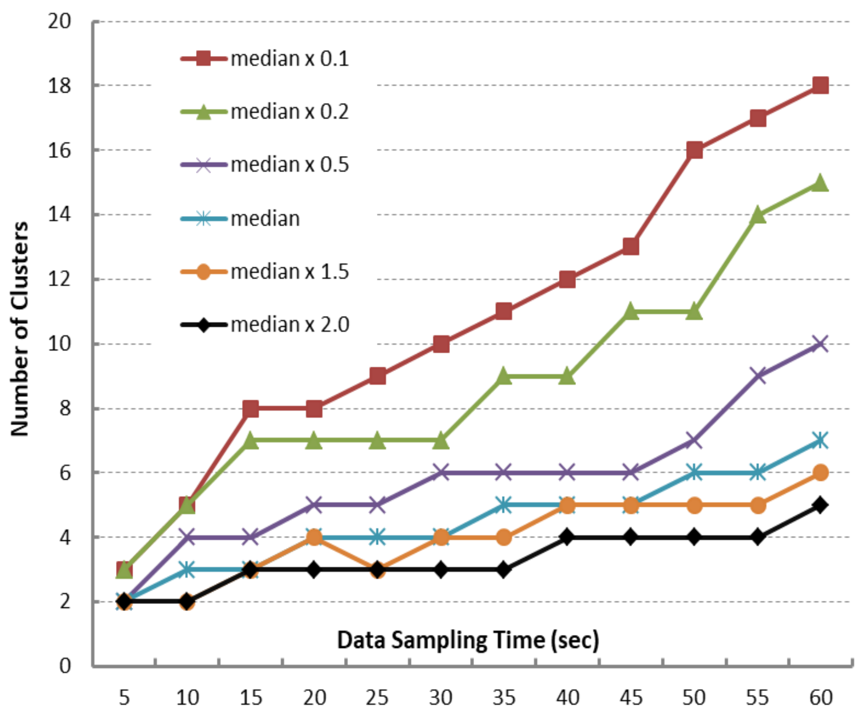


Fig.3.5 NMDP-APC Number of Clusters Changing the *px*-value

3.4 Summary

This study proposed a novel NMDP-APC scheme which consists of normalized multi-dimensional parameters in a similarity function targeting C-V2X VANET. Specifically, we introduced two dimensional parameters, which were vehicles' current position and current velocity with weighted value α and β over the terms. The idea of this scheme infers further extensibility of input terms and dimensions. We also conducted extensive simulations in proposed scheme to evaluate and verify its capability in the simulation part. We used a well proofed Gazis-Herman-Rothery (GHR) car following model in the process of simulations. This scheme greatly helped our simulations and contributed for obtaining various results in real traffic data. We applied C-V2X radio to improve the frequency of the data exchange, which significantly contributed minimizing APC clustering interval. In using simple vehicle formation, we tested the validity of proposed similarity function. This process clearly identified that velocity component is an essential term for the adequate clustering formation. We also applied the NMDC-APC to real traffic data and successfully clustered the live traffics in various conditions. The simulation result also identified that the number of minimum iterations increases when the number of target object increases. We demonstrated that controlling clustering granularity by changing the *px*-value, *i.e.*, it is called preference value in the AP. We identified success control of clustering granularity by changing the value in our NMDP-APC.

Application of APC to VANET is quite demanding. The results of this study further open up various fields of the NMDP-APC applications as our proposed similarity function enables to incorporate various metrics into the formula. The proposed scheme does not require specific implementation platform as it designed to exchange APC messages over an application layer. So far, realization of conventional proposals heavily depends on lower layers implementation such as MAC layer of DSRC. Instead, our scheme enables to realize in higher layer. This significantly contributes flexibility in its application to various demanded scenarios.

Chapter 4

4 VANETs Clustering Formulation Applying Machine Learning

4.1 Introduction

This part introduces research background of proposed VANET clustering formulation applying a machine learning. In the previous work, we were able to successfully construct VANET clusters from NMDP-APC scheme. we also obtained the control scheme of desirable cluster size and the simulation result clearly indicated. By changing *px*-value, we can formulate desirable VANET cluster size depends of the needs. However, un-solved question only remains *how* the ideal granularity shall be determined. This is a remaining question after we obtained the clustering control mechanism. The ideal cluster size and granularity should be determined autonomously incorporating traffic situation. In order to resolve this question, we applied a machine learning scheme. This is an incredibly powerful and extensible approach we have not leant so far. Here we introduce the details from the following section.

4.1.1 Objectives

This section explains the objectives and target we pursue through this study. The first objective is to derive ideal clustering granularity knowing vehicular dynamics. The second objective is to design a system scheme having a capability of autonomous clustering in integrating C-V2X procedure. The third objective is to find a scheme which has high adaptability and extendibility once we find further improvements and extension onto the original scheme. Eventually, this scheme is easily implemented through using a platform of machine learning once fundamental part is developed.

4.1.2 Related Works

The metrics of VANET clustering scheme observed vary: some schemes use single-metric and others multi-metrics. Such variations can be seen in [36], [37]. Specifically, mobility-based clustering which focuses on inter-vehicular mobility dynamics is identified as a key direction in contemporary study. A reference [32] outlines a mechanism of VANET-UMTS integration and proposes VANET dynamic clustering mechanism using specific multi-metrics for finding a minimum number of vehicular CHs. Although it proposes a comprehensive mechanism targeting a dynamic mobility environment, its realization would be challenging as it requires the inclination angles toward eNBs and between vehicles. Studies on a multi-hop clustering scheme with IEEE 802.11p and 4G hybrid systems are introduced in [33], [34].

Generally, VANET clustering schemes can be categorized as either *centralized*- or *distributed*-clustering schemes. In centralized clustering schemes, cluster formation is performed via a NB or Rode Side Unit (RSU) with periodical message exchanges to the target vehicles. W. Qi [16] proposed an SDN-based centralized clustering scheme by exploiting a social pattern, *i.e.,* knowing the vehicles' traffic pattern to deduce an expected route, where the metrics of inter-vehicular distance, relative speed and vehicle attributes are used. In distributed clustering schemes, cluster formation and CH selection are performed only between the vehicles through message exchanges without involving managements of central node in a fully distributed manner. The centralized clustering approach is advantageous in that it has more computational resources and managements available in the central entity. On the other hand, from the standpoint of nomadicity of vehicular mobility, the distributed clustering approach has advantages that it can avoid signaling concentration and a single point of failure; specifically, it allows autonomous decision making among widely spanned vehicular targets.

In contrast to the conventional application of APC, our proposed NMDP-APC in C-V2X is designed to target the following three key advantages. Firstly, our proposed scheme uses the vehicle's *current* position and velocity that are not associated with any predicted value. Secondly, our scheme autonomously deduces a desirable clustering granularity not only by knowing vehicles' motion dynamics but also by incorporating the traffic density and access congestion status by applying ML. Thirdly, it leverages the power of C-V2X technology for more improved performance, *i.e.*, a short uplink transmission interval of 0.1 sec [37] with higher reliability and lower latency. In a system-wide perspective, we designed our scheme by applying mobility-based metrics, specifically the position and velocity, in single-hop clustering formation, taking advantage of C-V2X wireless, which spans enough in one hop, and a distributed-clustering scheme, which totally fits vehicles' nomadicity .

Key salient features of vehicular communication to be applied to succeed in 5G systems are summarized by S. Shah *et al.* [39]. The importance and mechanism of 5G Mobile Edge

Computing (MEC) with VANETs are analyzed in [40] and Z. Ning, *et al.* [41]. Low latency is an essential key capability for VANET networks, especially when the target application involves unmanned vehicles that generally require ultra-low latency capabilities of the system. MEC can process and deliver data quite efficiently by pushing the computing resources to the Mobile Vehicular Cloud (MVC) located at the edge of the network as shown in [42] and X. Ma *et al.* [43]. It is also notable that a game theory for VANET clustering is employed by A. Khan *et al.* [44] to find a balance of total throughput capacity in consideration of the clustering granularity, *i.e.*, population share. In [44], the RSU is used as a centralized controller to signal an overall observation, compute the average payoff of the entire population of clusters and then broadcast it to all clusters.

Potential application of ML over vehicular networks was investigated by H. Ye *et al.* [24]. A framework for ML on vehicular networks was reviewed by L. Liang *et al.* [45]. D. Tian *et al.* proposed an application of a Hebb neural network, which intelligently learns the topology and forms vehicle clusters [46]. An application of Q-learning for cluster head identification is introduced by Z. Khan *et al.* in the proposal of two-level clustering [47]. Level-1 CHs are determined through fuzzy logic algorithm applying following metrics: relative velocity, multiple-connectivity and link-reliability. Level-2 CHs are determined by an improved Q-learning theory which will be a gateway to a Node-B. However, no clear proposal has been presented as to how an identical *number* of clusters can be identified in these studies.

4.1.3 Contributions

This part summarizes our contribution on autonomous VANET cluster formation applying a machine learning. Our proposed scheme and procedure largely relied on fundamental mechanism of C-V2X. This makes highly adaptable and greatly ease the implementation. One of the advantages of applying a ML is its expendability, in changing of components of task vector and evaluation criteria original mechanism can be easily extending to another needs. The main contribution in this study can be listed as follow;

#1 Vehicle clusters are autonomously formed through NMDP-APC process in consideration of real-time motion dynamics.
#2 The cluster granularity is designed to be derived from a ML algorithm with minimum criteria.
#3 Traffic density and cellular congestion states are specifically considered for granularity determination.

#4 A fully distributed system is designed with its entire procedure and sequence in line with 5G C-V2X.

4.2 Proposed VANETs Clustering Applying a ML

Having the explanation of the motivation and review of related works this section shows the detail of our proposed of VANET clustering scheme applying a ML in the following section. As shown in Fig.4.1, cluster formation takes place at Time = t and at an interval of time later. At Time = t, a cluster is formed based on the APC clustering scheme and identified as a CH with the members denoted as vehicle-B and C in the diagram. Reflecting the vehicles' motion dynamics, the associated members are updated at Time = $t+n$ as in Fig.4.1(b). In this case, although the CH has not changed, some members have been changed: vehicle-C has disappeared from the original wireless coverage and vehicle-D has become a new member because of its similarity evaluated by the clustering scheme. In this diagram, vehicle-E stays independent from the cluster as it has an intention to be a stand-alone or the dynamics makes it less likely to be a member of the cluster. The proposed procedures are explained in the following subsection.

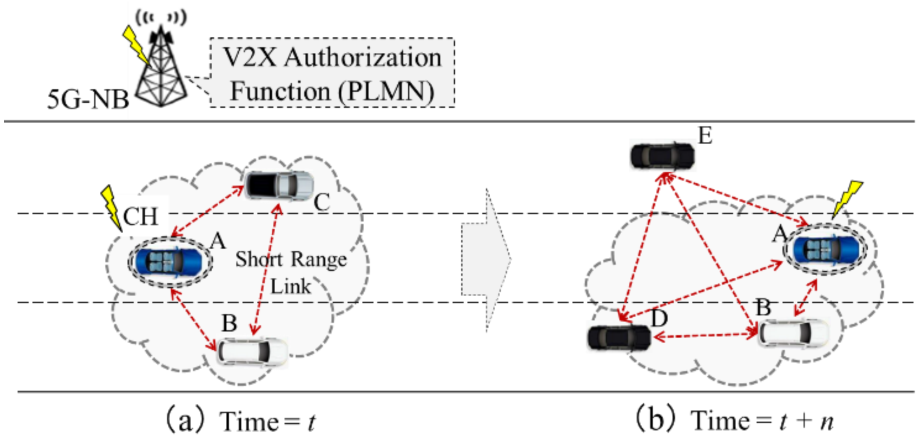


Fig. 4.1 Clustering Formation in Traffic Dynamics

4.2.1 Proposed Clustering Sequence over C-V2X

Fig. 4.2 shows the enhanced signaling sequence exchanged in the above scenario. VE represents the Vehicle-UE that complies with 3GPP C-V2X scheme. Step 1 indicates the provisioning process for one-to-many and PC5-based communication between the VEs. The provisioning includes three important aspects: a) identifying authorization policies and parameters; b) setting up V2X direct communication parameters; and c) setting up radio parameters for not-served by 5G-RAN. The authorization policies and parameters include UE's authorization parameter from a PLMN used to prevent malicious UEs from engaging. The V2X direct communication parameters include V2X-Layer2-ID, IPv4/v6 preferences and Application Layer IDs, which are used by the VEs to perform one-to-many communications. Radio parameters for VEs not-served by 5G-RAN provides the capability for one-to-many V2X communication in considering geographical dependencies of frequency bands. In Step 2, each VE starts to exchange messages to surrounding vehicles by looking at the destination and source V2X L2-IDs on a specific C-V2X short-range communication link. Step 3 is the message exchange of NMDP-APC process using one-to-many PC5 direct communication among the neighbors. Clustering Request is a specific message sent from a VE intending to form a cluster. Such request message can be sent from any of the neighbors such as VE-A, B and C. Triggering reception of the Clustering Request message, these vehicles initiate the message exchange of NMDP-APC. Step 4 indicates the results of clustering formation and the CH identified after the APC process for the cluster indicated in Fig. 4.1(a). When a VE becomes a member of the cluster, the VE will enter In-Clustering State and then inform neighbors that it has becomes a member of the cluster. Sometime later as in Fig. 4.1(b), VE-D initiates Step 6 and 7, which are identical to Step 1 and 2 respectively, and it also intends to form a cluster. After the provisioning process, an additional Clustering Request messages is sent from VE-D as shown in Step 8, and then an updated cluster is formed via additional NMDP-APC process considering the dynamics of the neighbors. Note that VE-E remains stand-alone as it has independent dynamics, or it has an intention to be in standalone communication mode. All parts shown colored in-red are the specific enhancements that we proposed over the C-V2X original sequence. The sequence in Fig. 4.2 covers most of the scenarios by which a new VE joins in and evacuates from a cluster. When CH needs to be changed or replaced, V2X-to-NW Relaying function can be applied to CH relocation process between the new and old VEs, which will be explained later section with its the procedures.

We have so far explained the fundamental mechanism of NMDP-APC, referring to a scenario that takes account of VE's motion dynamics. We have yet to offer explanation, however, for the question as to how the clustering granularity should be determined. We employed a ML scheme to challenge the question while vehicles are dynamically in motion. The following section will explain the ML scheme, starting from its concept.

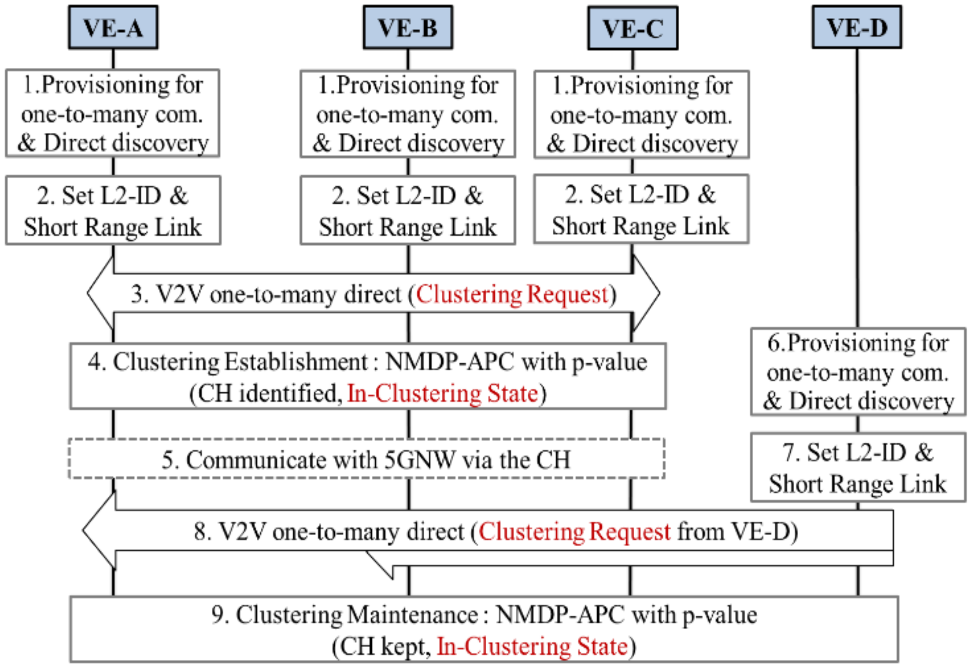


Fig. 4.2 Enhanced Message Sequence over C-V2X

4.2.2 Proposed Task Vector & Decision Tree

This part explains the proposed task vector and decision tree for a proposed ML scheme. It is generally known that obtaining an applicable *trained* decision function $f(\boldsymbol{X})$ requires a certain volume of training data. In this study, a predefined *decision-tree* is applied for identifying the level of clustering granularity, *i.e.*, an objective variable y from a set of explanatory variables, which consists of the task-vector $\boldsymbol{X}$. SVM is known to be well applicable for its sufficient volume of data and a clear relation of explanatory variables to the objective variable.

With the mapping decision of the task-vector $\boldsymbol{X}$ to an objective variable y, the classification is provided as $D = \{(\boldsymbol{X}, y) \mid y = sign[f(\boldsymbol{X})]\}$. In this study, we decided to classify the clusters in four levels based on the size: large, medium, small-medium, and small. In consideration of the obtained real traffic data and the span of C-V2X radio propagation, we set a four-level classification, although theoretically the number of levels can be changed to any. Numerically it can be shown as $D \in \{0,1,2,3\}$ corresponding to the above four levels of cluster size. We designed the task-vector $\boldsymbol{X}$ consisting of the following minimum set of explanatory variables for deciding the cluster size. We kept the number of elements to a minimum, otherwise it would have led to a longer computational time and potential deterioration of prediction performance because the target vehicles are in motion.

$$\boldsymbol{X} = (I_v,\ F_p,\ T_d,\ Nb_c) \tag{4.1}$$

In (4.1), I_v is the communication volume, which we assumed to be 100 kbps uplink data randomly generated in a certain interval from each vehicle. F_p represents an initial preference as to which mode is used to establish a session: clustered mode or individual mode, which can be denoted as $F_p \in \{0,1\}$. The preference value of $F_p = 0$ represents an intention of independent session establishment, in which case a session is established directly and connected to a cellular network. A broad-band data upload could be a use case. On the other hand, $F_p = 1$ represents an intention to form a cluster, in which case a session will be established via a CH. The third attribute constituting the task vector is the traffic density of vehicles shown as T_d. We obtained real traffic data from a highway and applied them in our simulation. According to the obtained traffic data, the maximum number of observed vehicles in a 2 km span, *i.e.*, in a 1 km radius in a single direction. The vehicle density was obtained from the number of vehicles in the observed range. The last attribute constituting the task vector is the congestion status of a PLMN access, denoted as Nb_c, the set of which can be denoted as $Nb_c \in \{0,1\}$, where $Nb_c = 0$ represents a normal access state and $Nb_c = 1$ represents a congestion access state. When a particular access is in a congestion state, the number of clusters is made smaller, as aggregation effect contributes to reducing the number of the cellular access connections. This parameter is therefore factored in our task-vector.

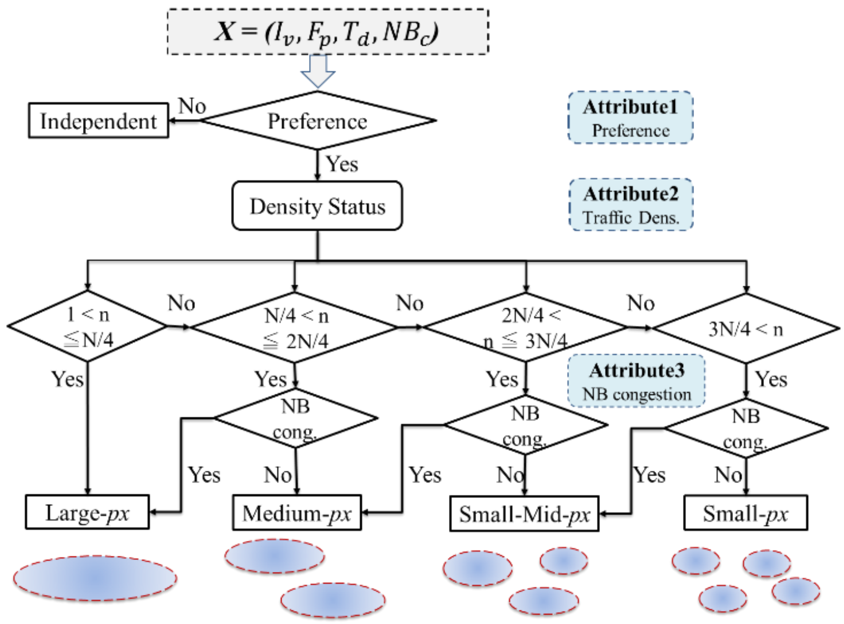


Fig. 4.3 Decision Tree for Clustering Granularity Determination

4.2.3 SVM Classification Mechanism

The fundamental idea and our proposed NMDP-APC scheme have already explained in the previous chapter. Hence, we proceed the introduction of novel part of cluster granularity identification scheme in the following subsection.

4.2.4 Procedure of NMDP-APC ML delivered granularity

This part explains the proposed procedure of NMDP-APC with ML delivered granularity. Once a *px* value is given, NMDP-APC enables the control of the desired clustering granularity in the process. We employed a ML scheme to obtain the desired clustering size from the *trained* decision function $f(\boldsymbol{X})$ as this process identifies the desired *px* value. Table 4.1 explains how NMDP-APC process uses the ML-deduced *px*-value segmenting three procedures parts. As Part-A has been mostly explained with the sequence in the previous section, let us focus on the procedure in Part-B and C for the sake of paper space.

Part-B covers the procedure of an NMDP-APC re-clustering case for a member change. Step 06 and 07 is the provisioning steps identical with step 01 and 02, respectively. Step 08 initiates re-clustering process triggered by a Clustering Request Message (CRM) from a VE which has engaged in a clustered V2X wireless range. Step 09 starts another NMDP-APC process with a ML driven *px*-value to form a new cluster, aiming for cluster maintenance upon receiving the CRM. Step 10 is a process in staying original CH and the CH updates the associated members' context information. Step 11 is a process triggered by a VE about to leave the cluster because of wireless range-out or re-clustering request reception from a member. Then, step 12 starts another NMDP-APC process with a ML-driven *px*-value to form a new cluster in the same manner as in step 09. The CH only maintain the context of updated cluster members. Part-C provides a set of procedures when initial CH replacement is required. In step 13, a CH receives a CRM from a VE. Step 14 initiates Cluster Maintenance by executing another NMPD-APC process. In step 15, an original CH is recognized to be replaced and CH_{NEW} is identified. Then in step 16, CH_{OLD} and CH_{NEW} exchange the context of associated members' information through the V2X-to-NW relay [48]. Step 17 is an annex procedure that enables a recovery from an abnormal state to a normal state.

TABLE 4.1

PROCEDURE OF NMDP-APC WITH MACHINE LEARNING DERIVED *PX*-VALUE

Procedure of NMDP-APC with ML deduced *px*-value
Step: Part-A, Initial Provisioning & Cluster Formation
01. Initiate Provisioning request to V2X function in PLMN from those VE desired to cluster. VE_{des}(*des* = 1⋯*Z*), *des* ∈ ∀ VE.

(Authorization, DC parameters, Radio parameter configuration)

02. Exchange neighbors in setting V2X L2 ID via C-V2X wireless link with those VEs are in the wireless rage. VE_{inr},(inr=1⋯R), $inr \in Z$.

03. **if** a Clustering Request Message is received from a neighbor. (any wireless propagation in-coverage VE_{des} are potential to initiate)

04. **then** initiate Cluster Establishment by executing NMDP-APC with trained *f(x)* provided a *px*-value. CH_j associated with the member VE_u(u=1⋯U), j and $u \in VE_{inr} \cap$ APC. (Determined CHs and the members enter In-cluster mode)

05. Communicate with PLMN via identified CH.

Step: Part-B, Re-clustering for Member Change

06-07 Identical provisioning process as in step 01-02 respectively.

08. **if** a Clustering Request Message is received from a neighbor.

(any wireless propagation in-coverage VE_{des} are potential to initiate)

09. **then** initiate Cluster Maintenance by executing NMDP-APC with trained *f(x)* provided a *px*-value, as in step 04.

10. **if** original CH stays, incoming VE becomes a cluster member. The CH updates the member, VE_u (u=1⋯U), $u \in VE_{inr} \cap$ APC, (The CH reflashes the member status and associated contexts.)

11. **if** an original member goes out the wireless range or received a Re-Clustering Request.

12. **then** initiate Cluster Maintenance by executing NMDP-APC with trained *f(x)* provided a *px*-value from the CH. CH_j only has the context of revised member VE_u (u=1⋯U), $u \in VE_{inr} \cap$ APC.

Step: Part-C, CH change and its Context Transfer via V2X-to-NW Relay

13. **if** a CH received a Clustering Request Message from a VE.

14. **then** initiate Cluster Maintenance by executing NMDP-APC with trained *f(x)* provided a *px*-value.

15. **if** an original CH is recognized to be replaced and CH_{NEW} is identified from the APC.

16. **then** CH_{OLD} and CH_{NEW} exchange the contexts of associated members information via V2X-to-NW relay.

CH_{OLD} replaced by CH_{NEW} by transferring the context of VE_k (k=1⋯K), $k \in VE_{inr} \cap$ new APC.

17. **if** this process failed, enters step 03 then restart initial clustering, evacuating from abnormal state and returning to normal procedure.

4.3 Simulation Results

4.3.1 Clustering Formation applying ML

This sub-section shows the sets of results of NMDP-APC simulation conducted on Matlab to observe cluster formation for different granularity parameter *px* values which derived from the proposed ML scheme. With one-minute real traffic data available at hand, we evaluated the clustering performance based on the data collected at different time points. Fig. 4.4-9 display the clustering results we obtained until the observation field is filled by using the traffic data collected at 35 sec, which is the middle part of the entire data. The reason we chose the data applied at 35 sec is our consideration of the practical radio propagation span of C-V2X. We set the observation range to within 1 km, *i.e.*, a 500 m radius, in this simulation as the reliable transmission range is considered to be a 500 m to 1 km radius [37].

Fig. 4.4 provides the result at t = 15 sec, the snapshot data from the overall observation in 35 sec. The top row in Fig. 4.4(a) shows the original vehicle distribution with the number of vehicles being indicated as N = 16. Fig. 4.4(b) shows the NMDP-APC clustering result with px = 2.0, which was assigned to produce a large group of fewest clusters. The proportion of NMDP-APC weight parameter φ represents a position component and ψ represents a velocity component, which are assigned as φ = 0.8 and ψ = 0.2 respectively, and the same values were applied throughout this simulation. The number of clusters formed in this condition is Clusters = 3. CHs are located mostly in the middle of each group with the associated members linked by dotted lines. From the results, it is easy to understand that few large-sized clusters were produced because a large px-value has been assigned. In this granularity assignment, it can be observed that vehicles are forced to be clustered due to the extreme px-value applied. Fig. 4.4(c) shows the Net Similarity value in the APC iteration process. By reaching the stable net similarity value, the APC process identifies the final shape of stable sets of clusters. Fig. 4.4 and 4.6 provide the result sets applying the same px-value, then the NMDP-APC clustering status is observed at t = 25 and 35 sec respectively during the overall observation time of 35 sec. It can be observed that a higher number of vehicles come into the observed space, and the NMDP-APC is producing the same levels of granularity as the same px-value has been applied.

Fig. 4.7(b) displays the NMDP-APC clustering result obtained with the granularity parameter px = 1.4 assigned to form medium sized clusters on the data at t = 35 sec where the observation space is filled. The number of clusters has increased to Clusters = 5 and it clearly shows that the number of members in a cluster tends to be smaller. Similarly, it is observed that CHs are located mostly at the center of the group and linked with the members in dotted lines. The rest of parameters are assigned in the same manner. Fig. 4.7(c) provides the evidence that the clusters have reached stability. Fig. 4.8(b) displays the clustering result obtained from assigning the granularity parameter px = 0.5, which is targeted at forming small-to-medium sized clusters on the same data at t = 35sec. As a smaller value of granularity parameter is assigned, the number of clusters has changed to Clusters = 6 and it is observed that members are more inclined to form independent clusters compared to Fig. 4.7(b). Fig. 4.9(b) displays the clustering result obtained by assigning the smallest granularity parameter value px = 0.1 at t = 35 sec on the same data used in Fig. 4.6-8. Now, the number of clusters largely has increased to Clusters = 13 and it is easily observed that the members are more isolated to other clusters as the extreme *px* value has been applied.

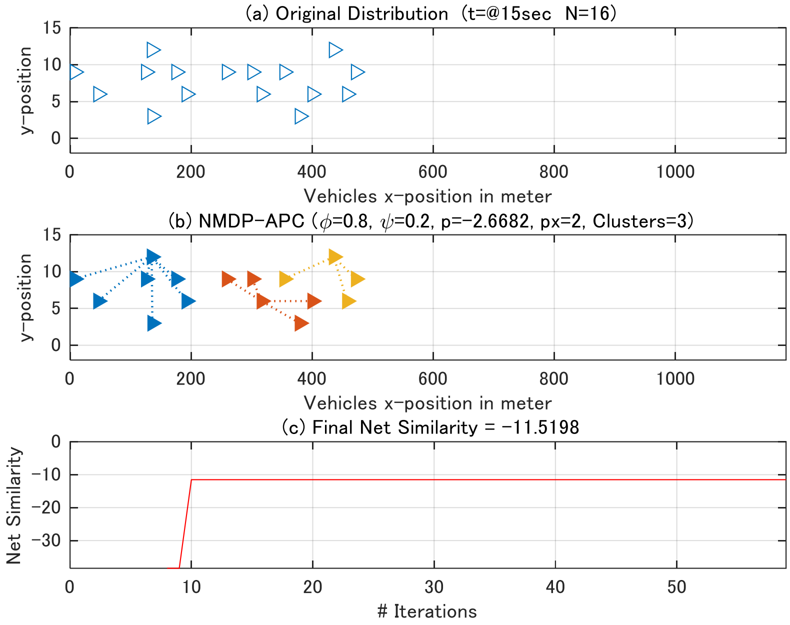


Fig 4.4 NMDP-APC Clustering Result, px=2.0, t=@15sec

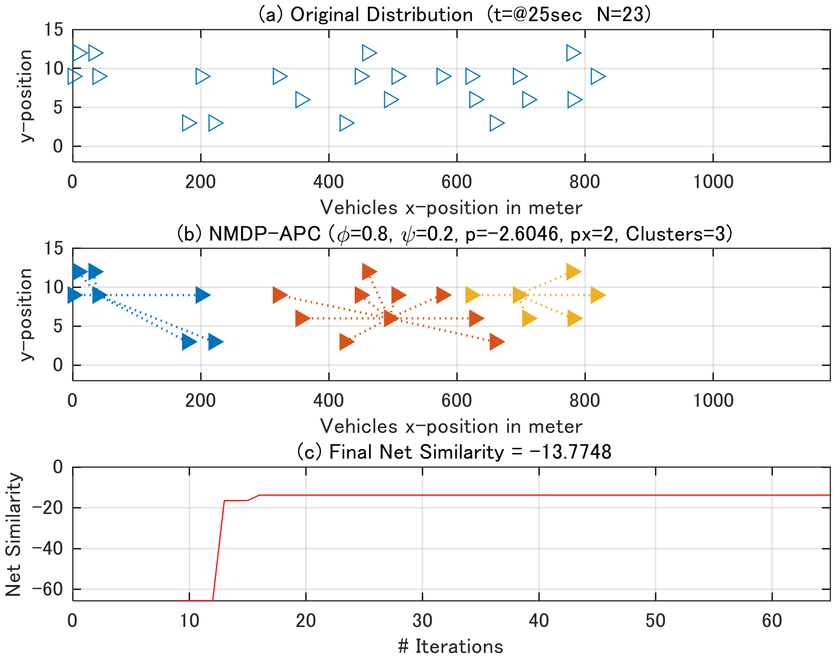


Fig 4.5 NMDP-APC Clustering Result, px=2.0, t=@25sec

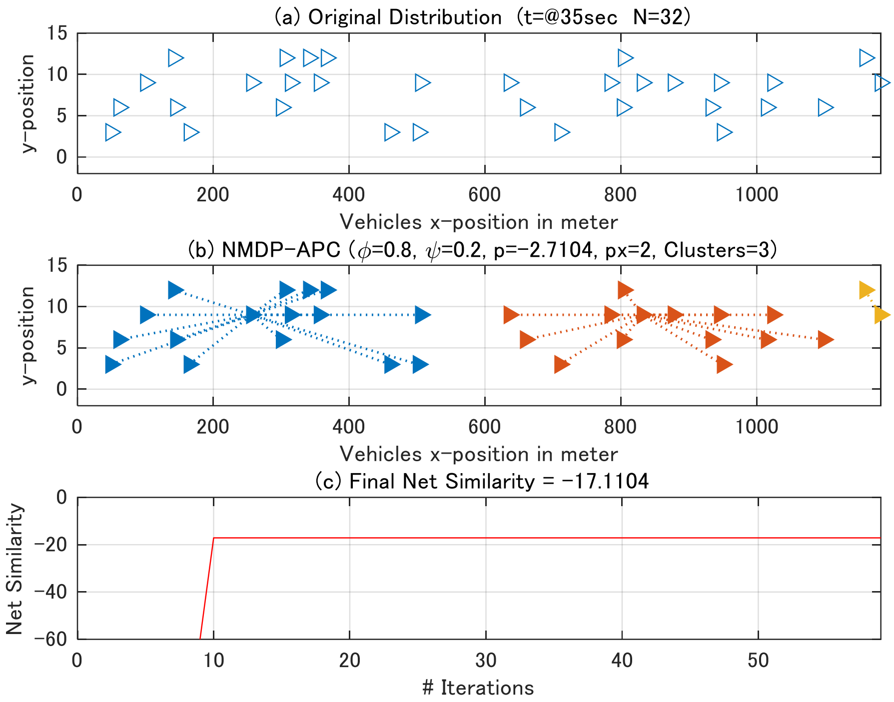


Fig 4.6 NMDP-APC Clustering Result, px=2.0, t=@35sec

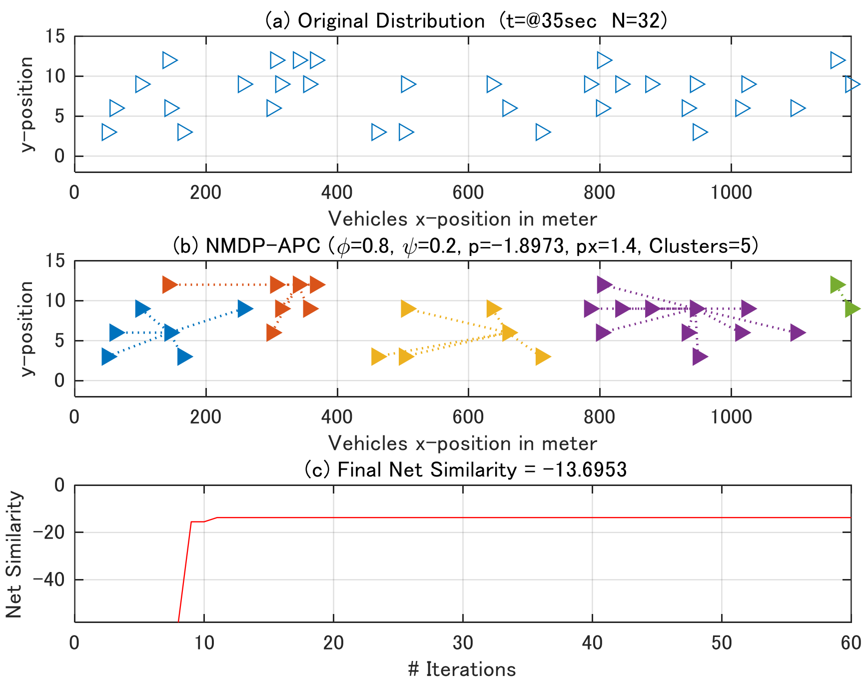


Fig 4.7 NMDP-APC Clustering Result, px=1.4, t=@35sec

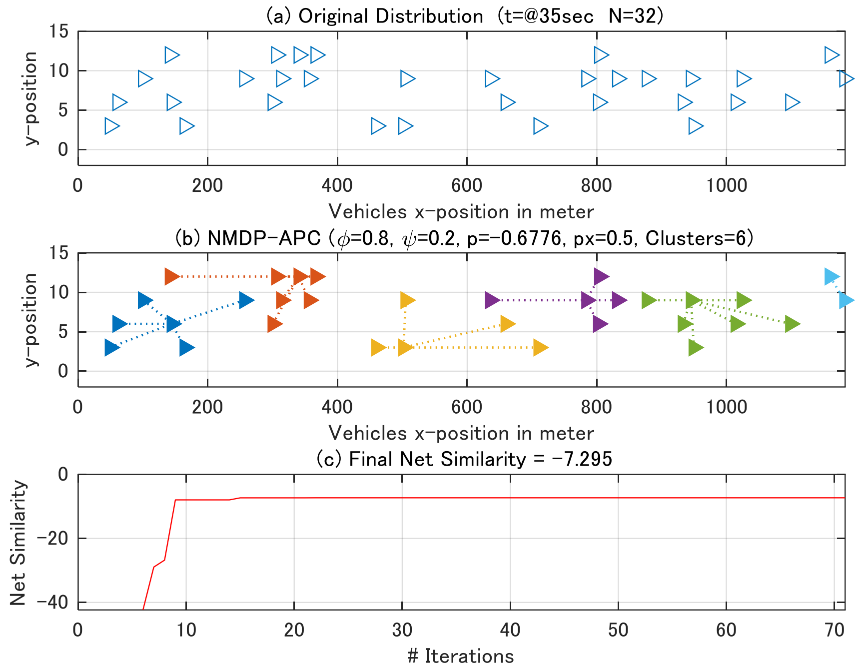


Fig 4.8 NMDP-APC Clustering Result, px=0.5, t=@35sec

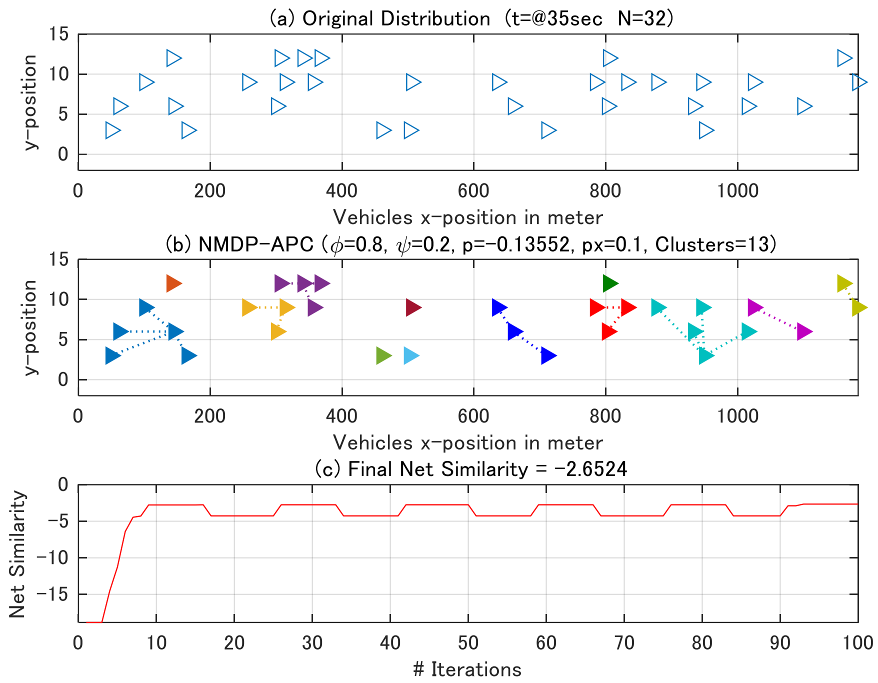


Fig 4.9 NMDP-APC Clustering Result, px=0.1, t=@35sec

4.3.2 Prediction Performance Evaluation

This part explains the prediction performance of the proposed scheme. In this part, we present sets of simulation results on the prediction performance of the soft-margin-based SVM-ML with GRBF kernel operation. We conducted these simulations in a Python coding environment as it has a series of well-prepared functions and libraries specifically for ML. As shown in Table-I, SVM and NN are the major candidate algorithms for classification. Simulation results obtained in using NN however provided an inferior performance compared to those in SVM in this simulation. We recognized that this is because our task vector $\boldsymbol{X}$ has only four dimensions and that algorithms using NN would be more applicable to a large dimensional problem case. Thus, we conducted further simulation focusing on the SVM by varying the formerly explained C and γ parameters.

Fig. 4.10 shows the prediction accuracy obtained by applying constant $C = 1.0$ and then varying two types of γ-parameters, which are SVM-default γ and SVM-scaled γ. The SVM-default uses γ = 1/feature number, *i.e.*, 0.25 as our task vector is 4 dimensional features. The SVM-scaled γ = 1/feature number multiplying by the variance of the task vector $\boldsymbol{X}$. As the result indicates, SVM-default γ provided better prediction performance compared to SVM-scaled γ parameter. It is notable that reached 100% of production accuracy with the sampling data given around at 140. It is remarkable that such high performance was obtained even with the small number of training data. This is important specifically in a dynamic vehicular application scenario. We recognize that this is attributable to the compact structure of proposed decision tree, which only consists of essential criteria with a four-dimensional task vector.

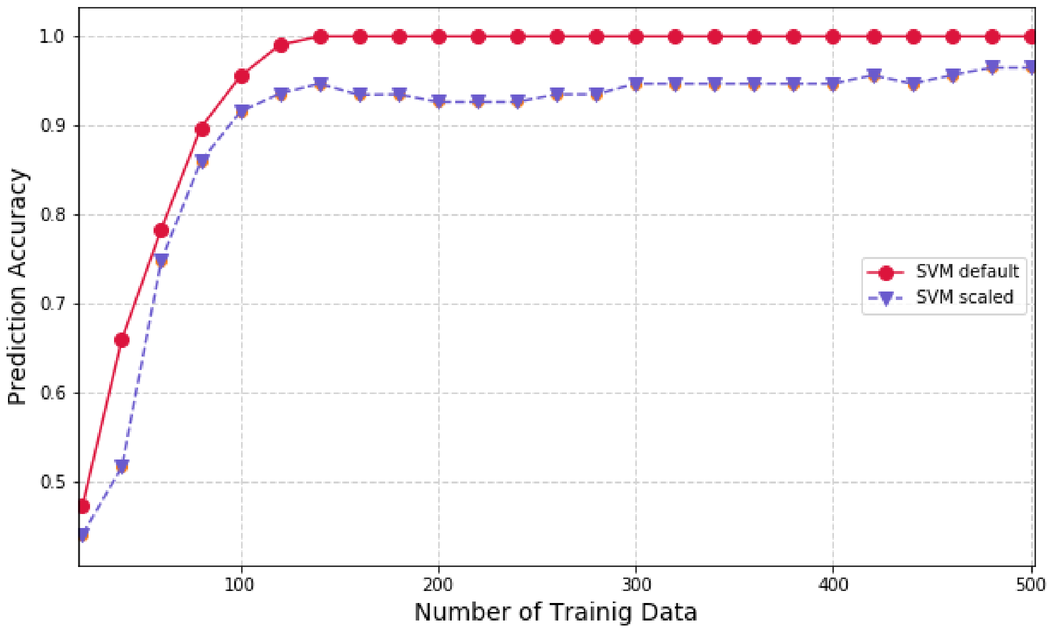


Fig 4.10 Prediction Accuracy in $C = 1.0$, with Applying Default-γ and Scaled-γ

Now in Fig. 4.11, we demonstrated prediction performance evaluation comparing default-γ with $C = 1.0$ where best performance obtained in above, versus scaled-γ by changing parameter-C to observe how the value influence the result. As the theoretical analysis reviewed, the cost-parameter C controls the balance of margin allowance and miss-classifications. In this simulation, by increasing C value, improvements in the prediction performance have been observed in SVM scaled-γ. However, 100 % accuracy has not been reached in SVM scaled-γ although increasing the C value and even large volume of training data applied. From these results, it can be noticed that applying combination of SCM default-γ with C=1.0 potentially provides desirable prediction performance in the simulation condition.

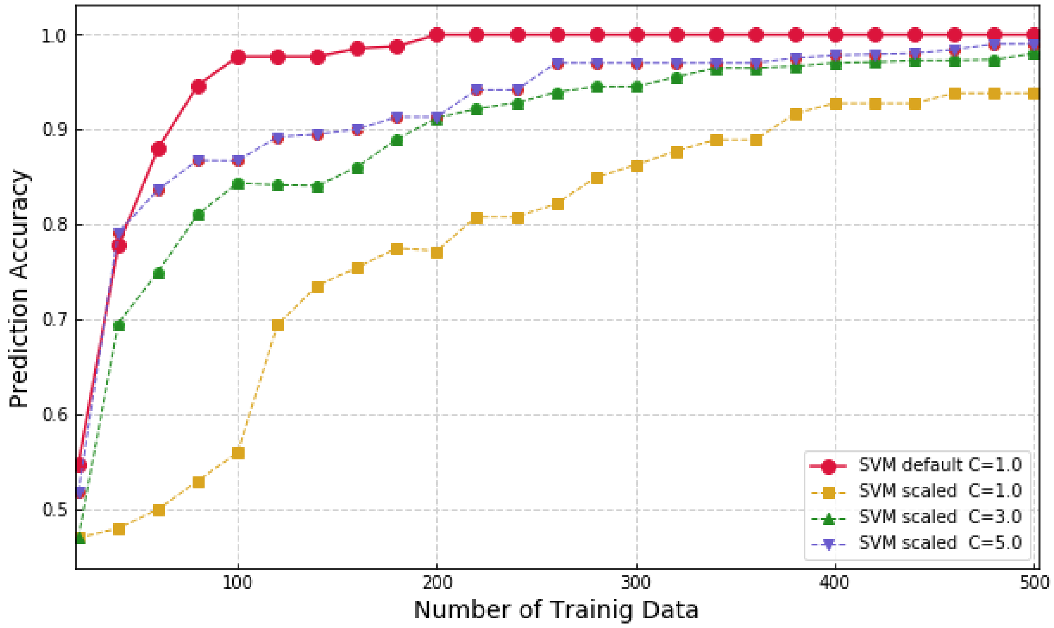


Fig. 4.11 Prediction Accuracy in Default-γ with C =1.0 Versus Scaled-γ with C Value

4.3.3 Statistical Average Access Latency

We observed the statistical average access latency in 5G compared with LTE through a simulation, as in Fig. 4.12. The access latency was statistically measured from a CH to gNB and eNB, *i.e.*, the Uu interface by assigning the target latency as the median value, *i.e.*, μ_{5G} = 4 ms and μ_{LTE} = 11.5 ms [49], [50]. The results are obtained by applying a truncated normal distribution [51] to avoid extreme and unrealistic values being produced. The simulation applied 10,000 times of sampling by changing different values of latency distribution variance σ^2 = 1.0, 5.0 and 10.0 then, we observed how the system stability influenced the average access latency in this simulation. The outcomes showed that the average access latency observed in 5G was 3.9 ms at σ^2 = 1.0, 5.8 ms at σ^2= 5.0 and 7.8 ms at σ^2= 10.0 respectively. In LTE access, the average access latency observed was 11.0 ms at σ^2 = 1.0, 12.0 ms at σ^2 = 5.0 and 15.4 ms at σ^2 = 10.0 respectively. A clear tendency observed was that when a large variance is applied, *i.e.*, the system is *instable*, the average access latency has deteriorated. The result can provide an understanding that the more stable access is provided, the less access latency can be achieved. Therefore, our objective of minimizing the number of accessing objects can also contribute to provide the reduction of access latency, contributing to the system scalability for avoiding access storms from individual vehicles and mobile MTC terminals potentially to be deployed in a large number in the future.

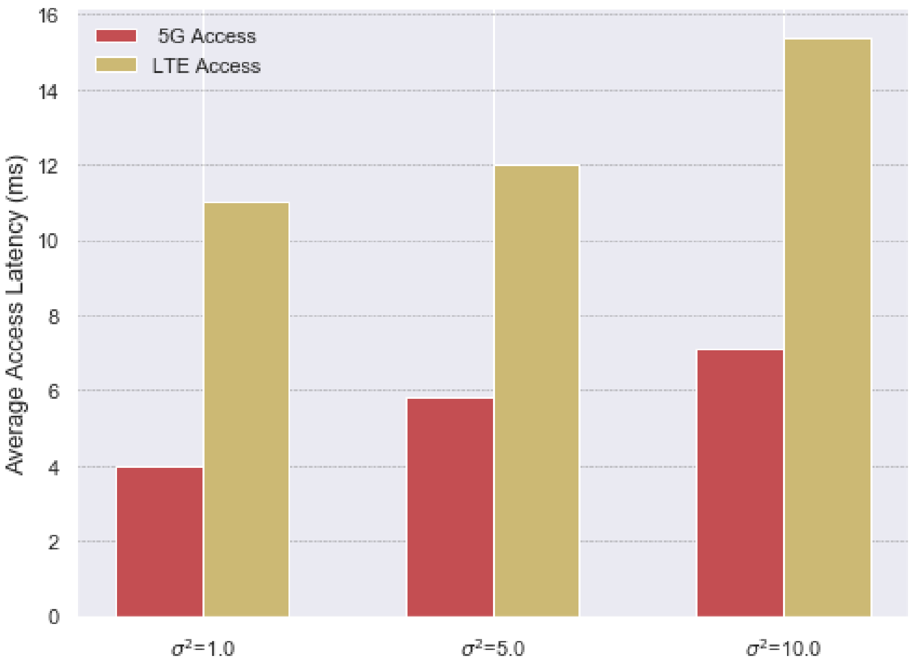


Fig. 4.12 Statistical Average Access Latency in 5G and LTE

4.4 Summary

This study has proposed an advanced VANET clustering scheme called NMDP-APC to form stable single-hop clusters in a distributed control manner. Specifically, in consideration of the real time motion dynamics of vehicles, two parameters, *i.e.*, inter-vehicular Euclidian distances and its velocities, are assigned as the metrics without associating them with any prediction values. In the NMDP-APC process, we controlled the clustering granularity level by adjusting the granularity parameter. Particularly, this study deduced the desirable clustering size, applying a ML scheme which employs soft-margin-based SVM-ML with Gaussian Radial Basis Kernel Function.

As ML has been known as a powerful tool in decision making, we applied it to find an ideal VANET granularity incorporating minimum sets of decision criteria. Relative to the limited and essential criteria, the ML prediction performance achieved satisfactory results with fewer training data. In addition, we designed a message sequence and procedure of NMDP-APC by associating them with the ML deduced clustering granularity. The procedure and sequence are enhanced on the existing 3GPP C-V2X specifications. With the PC5 interface, therefore, the proposed scheme can be easily implemented with emerging 5G cellular systems. Especially, the proposed scheme is designed through a distributed control approach to adapt nomadic vehicles' mobility. Through the simulations, the cluster formation and granularity control has been clearly observed through different values of the *px*-parameter. For improving ML prediction performance, two key parameters C and γ were adjusted to tune the prediction performance. The simulation results indicated that a particular selection of parameters resulted in a better performance. We also observed the performance with lesser access latency in a stable PLMN system via the simulation, which explains that the clustering capability is contributing to providing aggregation effects and thus it provides a stable system even when a large number of objects are connecting to a PLMN. Although this study was originally targeted to vehicular applications, it has potential applicability to any machine type moving objects expected to increase reflecting the growing demand for such applications in 5G and beyond. Future work should focus on enhancing the proposed scheme by applying more complicated scenarios and/or using different types of ML algorithms.

Chapter 5

5 Dynamic Server Selection Employing Machine Learning for 5G VANET

5.1 Introduction

This part of study conducted targeting to propose dynamic server Selection in employing machine learning specifically targeting to 5G VANET. Distributed computation is a strong demand of cotemporally computation and specifically vehicular computation need of using external resources because of large machine learning applications. However, such computational resources, latency of the access, and remaining resources are changing time to time. Therefore, for expecting to use appropriate resources on-time these dynamics has to be considered during the target resource selection process. We pursued this question for finding an answer in applying a ML scheme. Extensive simulation also performed with analysis of fundamental mechanism of applied ML algorithms. For targeting to real application, we also observed prediction performance in missing data scenario. Eventually, the ML prediction performance of missing data and missing data handling capability of MLs also evaluated in the simulation.

5.1.1 Objectives

The objectives of this study targets to realize various and extensive application needs of vehicular computation in distributed manner. It has been known that the conventional server selection has been performed in static and heuristic approach. For novelty, this study employs a power of machine learning incorporating real time network status. Key contributions on this study can be summarized as follows;

#1 Designed and proposed a distributed ML system for server selection in considering network dynamics with carefully selected essential evaluation criteria.

#2 Applied conventionally well-known SVMC as well as newly introduced GMBC-based classification algorithms and carefully examined prediction performance while varying the training data volume.

#3 Isolated the learning offline process from the online ML execution process for enabling a time critical system regardless of its sampling volume.

#4 Incorporated a missing data handling capability as un-obtained data cannot be excluded in practical usage and this causes stagnation.

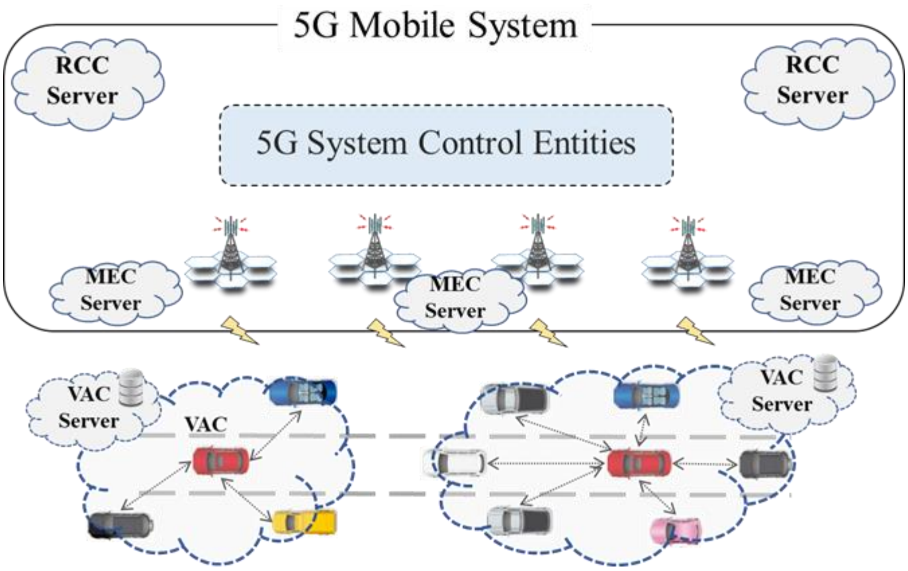


Fig. 5.1 Distributed Resource Selection from Vehicles

5.1.2 Related Works

Amid the above circumstances, various research activities on vehicular communication with 5GS have been widely conducted. A well-organized article [39] summarizes an overview of salient features of vehicular communication with 5GS and indicates that proximity services, MEC and network slicing are key technical components. El-Sayed *et al.* emphasizes the importance of Edge Computing (EC) over central cloud architecture as EC pushes the computing capability to the network edge, contributing significantly to improving response time with resource optimization [40]. Ning *et al.* studies an efficient access management scheme and indicates MEC as an essential capability for minimize latency and handling a large number of SIVs [41]. A low latency networking feature has been specifically demanded more than ever, with various time-critical vehicular applications being envisioned. Mobile Vehicular Cloud (MVC) is a typical realization of vehicle communication with the MEC function placed in the radio access network [42], [43]. Focusing on task-offloading of Vehicular Edge Computing (VEC), [52] proposes a multi-access capability, introducing a three-level hierarchical framework consisting of the cloud-enabled control layer, MEC layer and multi-access connected cloud layer. A vehicular fog network is a realization of distributed computing over vehicles, and a study [53] applies ML to identify the target fog and predict task allocation over the fog vehicles, realizing interactions between the central cloud server and local fog entities. Although such studies have consistently addressed the importance and advantages of edge computing, a system-wide scheme addressing *how* to select a best target server among the candidates has been missing.

Ye *et al.* investigates potential application of ML over vehicular communication in the era of 5G [24]. They summarize the variations of ML and introduced potential fields of applications while Data-driven decision making in vehicular networks and intelligent resource management are stated as potential ML application fields. An extensive study of ML application to VANET clustering can be seen in [54]. This considers vehicular dynamics in real traffic data and found best clusters using a ML classification scheme. In some more advanced manner, Liang *et al.* studied a ML flamework toward intelligent vehicular networks [52]. They categorize ML variations and shows areas where learning-based decision making is applicable in high-mobility vehicular networks. They also indicated brief architecture with ML agents through an example and implied that ML can be a promising tool to learn the *network dynamics* and intelligent decision making in vehicular networks.

In more recent studies, the concept of federated ML scheme has been introduced in vehicular computing. This concept provides distributed and interactive learning process between the vehicles with external ML nodes such as one in a Road Side Unit (RSU) or in a central crowd server [45], [57]. With regard to federated ML in vehicular edge computing, D. Ye *et al.* [55] proposes a ML federation model where individual vehicles performed local deep neural network at local data, then aggregate a global neural model at a central server for image classification on captured video. This study also provides a valuable list of federated ML variations in a distributed network. However, it assumes a single central server and does not sufficiently mention how to select target servers. Instead, J. Cao *et al.* proposed a federated ML scheme between individual vehicles and RSUs in evaluating the learning cost and delay functions [56]. They apply a distributed Q-learning scheme to find near-optimal computation offload, avoiding an NP-hard problem, from vehicles to either an RSU or central cloud server [57]. This uses reinforcement learning with an agent possessing a specific policy for making an offloading selection by maximizing the reward feedback. Although it proposes offloading decision, they have not addressed how to evaluate the results as it is an unsupervised ML approach. No sufficient consideration has been given to network dynamism with merely static resources and latency values assigned. According to our observation, these are constantly changing and therefore we need to take network dynamics into account.

After reviewing these works, we decided to take into consideration network dynamics as well as vehicles' mobility dynamics for appropriate server selection in our study. Therefore, we designed the proposed ML system able to minimize the learning time and to use less sampling data. As this requires a highly capable classification-engine and there are several alternatives. From the candidates, we applied the following ML algorithms as they are known specifically applicable for low dimensional tabular based data [28] and uses fundamentally different classification approach. Firstly, extensively studied and widely applied algorithm, Support Vector Machine Classification (SVMC). This finds a hyperplane for the classification and we also have some experiences on the extent of apply. Then relatively new algorithm called eXtream Gradient Boosting Classification (XGBC) [58], which is an extension of Gradient Boosting Machine Classification (GBMC) algorithm. The concept of *boosting* trains models in succession, in concatenate and feedback the result of weak model then, each new model being trained to correct the errors made by the previous steps. The fact that data analysis using XGBC were recently awarded a prize in the KDD contest also motivated us to select this as one of the candidate algorithms for our evaluation.

5.1.3 Contributions

This study proposes a novel ML-based scheme for selecting distributed computational resources, *i.e.*, servers, giving consideration to network dynamics in reducing the leaning time and yet aiming to keep a high level of prediction performance for mobile SIV. We designed a system specially focusing on minimizing learning latency for adapting real time application of SIVs in 5G C-V2X architecture. The contributions of this paper can be summarized as follows:

1. Designed and proposed a distributed ML system for server selection in considering network dynamics with carefully selected essential evaluation criteria.
2. Applied conventionally well-known SVMC and newly introduced GMBC-based classification algorithms and carefully examined prediction performance while varying training data volume.
3. Isolated the learning offline process from the online ML execution process for enabling a time critical system regardless of its sampling volume.
4. Incorporated a missing data handling capability as un-obtained data cannot be excluded in practical usage and this causes stagnation.

5.2 Proposed Decision Flow and Server Selection Scheme

5.2.1 System Architecture in 5G C-V2X Server Selection

This Fig.5.2 plots typical computational server locations onto 5GS architecture [59]. Vehicular networks have a potential to form VAC server in the VANET. MEC is co-located in the Next Generation-Radio Access Node (NG-RAN). RCC is placed in system depth beyond the cellular system as a Data Network (DN) at the end of the network.

The far-end RCC is typically used for the tasks requiring large computation and for acquiring a wide-range of information, such as in 3D map data collection and/or central processing of pre-processed results from edge servers [55]. MEC is typically placed near a network edge such as in a RAN. A scenario of collision avoidance at an intersection would be a typical example. As this scenario requires various types of information in low latency, not only on vehicles but also on assigned pedestrians and state of intersections. Instead Vehicular Ad hoc Cloud (VAC) is formed in a group of proximity vehicles and it expects extremely time-critical communication such as inter-vehicular waring notice, collision avoidance and vehicle clustering management [54]. The VAC will largely contribute to providing various inter-vehicular services and applications through flush data repository dedicated to the group.

Table 5.1 Server Computing Characteristic Summary

Characteristics	VAC	MEC	RCC
Typical expected Service use case	Inter vehicular proximity	City & Metropolitan	Global Wide
Available Resource & Storage Size	Low	Middle	Abundant
Latency, Response Time	Extremely Short	Short	Large
Required Signaling amounts	Low (Inter vehicular)	Middle (to the RAN)	Large (System wide)

In this study, we assumed a 3-server scenario as a typical case, but this study is extendable and applicable to any number of candidate computational locations once necessary information set are prepared. This flexibility motivated us to apply ML for decision making. Table 5.1 summarizes the characteristics and properties of the above-mentioned three types of cloud computational servers. The use cases and scenarios of server will vary depending on the service and the server will be selected at each session initiation. We will use these characteristics later in the ML decision flow in the following section.

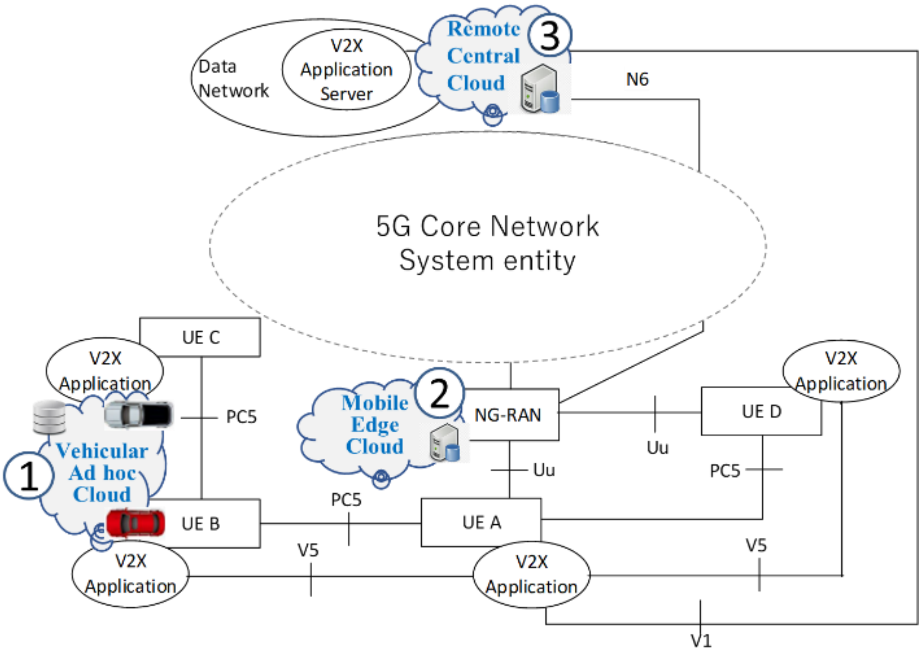


Fig.5.2 5G Cellular and C-V2X System with Three Types of Servers

5.2.2 Proposed Task Vector and Decision Flow

The issue of classification defines the relation of the objective variable y through a set of explanatory variables in task vector $\boldsymbol{X}$. In this work, we used a deliberated decision flow to find a particular server, *i.e.*, the objective variable y in considering the given decision criteria. The mapping relation can be denoted as $\boldsymbol{X} \rightarrow y$, where $y = \{-1,0,1\}$ to distinguish three types of servers. The labeling of y = -1 represents VAC, $y = 0$ represents MEC and $y = +1$ represents RCC server. We designed the task vector $\boldsymbol{X}$ with the following set of explanatory variables.

$$\boldsymbol{X} = (x_{in}, p, u_{rq}, u_{rm}, t_{lo}, t_{of}, L_{lo}, L_{of}) \tag{5.1}$$

In (5.1), x_{in} represents the input task size of a service from a vehicle, and p allows potential initial preference of the server alternative. The value of $p = 0$ indicates a local service execution preference, and $p = \{1, 2, 3\}$ shows an offload service execution preference, which number indicates VAC, MEC, RCC respectively. Second attribute considers the status of required and remaining resources, u_{rq} and u_{rm} respectively. The u_{rq} represents the required computational resource to execute a given task and u_{rm} represents the remaining resources of the server on the request. The third attribute considers the latency aspect by t_{lo} and t_{of}. The t_{lo} represents the summation of overall latency in using a local server and t_{of} represents the summation of the latencies in using an offloading server. The fourth attribute considers the signaling loads required for local execution and offload server execution, shown as $L_{lo}$and L_{of} respectively. Having the above structure of explanatory variables, a server selection is executed using the decision flow shown in serial-offloading with a three-layer server hierarchy. Therefore, an offloading decision is considered either from VAC to MEC or MEC to RCC. However, in order to incorporate network dynamics, the above process needs to be performed with knowledge of up-to-date resource consumption and congestion status. Therefore, this study asked for the power of ML. The following part introduces theorical backgrounds of the algorithms applied.

This section explains proposed decision flow of dynamic server selection scheme. We introduce a four-layer decision flow in associating essential attributes to deduce the target server as in Fig.5.3. This should be constructed as a concise one in consideration of the decision criteria, otherwise the performance will be impaired and more computational time will be required.
The first attribute allows incorporation of an initial preference in choosing a specific server. For example, time-critical inter-vehicle proximity communication, *e.g.*, emergency break information and trajectory information for collision avoidance will be executed locally, therefore VAC virtual server will be preferred. On the other hand, a centralized control system will be preferred in such cases as traffic light control and safer vehicle traffic control, which are executed with knowledge of vehicles' motion dynamics with road-side pedestrians' information at an intersection. In such a case, selecting a server located at the mobile network edge, *i.e.*, MEC server will be a reasonable choice. In a wider scope, for obtaining street 3D map information required to collect a large amount of information into a single central server [56], RCC will be preferably used for a global perspective application.

The second criterion of the decision flow is resource availability at the target server. This evaluation needs to evaluate if the local server resources are enough in consideration of the input volume. Hence, in the case of $u_{rm} > u_{rq}$, a local server should be chosen; otherwise a third criterion should be considered. The third criterion evaluates the degree of latency at local and remote executions. In the case of $t_{rm} > t_{lo}$, the local server should be chosen; otherwise the last attributes should be entered. The fourth criterion evaluates the volume of system loads, considering how much signaling is exchanged for selecting a particular server. If $L_{re} > L_{lo}$, the local server should be chosen; otherwise remote server should be selected for the task.

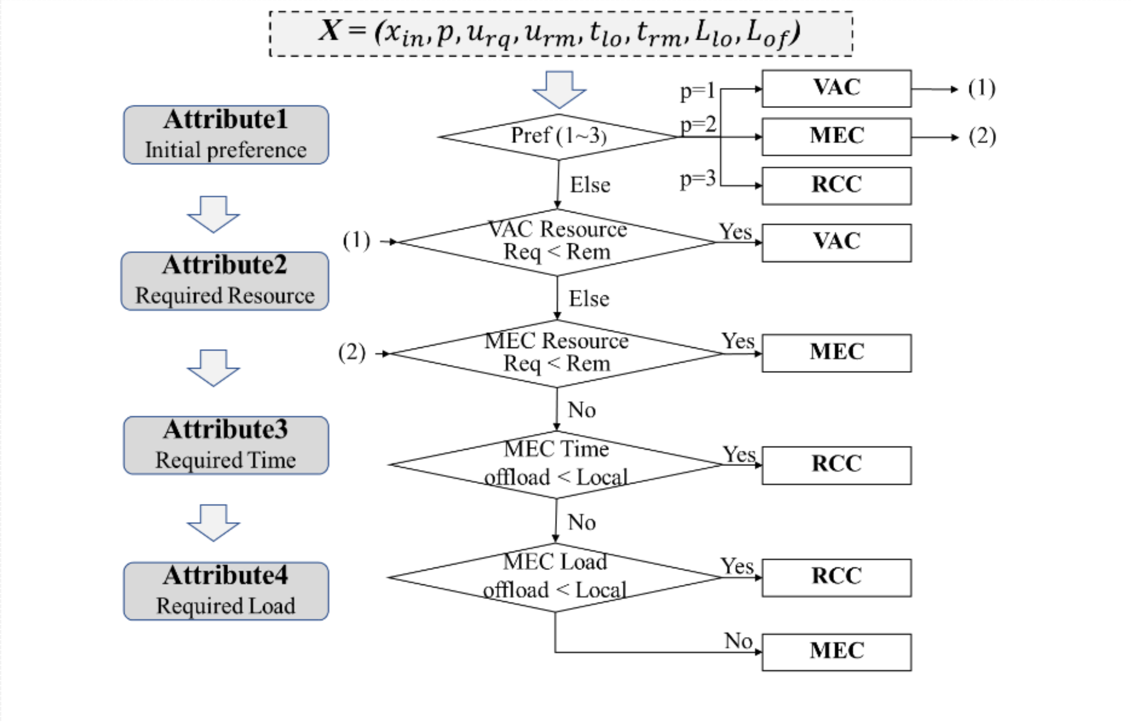


Fig.5.3 Decision Flow with the Criteria

5.2.2 Offline Learning and Online ML Execution Process

The prediction performance of a ML is proportionally improved when a larger volume of training data volume is given. However, this learning time is critical for vehicular application as vehicles are in motion. Therefore, we developed a learning scheme that isolates the execution phase from the learning phase as outlined in Fig.5.4. The scheme designed where the offline learning process should be performed on the network edge and the online execution process at the vehicle side. The trained decision function $f(\boldsymbol{X})$ is deduced from a large amount of training data that composes network dynamics by explanatory variables in $\boldsymbol{X}_i$ except x_{in} and p. The statistical record of data size x_{in} can be sent from vehicles to the leaning agent with location update messages that periodically inform the position of mobile terminals to the network. This is also applicable to the preference value p which simply consists of binary information.

The online process is executed at the timing of a service execution for selecting a particular server by applying the trained decision function derived from the offline process. The concept behind this mechanism is similar to that of federated learning, and the proposed scheme fundamentally contributes to the reduction of learning time in the execution phase. The performance of the proposed scheme is carefully examined in the simulation part. The offline learning process of

Step1 collects a large number of explanatory variables of task vector $\boldsymbol{X_i}$. Step2 is a learning process to deduce a decision function $f(\boldsymbol{X})$ based on the given task vector so far provided. Step 3 provides an occasion for updating the $f(\boldsymbol{X})$ associating additional new data of the explanatory variables. Step 4 evaluates the data updates of explanatory variables in task vector $\boldsymbol{X_i}$. We suppose this learning process is performed at an agent placed on the network edge, *e.g.*, gNB or MEC located in the vicinity of vehicles. The online process is executed at the timing of actual service execution phase upon a session initiation. The server selection is performed based on the decision function $f(\boldsymbol{X})$ derived in Step 3. The online process also includes the evaluation step to check if the initial server selected meets the target criteria. Therefore, this scheme can identify a best server based on the trained function to meet the criteria in limited processing time.

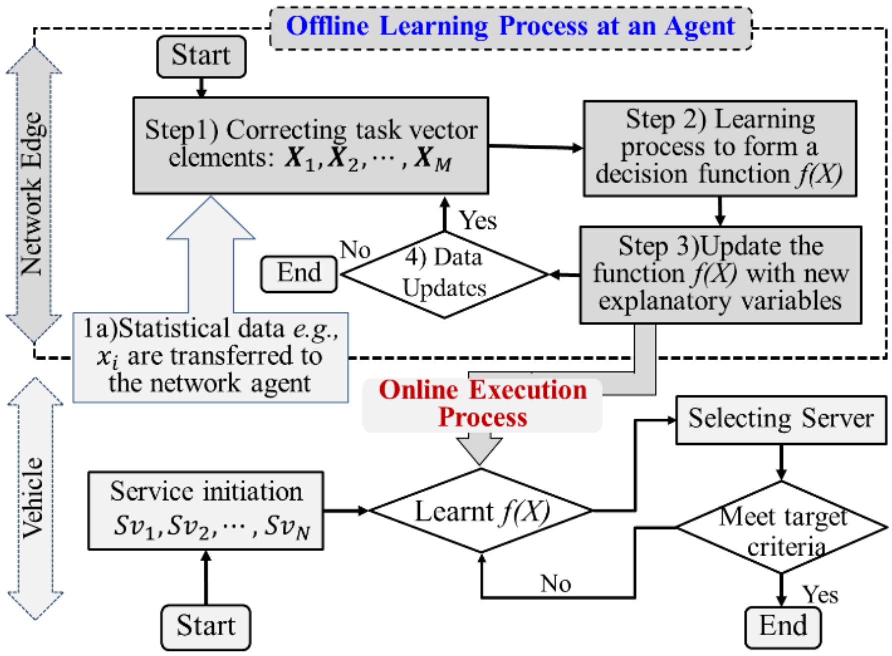


Fig.5.4 Proposed Server Selection Scheme in Online and Offline

5.3 Simulation Results

5.3.1 Parameter Setup and Preparation

This section explains the parameter setup and preparation for the simulation. We applied a Python 3.0 simulation platform and executed these simulations on a PC with 2.6GHz CPU rate in 16 GB onboard RAM. Through these simulations, we examined the following four aspects. The first simulation examined the performance of prediction accuracy over the three types of ML algorithms. The second simulation observed the missing data handling performance over the algorithms. Third simulation evaluated latency contribution on the proposed system. Last simulation observed an advantage of a 3-server model versus a conventional 2-server model.

Table 5.2 Simulation Parameters

Parameter	Mean Value
Remaining & required resources at RCC	40 GHz
Remaining & required resources at MEC	20 GHz
Remaining & required resources at VAC	5 GHz
Required latency for using RCC	100 mil-sec
Required latency for using MEC	50 mil-sec
Required latency for VAC	3 mil-sec
Signaling volume for RCC	25 signals
Signaling volume for MEC	15 signals
Signaling volume for VAC	3 signals
Variance of Gaussian distribution	$1.0, 5.0, 10.0$

Table 5.2 summarizes the assigned simulation parameters we set by referring to several sources. For the resource values, we referred to [60] and, for the latency values to several internal information sources. We also counted the number of messages in the signaling volumes given in 3GPP specifications [04], [48]. We applied x_{in} 500kbps and x_{ou} 100kbps in each vehicle with 1 sec interval. Other mean values assigned were the network transmission rate T_r 5 MB/s, CPU processing rates at VAC: 1 GB/s, at MEC: 5 GB/s and at RCC: 10 GB/s in referring to [60]. Randomly generated 15,000 Gaussian distributed synthetic data were prepared, with variances of 1.0, 5.0 and 10.0 for each of the explanatory variables of $\boldsymbol{X}_i$. We set a very low probability of assignment of p, which is 0.01% as the system more autonomously selects a server. For observing a wide-range of performance, up to 10,000 data were used for training data and 2,000 for test data for prediction evaluation. For observing the performance in a short range, up to 1,000 training data and 200 test data were assigned. The extensive results obtained are described below from different aspects

5.3.2 Prediction Performance Evaluation

This part explains the prediction performance evaluation from the proposed machine learning. Fig.5.5 displays the prediction accuracy on the three algorithms XGBC, GMBC and

SVMC observed for a wide range of the sampling volume from 1,000 to 10,000 in steps of 1,000. Each point of the result represents the average prediction accuracy obtained after data computation was repeated 100 times for excluding random variations. Hence, the graph shows a clear tendency of the result despite the use of synthetically generated data. There are sets of parameters for tuning on each algorithm, but we have touched them minimally. The default parameter setting is applied on XGBC and SVMC. The default SVMC uses a Radial Basis Function (RBF) kernel. GMBC applied the learning rate of 0.9, and the max features and max depth of 3 respectively. Over the observation range, a quite stable and constant performance is observed in each algorithm. XGBC keeps a better performance compared with the other two algorithms; the more training data are applied, the better performance is achieved. Note that the scale of y axis ranges from 0.8 to 1.0 in this figure, and the best performance of XGBC was shown as 0.963 in 10K samples.

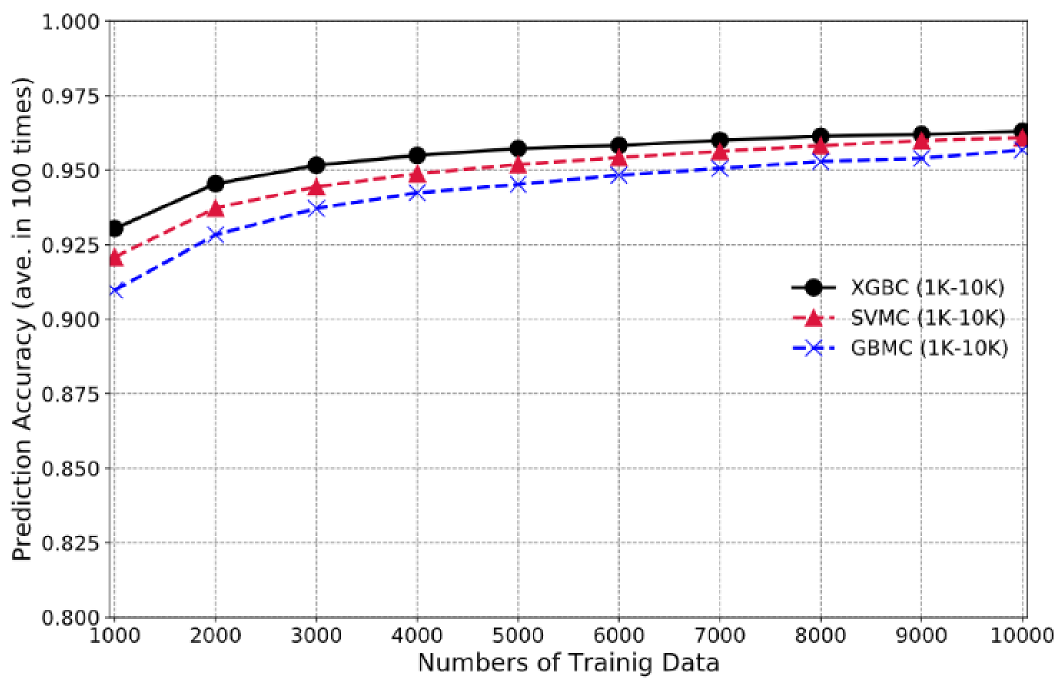


Fig. 5.5 Prediction Acracy over the Algorithms in Wide Range

Fig.5.6 shows the result of downsized training data one-tenth the volume used in the previous simulation, as sampling numbers are linearly proportional to the learning time. The number of training data spans now from 100 to 1,000 in 100 steps and y-axis ranges from 0.5 to 1.0. In the same manner, a clear performance tendency is observed while XGBC keeps its superiority over the competing algorithms during the sampling span. The prediction performance of XGBC marked 0.903 at 400 sets of training data and 0.929 at 1,000 sets of training data.

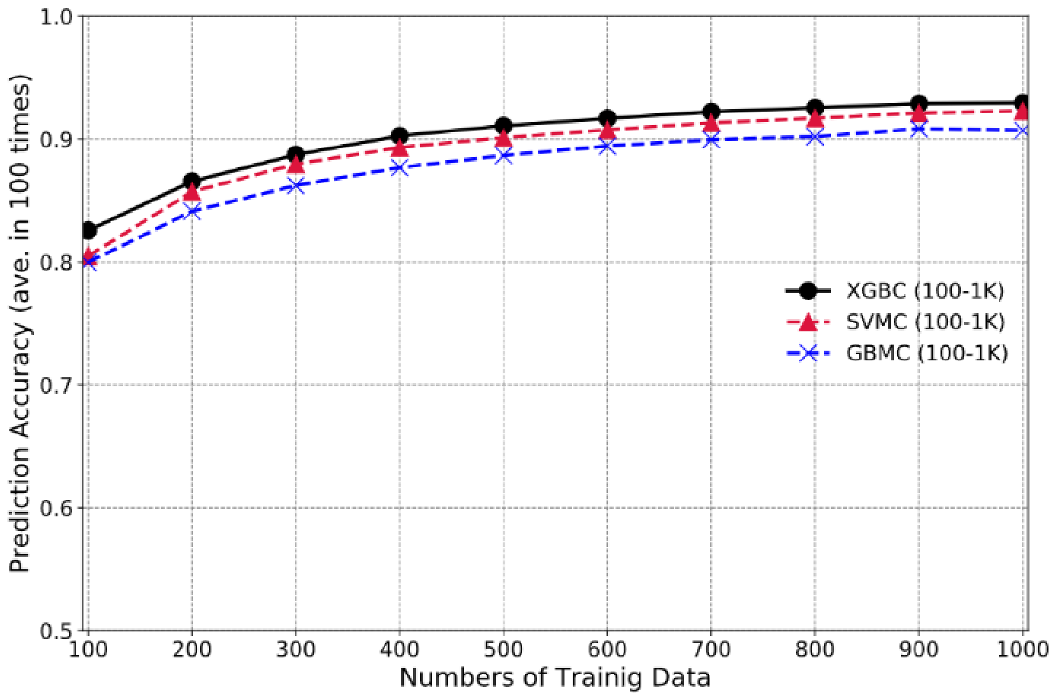


Fig.5.6 Prediction Acracy over the Algorithms in Short Range

5.3.3 Prediction Performance Evaluation with Missing Data

Next, we conducted performance evaluation in a scenario associated with missing data in task vector $\boldsymbol{X}_i$. This reflects a situation where missing data handling should be considered because obtaining all necessary information on time is unrealistic. Our training data were composed of large data sets, and the maximum volume of data has 8 times that of 10,000 tabular size. Furthermore, the data collection time in a vehicular application is critical as the vehicles are in dynamic motion. A reliable ML scheme has to be prepared to address these missing data scenarios, and this is what we observed in this simulation. Fig.5.7 shows the prediction performance in three ML algorithms with 10% contamination of NaN (Not a Number), *i.e.*, missing data randomly distributed in the task vector $\boldsymbol{X}_i$. The number of training data spans from 100 to 1,000 in 100 steps and y-axis ranges from 0.5 to 1.0. The imputation represents the missing data handling scheme of a ML algorithm. In XGBC, a default missing data handling scheme is embedded, which uses a pre-assigned brunch direction. However, neither GMBC nor SVMC Python library has such an embedded default missing data handling scheme. Therefore, we applied KNN (K Nearest Neighbor) to impute the missing data. The figure still shows the superiority of XGBC across the overall sampling range. Interestingly, SVMC shows the lowest performance in the smaller training volumes but it overtakes GBMC around at the data size of 800 in this simulation.

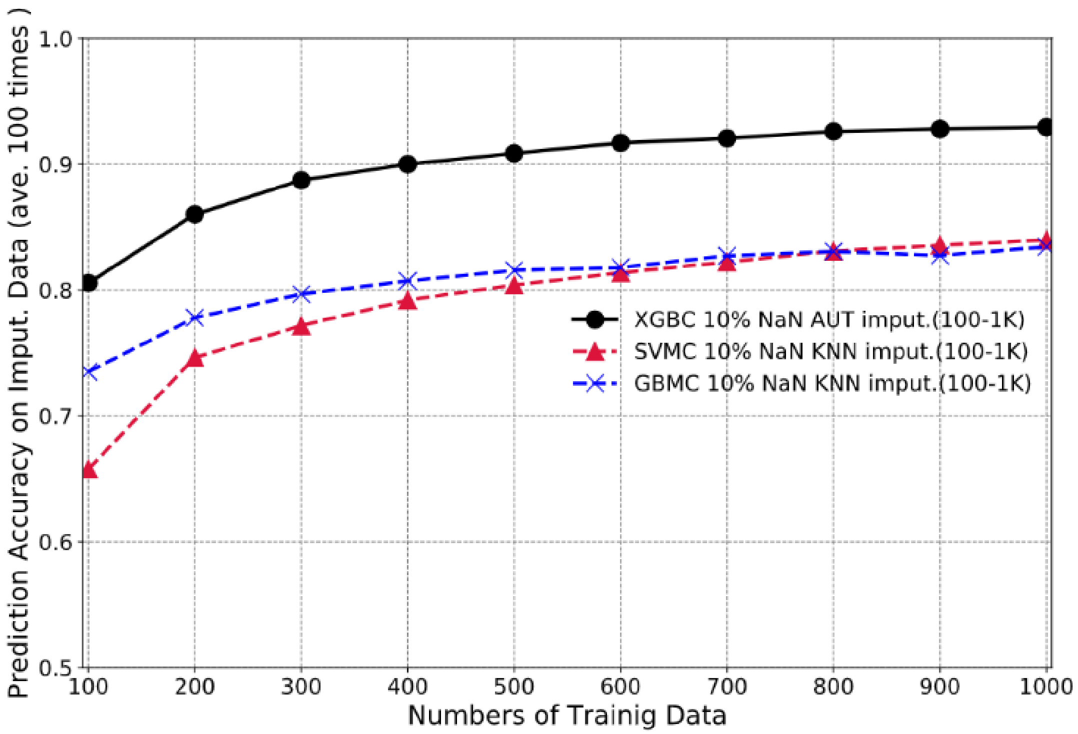


Fig.5.7 Prediction Acracy over the Algorithms in 10% Missing Data

Fig.5.8 displays prediction performance focusing on XGBC with 10% missing data contained and different types of missing data handling schemes applied. AUT imputation represents the default setting. SPL imputation takes a simple average of the column and replaces it with the missing data. KNN imputation is conducted as mentioned above. In this simulation, the performance of XGBC default missing data handling scheme shows better than the other two algorithms across all the sampling size. Hence, the embedded default missing data handling scheme is presumed to be a reliable one. The best performance of XGBC in 10% missing data was shown as 0.92 in 1,000 samples.

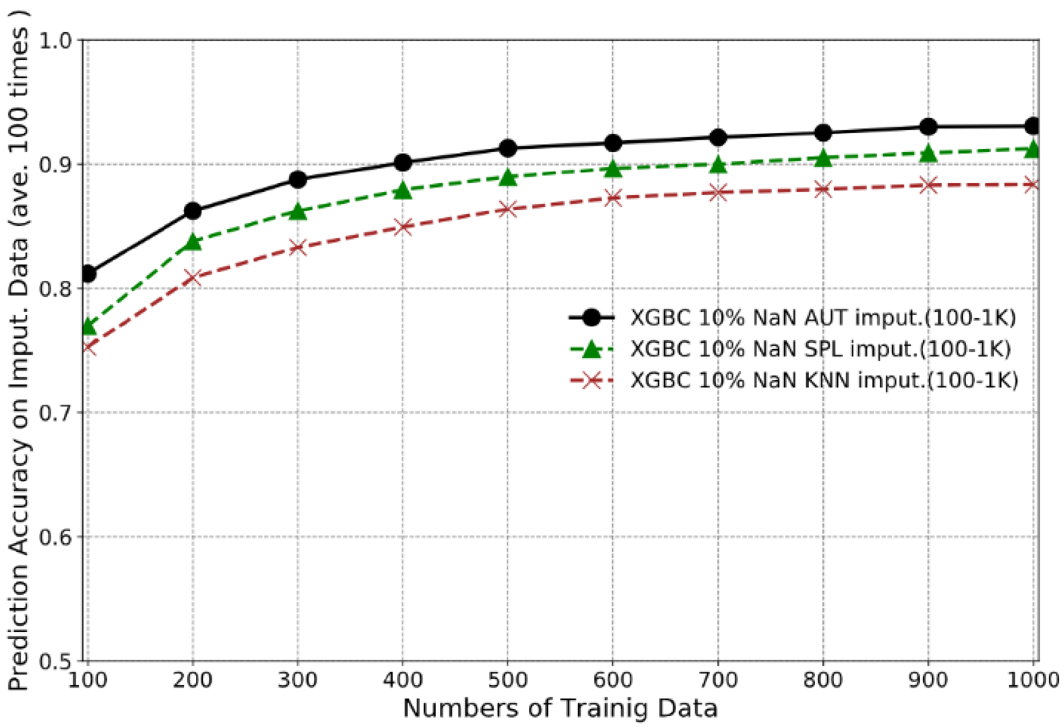


Fig.5.8 XGBC Accuracy Imputation Variations in 10% Missing Data

5.3.4 Evaluation on Latency Contribution on Proposed Scheme

This part explains latency contribution through proposed mechanism evaluating the execution time versus processing and learning time. Fig.5.9 shows the latency contribution derived from isolating the offline learning process from the online execution scheme as depicted in Fig.5.4. We observed the performance at the training data volume of 100, 1,000, 5,000 and 10,000. The offline process time includes the time for processing the given training volume of $\boldsymbol{X_i}$, the learning time and the time for forming a learnt function $f(\boldsymbol{X})$ from the data set. The online process measures the executing time in calling the learnt function in the simulation code. All the data are made in a Gaussian-distributed synthetical manner. The left axis measures the latency component of the offline process shown as blue bars. According to this graph, the processing time clearly increases in proportional to the sampling data volume. ML can improve the performance as the sampling volume increases, and this reduces latency in the conventional scheme. However, as the proposed scheme isolates the learning process from the ML execution process, it only requires the time necessary for the part shown in red, which is remarkably constant despite the changing sampling volume. Numerically, the averaged execution time is around 13 mil-sec for the overall sampling range in the simulation. This result shows a prime advantage of the proposed scheme and an evidence of its applicability to time critical vehicular applications.

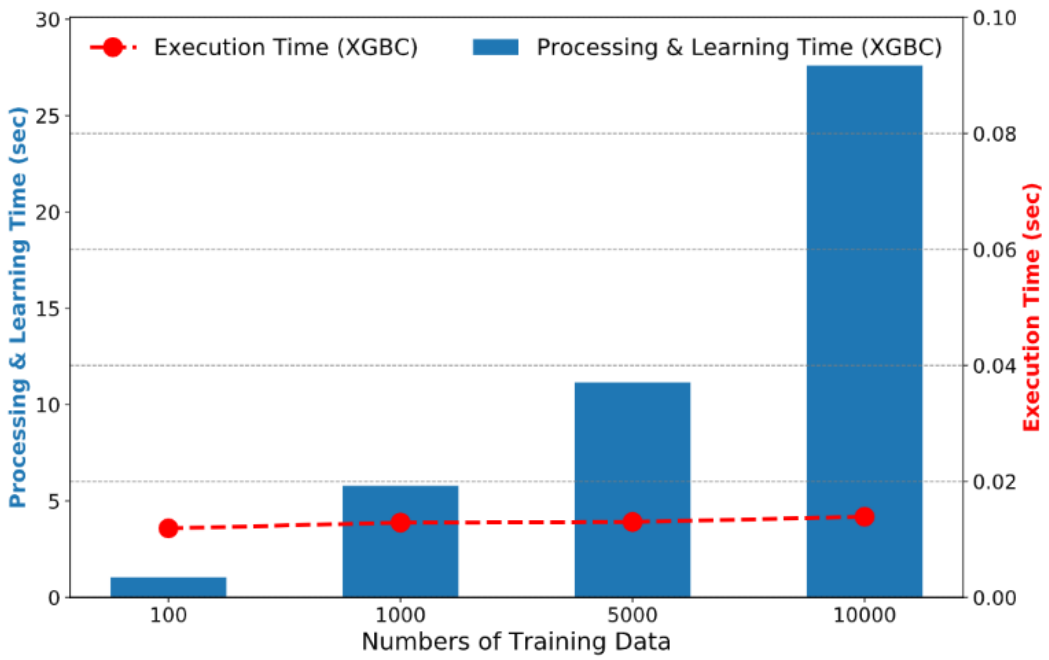


Fig.5.9 Execution Time v.s. Processing & Learning Time

5.3.5 Evaluation on Access Latency on Proposed Scheme

This part evaluates the access latency contribution on the proposed scheme. Finally, Fig.5.10 shows the averaged latency of the proposed 3-server model and conventional 2-server model obtained from synthetically generated 15,000 sets of data with variances applied at 1.0, 5.0 and 10.0 respectively. The 3-server model consists VAC, MEC and RCC in the system components instead, 2-server model consists MEC and RCC only. The outcome clearly indicates that the 3-server model shows better latency contribution compared to the conventional 2-server model *i.e.*, a system without VAC, which contributes to shortening the access time. Numerically, the contribution of the 3-server model observed through this simulation was the reduction of latency up to 68% over the 2-server model at variance 1.0.

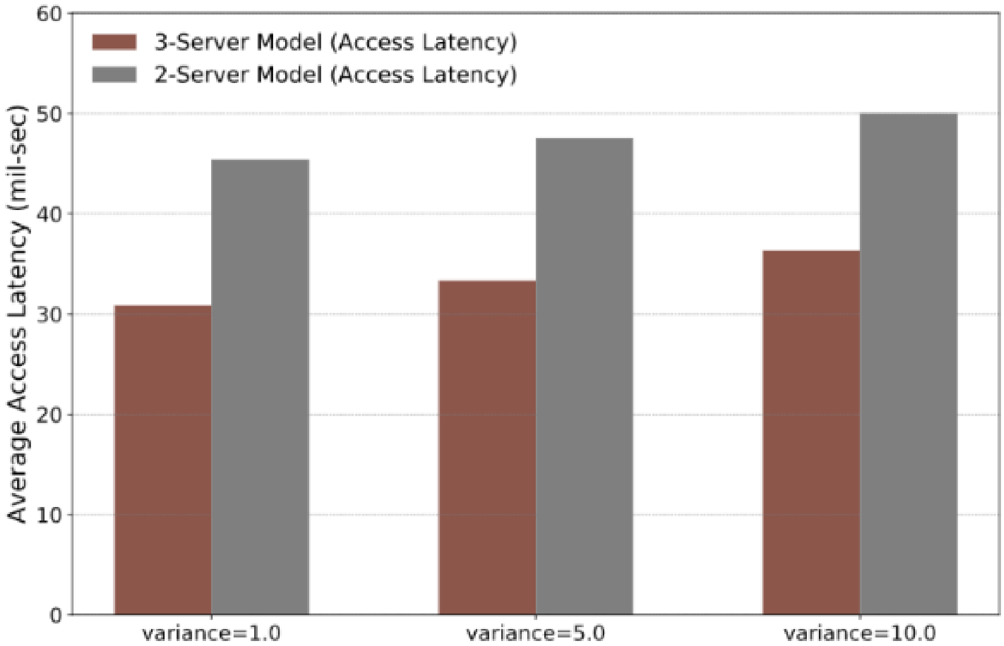


Fig.5.10 Average Access Latency in 3 & 2 Server Model

5.4 Summary

This section indicated our study and a novel cloud resource selection scheme employing ML in 5G VANETs taking into consideration network dynamism. The study applies the knowledge of the latest cellular system, distributed cloud computing, vehicular communication and machine learning. The potentiality of ML in communication system is uncountable. We examined three types of machine learning algorithms and extensively demonstrated the applicability to vehicular systems. Our proposed system separates the time-consuming learning process from the execution process, placing a ML agent at network edges.

Substantial sets of simulations were thoroughly conducted to evaluate prediction performance over the algorithms, missing data handling performance, latency contribution of the proposed scheme, and we observed the performance in the proposed 3-server model. According to the simulation results, XGBC ML algorithm constantly demonstrated better prediction performance in both the short and long observation ranges. The best performance of XGBC shown was 0.963 in 10K sampling volume. The simulation results also indicated that XGBC has stable missing data handling capability and thus this will ensure the system stability. The best performance of XGBC in 10% missing data was 0.92 in 1,000 sampling volume. The simulation also confirmed its advantage in latency contribution on the proposed learning agent system. Remarkably, it took only 13 mil-sec execution time on average irrespective of the sampling data volume on average. This implies a large set of training data can be applied because of the learning time isolation in a highly mobile vehicular scenario. The latency contribution was reduced by up to 68% on the proposed 3-server model with VAC accompanying the model.

It also notable that the implementation of the proposed scheme is remarkably simple. As it can be realized by adding a software program which reflects this scheme onto the application layer without touching any low-layer mechanisms. If we add such application software onto the network edge and vehicles, this scheme will be realized from day-one. Furthermore, the function that has learnt the server selection in an agent at the network edge can be used for any vehicles' request as this is not a specific function. The data collection schemes in a cellular system are in progress. They can be observed such as in advanced zero touch OAM management mechanisms [61], [62]. We have not touched upon any possibility of improving the performance of applied ML algorithms such as by changing depth of parameter adjustments. We believe however this is a room for performance improvements in this area. Extension of GBMC is also a part of up-to-date study. Light GBM introduced by Microsoft is an improvement of GBM algorithms [63]. In addition, there is still a large potential in the application of ensemble learning onto a large dimensional data. There are still plenty of opportunities for further study on these subjects.

Chapter 6

6 Conclusion and Discussion

In this thesis, we pursued improved and innovative proposals on vehicular communications in conjunction with cellular systems specifically leveraging the power of machine learning (ML). Specifically, we are targeting to propose an advanced vehicular clustering scheme and an appropriate computational server selection scheme knowing dynamic network resource status applying several ML approaches. Throughout this research we held on two important policies, one is to stand on a mathematically well-proofed concept and understand it. The second policy is to evaluate the mathematically-designed concept considered over computational simulations. Finally, we were able to find quite interesting mathematical models and eventually able to implement them over several simulation platforms. The result provided us with uncountable findings and a significant potentiality of various fields of applications. This section concisely summarizes these our research findings and results throughout the activities.

6.1 Conclusion

Vehicular communications as well as moving MTC communications through a cellular network system have been motivated by strong industry needs. In view of the wide scope and countless number of applications, it is easy to imagine that there are a large number of expected use cases. Chapter 1 provided an overview of typical scenarios, starting with updating the standardization status and introducing C-V2X as an improved communication platform for advanced vehicular communications. Smart Internet Vehicles (SIV) equipped with various advanced sensors compute a huge volume of data while connecting to the cellular system. Such distributed vehicular computation will come to be realized for the sake of virtualized and distributed computing networks. Various mechanisms of ML support the complex decision making on behalf of human beings.

For the realization of improved vehicular communications in conjunction with cellular systems, specifically this thesis studied two key research subjects: an advanced vehicular clustering scheme and a dynamic server selection scheme for nomadic vehicles. Their details were explained in Chapters 3 to 5.

Chapter 2 reviewed related fundamentals of vehicular clustering schemes and mainly ML theories. For a sophisticated object clustering algorithm, we introduced Affinity Propagation Clustering (APC) based on the factor-graph theory. Gazis-Herman-Rothery (GHR) car following model was introduces as an additional theory for processing real traffic data for simulation analysis. We reviewed the ML variations and algorithms applied in this thesis. Support Vector Machine Classification (SVMC), Gradient Boosting Machine Classification (GBMC) and eXtream Gradient Boosting Machine Classification (XGBC) were fully analyzed to be applied in our proposals.

Chapter 3 introduced our first study, which is applied the Affinity Propagation Classification (APC) theory for VANET dynamic clustering. The proposal extended the original concept of APC and redesigned it to leverage the power of C-V2X wireless communication platform. Specifically, the study introduced two dimensional parameters: the vehicles' current position and current velocity with weighted values α and β over the terms. The idea of this scheme infers further extensibility of input terms and dimensions. The study applied the Cellular V2X radio to improve the frequency of data exchange, which significantly contributed to minimizing the APC clustering interval. The study treated real traffic data with the GHR car following model and successfully clustered live traffic in various conditions. The simulation result identified that the number of minimum iterations increases when the number of target object increases. In addition, controlling the VANET clustering granularity by changing the *px*-value, *i.e.*, the preference value of the AP, was demonstrated. The simulation indicated a successful control of the clustering granularity by changing the *px*-value in our NMDP-APC.

Chapter 4, continuing the work as an extension of the previous work, studied how to control the cluster granularity by adjusting the granularity parameter. The study deduced the desirable clustering size, applying a ML scheme which employs soft-margin-based SVM-ML with Gaussian Radial Basis Kernel (GRBF) Function. We tried to find an ideal VANET granularity incorporating minimum sets of decision criteria. Relative to the limited and essential criteria, the ML prediction performance achieved satisfactory results with fewer training data. The procedure and sequence are enhanced on the existing 3GPP C-V2X specifications. The proposed scheme is also designed through a distributed control approach to adapt nomadic vehicles' mobility. Through the simulations, the cluster formation and granularity control were clearly observed through different values of the *px*-parameter. For improving the ML prediction performance, two key parameters C and γ were adjusted to tune the prediction performance. We also observed the performance with lesser access latency in a stable PLMN system via the simulation, which explains that the clustering capability is contributing to providing aggregation effects and thus it provides a stable system even when a large number of objects are connecting to a PLMN.

Chapter 5 studied and proposed a novel cloud resource selection scheme employing ML in 5G VANETs taking into consideration network dynamism. The proposed system separates the time-consuming learning process from the execution process, placing a ML agent on the network edge. Substantial sets of simulations were thoroughly conducted to evaluate prediction performance over the algorithms, missing data handling performance, latency contribution of the proposed scheme, and we observed the performance in the proposed 3-server model. According to the simulation results, XGBC ML algorithm constantly demonstrated better prediction performance in both the short and long observation ranges. The simulation results also indicated that XGBC has a stable missing data handling capability and thus this will ensure the performance stability. The simulation also confirmed its advantage in latency contribution on the proposed learning agent system. This implies the applicability of a large set of training data taking advantage of learning time isolation in a highly mobile vehicular scenario. The latency contribution was reduced by up to 68% on the proposed 3-server model with VAC accompanying the model. It is also notable that the implementation of the proposed scheme is remarkably simple. As it can be realized by adding a software program by which to reflect this scheme onto the application layer without touching any low-layer mechanisms. Furthermore, the function that has learnt the server selection in an agent on the network edge can be used for any vehicles' request as this is not a specific function. Those studies on data collection schemes in a cellular system are in progress [61], [62]. Eventually in Chapter 6, we concluded the overall activity of this thesis.

6.2 Discussion

Studies of vehicular communications operating in conjunction with a cellular network are essential and highly demanded subjects reflecting the industry needs. It has been an honor that we are able to engage in such valuable and challenging work though this thesis. Specifically, employment of ML onto the study has been quite challenging, but at the same time very intriguing throughout the activities. Applications of ML to the communication field will continue to increase in number and variety. We have identified a quite large number of papers and contributions through this study. This field is surprisingly expanding and increasing with related researches underway in various application fields in the cellular communications. One of such initiatives can be seen in RISING (Cross-Field Research Association of Super-Intelligent Networking) project in IEICE. Activity of ITU is also making progress in a cross-country competitive project called 5G AI Challenge, which is targeted at applying ML to 5GS in quite widely spreading fields. While it has been challenging to compete with global experts of ML and AI (Artificial Interagency), this field is gaining momentum, opening up wider opportunities for many more players to come.

Although the studies of this thesis were originally targeted at vehicular applications, it has a potential applicability to any machine type moving objects expected to increase reflecting the growing demand for their applications in 5G and the beyond. Our future work can include studies focusing on enhancing the proposed scheme by applying more complicated scenarios and/or using different types of ML algorithms. In addition, studies on data collection schemes in a cellular system are also in progress, such as for advanced zero touch OAM management mechanisms. As research on the application of intelligence to cellular systems is also ongoing, we should keep a close eye on these activities as well.

Bibliography

[01] 3GPP TS 22.261, “Service requirements for the 5G system; Stage 1”, v15.7.0, Rel. 15, December 2018

[02] 3GPP TS 36.300, “Evolved Universal Terrestrial Radio Access (EUTRA) and Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Overall description; Stage 2”, v15.2.0, June 2018

[03] 3GPP TS 23.501, “System architecture for the 5G system; Stage 2”, Release 15, v15.5.0, March 2019

[04] 3GPP TS 23.502, “Procedures for the 5G System; Stage 2”, Release 15, v15.5.1, April 2019

[05] GSMA 5G Global Launches & Statistics, https://www.gsma.com/futurenetworks/ip_services/understanding-5g/5g-innovation/

[06] White Paper, “C-V2X_Use Cases, Methodology, Examples and Service Level Requirements”, 5GAA, v1.3.1, July 2019, https://5gaa.orgwp-contentuploads2019075GAA_191906_WP_CV2X_UCs_v1-3-1.pdf

[07] 5GAA live Demonstrations “C-V2X as a market reality”, Turin Italy, November 2019, https://5gaa.org5gaa-in-motionnews

[08] Quasi-Zenith Satellite System, Cabinet Office, Government of Japan, https://qzss.go.jp/en/overview/services/sv01_what.html

[09] 3GPP TS22.186, “Service requirements for enhanced V2X scenarios; Stage1”, Rel.16, v16.1.0, December 2018.

[10] 3GPP TS 23.287, “Architecture enhancements for 5G System (5GS) to support Vehicle-to-Everything (V2X) services”, v16.4.0, Release 16, September 2020.

[11] 3GPP TS 23.285, “Architecture enhancements for V2X services; Stage 2”, Release 16, March 2019.

[12] Laser Imaging Detection and Ranging, https://en.wikipedia.org/wiki/Lidar

[13] Distributed Computing, Wikipedia, https://en.wikipedia.org/wiki/Distributed_computing

[14] A.Malathi and Dr. N. Sreenath, “A Comparative Study of Clustering Protocols in VANET”, International Journal of Emerging Trends & Technology in Computer Science (IJETTCS), Vol 6, Issue 1, Feb 2017.

[15] C. Cooper, D. Franklin, M. Ros, F. Safael, and M. Abolhasan, “A Coorporate Survey of NANET Clustering Techniques”, IEEE Communications Survey and

Tutrial, vol 10, No 1, 1Q 2017.
[16] W.Qi, Q.Song, X.Wand, L.Guo, and Z.Ning, “SDN-Enabled Social-Aware Clustering in 5G-VANET Systems”, IEEE Access, Special Edition on Advanced Big Data Analysis for Vehicular Social Network, pp.28213-28224, vol.6, 2018.
[17] C. Shea, B, Hassabadi, and S. Valaee, “Mobility-Based Clustering in VANETs using Affinity Propagation,” 2009 IEEE GLOBECOM, December 2009.
[18] B. Hassnabadi, C. Cha, L. Zhang, S. Valaee, “Clustering in Vehicular Ad Hoc Networks using Affinity Propagation.”, Ad Hoc Networks 13 (2014), pp 535-458, ELESEVIER.
[19] H. Shahwani, T. D. Bui, J. Jeong, and J. Shin, “A Stable Clustering Algorithm based on Affinity Propagation for VANETs,” 19th International Conference on Advanced Communication Technology (ICACT), pp.501-504, February 2017.
[20] B. J. Frey and D. Dueck, “Clustering by Passing Messages Between Data Points,” Science Magazine published by AAAS, vol. 315, February 2007, pp. 972-976.
[21] AFFINITY PROPAGATION Frequently Asked Questions, http://www.psi.toronto.edu/affinitypropagation/faq.html
[22] K. Bevrani, E. Chung, M. Miska, “Evaluation of the GHR Car Following Model for Traffic Safety Studies”, 25th ARRB Conference, Perth, Australia 2012.
[23] Dr. T. V. Mathew, “Car Following Models,” Transportation System Engineering, Chapter 35, pp. 35.2 ~ 35.6, IIT Bombay, March 2017.
[24] H. Ye, L. Liang, G. Y. Li, J.B. Kim, L. Lu, and M. Wu, “Machine Learning for Vehicular Networks: Recent Advances and Application Examples”, Vol.13, Issue 2, IEEE vehicular technology magazine, June 2018.
[25] A.Ben-Hur and J.Weston, “A User’s Guide to Support Vector Machines”, January 2010.https://www.researchgate.net/publication/41896604_A_User's_Guide_to_Support_Vector_Machines
[26] C.Junli and J.Licheng, “Classification Mechanism of Support Vector Machines”, 5th International Conference on Signal Processing Proceedings, ICSP 2000.
[27] C. Cortes and V. Vapnik. Support vector networks, Machine Learning 20, pp 273-297, Kluwer Academic Publishers, Boston. 1995.
[28] Zaburo, “GRADIENT BOOSTING and XGBOOST”, May 2017, https://zaburo-ch.github.io/post/xgboost/
[29] A. Rogozhnikov, “Gradient Boosting Interactive Playground”, July 2016, http://arogozhnikov.github.io/2016/07/05/gradient_boosting_playground.html
[30] G. Seif, “A Beginner’s guide to XGBoost”, March 2019, https://towardsdatascience.com/a-beginners-guide-to-xgboost-87f5d4c30ed7

[31] M. Pathak, “Using XGBoost in Python”, November 2019, https://www.datacamp.com/community/tutorials/xgboost-in-python

[32] A. Benslimane, T. Taleb, and R. Sivaraj, “Dynamic Clustering-Based Adaptive Mobile Gateway Management in Integrated VANET - 3G Heterogeneous Wireless Networks,” IEEE Journal on Selected Areas in Communication, vol. 29, no. 3, March 2011, pp. 559-570.

[33] S. Ucar, S. C. Ergen, and O. Ozkasap, “Multihop-Cluster-Based IEEE 802.11p and LTE Hybrid Architecture for VANET Safety Message Dissemination,” IEEE Transactions on Vehicular Technology, vol. 65, no. 4, April 2016, pp. 2621-2636.

[34] S. Ucar, S. C. Ergen, and O. Ozkasap, “VMaSc: Vehicular Multi-hop algorithm for Stable Clustering in Vehicular Ad Hoc Networks,” IEEE Wireless Communications and Networking Conference (WCNC), April 2013, pp. 2381-2386.

[35] B. Hassnabadi, C. Cha, L. Zhang, S. Valaee, “Clustering in Vehicular Ad Hoc Networks using Affinity Propagation.”, Ad Hoc Networks 13 (2014), pp 535-458, ELESEVIER.

[36] A.Malathi and Dr. N. Sreenath, “A Comparative Study of Clustering Protocols in VANET”, International Journal of Emerging Trends & Technology in Computer Science (IJETTCS), Vol 6, Issue 1, Feb 2017.

[37] C. Cooper, D. Franklin, M. Ros, F. Safael, and M. Abolhasan, “A Comparative Survey of VANET Clustering Techniques”, IEEE Communications Survey and Tutorial, vol 10, No 1, 1Q 2017.

[38] C-V2X Trial in Japan, https://www.qualcomm.com/media/documents/files/c-v2x-trial-in-japan.pdf, Qualcomm, 2018.

[39] S. A. A. Shah, E. Ahmed, M. Imran and S. Zeadally, “5G for Vehicular Communications”, IEEE Communications Magazine, Vol.56, Issue 1, Jan. 2018.

[40] H. El-Sayed, S. Sankar, M. Prasad, D. Puthal, A. Gupta, M. Mohanty, and C.T. Lin, “Edge of Things: The Big Picture on the Integration of Edge, IoT and the Cloud in a Distributed Computing Environment”, IEEE Access, Vol.6, 2018.

[41] Z. Ning, X. Wang, and J. Huang, “Mobile Edge Computing-Enabled 5G Vehicular Networks, Toward the Integration of Communication and Computing”, IEEE vehicular technology magazine, March 2019.

[42] L. Li ; Y. Li and R. Hou, “A Novel Mobile Edge Computing-Based Architecture for Future Cellular Vehicular Networks”, IEEE Wireless Communications and Networking Conference (WCNC), May 2017.

[43] X.Ma, J.Zhao, Y. Gong and Y.Wang, “Key Technologies of MEC Towards 5G-

Enabled Vehicular Networks", International Conference on Heterogeneous for Quality, Reliability, Security and Robustness in Heterogeneous System, Springer 2017.

[44] A.A.Khan, M.Abolhasan, and W.Ni, "An Evolutionary Game Theoretic Approach for Stable and Optimized Clustering in VANETs", IEEE Transactions on Vehicular Technology, vol. 67, no.5, May 2018, pp.4501-4513.

[45] L. Liang, H. Ye, G. Y. Li "Toward Intelligent Vehicular Networks: A Machine Learning Framework", IEEE Internet of Things Journal, Vol.6, Issue.1, February 2019.

[46] D. Tian, Y. Wang, H. Xia, and F. Cai, "Clustering multi-hop information dissemination method in vehicular ad hoc networks," IET International Transportation Syst., vol. 7, no. 4, pp. 464-472, Jan. 2013.

[47] Z. Khan, P. Fan, F. Abbas, H. Chen, and S. Fang, "Two-level cluster based routing scheme for 5G V2X communication", IEEE Access, vol. 7, pp. 16194-16205, 2019.

[48] 3GPP TS 23.303, "Proximity-based services (ProSe)", Stage 2, Release 15, June 2018.

[49] G. Araniti, C. Campolo, M. Condoluci, A. Iera, A. Molinaro, "LTE for Vehicular Networking: A Survey", IEEE Communications Magazine, vol.51, no. 5, May 2013.

[50] A. Khan, A. Minokuchi, K.Tsubouchi, G.Kunito, S.Iwashina, "Technology and Standards Accelerating 5G Commercialization", IEICE Transactions on Communications, Vol. E102.B, No. 3.

[51] Truncated normal distribution, Wikipedia, https://en.wikipedia.org/wiki/Truncated_normal_distribution.

[52] G. Qiao, S. Leng, K Zhang, and Y. He, "Collaborative Task Offloading in Vehicular Edge Multi-Access Networks", IEEE Communications Magazine, August 2018.

[53] S. Memon, M. Maheswaran, "Using machine learning for handover optimization in vehicular fog computing", Computer Science, Mathematics, Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing, 2019.

[54] T. Koshimizu and S. Gengtian et al, "Multi-Dimensional Affinity Propagation Clustering Applying a Machine Learning in 5G-Cellular V2X", IEEE Access, Volume 8, 2020.

[55] D. Ye, R. Yu, M. Pan, and Z. Han, "Federated Learning in Vehicular Edge Computing: A Selective Model Aggregation Approach", IEEE Access, 2020.

[56] J. Cao, K. Zhang, F. Wu and S. Leng, “Learning Cooperation Schemes for Mobile Edge Computing Empowered Internet of Vehicles”, IEEE Wireless Communications and Networking Conference (WCNC), 2020.
[57] J. Shi, J. Du, J. Wang and J. Yuan, “Distributed V2V Computation Offloading Based on Dynamic Pricing Using Deep Reinforcement Learning”, IEEE Wireless Communications and Networking Conference (WCNC), 2020.
[58] T. Chen and C. Guestrin, “XGBoost: A Scalable Tree Boosting System”, KDD ’16, August 2016, San Francisco, CA, USA.
[59] 3GPP TS 23.287, “Architecture enhancements for 5G System (5GS) to support Vehicle-to-Everything (V2X) services”, v16.4.0, Release 16, September 2020.
[60] S. Wu, W. Xia, W. Cui, Q. Chao, Z. Lan, F. Yan and L. Shen, “An Efficient Offloading Algorithm Based on Support Vector Machine for Mobile Edge Computing in Vehicular Networks”, International Conference on Wireless Communications and Signal Processing (WCSP), December 2018.
[61] Zero touch network & Service Management (ZSM), https://www.etsi.org/technologies/zero-touch-network-service-management?jjj=1601440615846
[62] C. Benzaid and T. Taleb, “AI-Driven Zero Touch Network and Service Management in 5G and Beyond: Challenges and Research Directions”, IEEE Network, Volume 34, Issue 2, 2020.
[63] G. Ke, Q. Meng. T. Finely, T. Wang et al, “LightGBM: A Highly Efficient Gradient Boosting Decision Tree”, Advances in Neural Information Processing Systems 30 (NIP 2017), December 2017.

早稲田大学　博士（工学）　学位申請　研究業績書

(List of research achievements for application of doctorate (Dr. of Engineering), Waseda University)

氏 名　(輿水 敬)　　　　　印(　　)

（As of February, 2021）

種 類 別(By Type)	題名、　発表・発行掲載誌名、　発表・発行年月、　連名者（申請者含む）(theme, journal name, date & year of publication, name of authors inc. yourself)
Journals	#1〇 **Takashi Koshimizu**, Shi Gengtian, Huan Wang, Zhenni Pan, Jiang Liu, and Shigeru Shimamoto, “Multi-Dimensional Affinity Propagation Clustering Applying a Machine Learning in 5G-Cellular V2X”, *IEEE Access* July 2020, Vol 8.
Conferences	#2 Shi Gengtian, **Takashi Koshimizu**, Megumi Saito, Pan Zhenni, Liu Jiang and Shigeru Shimamoto, “Power Control Based on Multi-Agent Deep Q Network for D2D Communication”, *IEEE International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, Fukuoka, Japan, February 2020. #3 Huan Wang, **Takashi Koshimizu**, Zhenni Pan, Jiang Liu and Shigeru Shimamoto, “Signal Priority Management for Bus with V2I Network”, *The 37th Annual Conference on JSST (Japan Society for Simulation and Technology)*, Hokkaido, Japan, September 2018. #4〇 **Takashi Koshimizu**, Huan Wang, Zhenni Pan, Jiang Liu and Shigeru Shimamoto, “Normalized multi-dimensional parameter based affinity propagation clustering for cellular V2X”, *IEEE Wireless Communications and Networking Conference (WCNC)*, Barcelona, Spain, April 2018. #5 Huan Wang, Jiang Liu, Zhenni Pan, **Takashi Koshimizu** and Shigeru Shimamoto, “Cooperative Traffic Light Controlling Based on Machine Learning and a Genetic Algorithm”, *IEEE 23rd Asia-Pacific Conference on Communications (APCC)*, Perth, WA, Australia, December 2017.
Standardization	#6 NTT DOCOMO & Intel, S2-173339 "23.502: QoS mapping for 5GC-EPC interworking", 3GPP TSG SA WG2 Meeting #121, 15-19 May 2017, Hangzhou, P.R. China. #7 NTT DOCOMO, S2-162722, "Support for Dynamic control of Idle-mode mobility", SA WG2 Meeting #115, 23 - 27 May 2016, Nanjing, P.R. China #8 NTT DOCOMO, S2-152845, "Removing per UE GTP-C on S11 during idle-mode", SA WG2 Meeting #110AH, 01-03 September 2015, Sofia Antipolis, France and more than 70 contributions in entire 3GPP standardization activities.