Republic of Tunisia
Ministry of Higher Education,
Scientific Research

University of Sfax
National Engineering School of Sfax

*Doctoral school*
*Science and Technology*

**Doctoral thesis**

**Computer Systems Engineering**
**Order N. 21/2023**

# THESIS

***Presented in***

## National Engineering School of Sfax

***In order to obtain the***

## DOCTORATE

**In**
***Computer Systems Engineering***

***By***
**Mays Kareem Jabbar Alsabah**

**(Computer and Wireless Communication Engineering)**

# HyperGraph Based Stable Clustering in VANET

***Defended on June 10, 2023 in front of the following Committee:***

| | | | |
|---|---|---|---|
| **Mrs.** | **Ines Kammoun Jemal** | **(Professor)** | President |
| **Mr.** | **Faouzi Derbel** | **(Professor)** | Reporter |
| **Mr.** | **Tarek Abbes** | **(Associate Professor)** | Reporter |
| **Mr.** | **Mohamed Adel Alimi** | **(Professor)** | Examiner |
| **Mr.** | **Hafedh Trabelsi** | **(Professor)** | Supervisor |

# *Dedications*

*To the one who taught me that the world is a struggle and its weapon is science and knowledge... To the one who did not spare me anything... To the one who strived for my comfort and success...*

*To the greatest and dearest man in the universe,*

***My dear father (Kareem).***

*To the one who helped me in her prayers and supplications.... To the one who shared my joys and sorrows with me....*

*To the source of kindness and tenderness...*

*To the most wonderful woman in existence,*

***My dear mother (Khawlah).***

*To my gift from God… the most valuable thing I have... the sweetest feeling... and the most beautiful joy,*

***My son (Hasan) and my daughter (Asal)…***

*who shared all the difficulties with me and endured the separation of their friends for me... and supported me with their love..*

*Without you, I would not have reached where I am now.*

*Mays Alsabah*

# *Acknowledgment*

I would like to start by thanking Allah Almighty for giving me strength and patience, and every time I get weak I find God my only refuge and the source of my strength… Thanks my God, for hearing all my prayers.

I express my gratitude and thanks to my main supporter and companion on difficult days, my husband (Thaar). A big thanks to you because without you, I would not have been able to overcome all of my difficulties.

I am very appreciative of the honor Mrs. Ines Kammoun Jemal, Professor at ENI-Sfax, University of Sfax, has bestowed upon me by agreeing to preside over the jury for my thesis defense. I would like to thank her for her interest in this work.

My thanks are also addressed to Mr. Faouzi Derbel, Professor and chair of smart diagnostic and online monitoring at Leipzig University of Applied Sciences, Germany, and Mr. Tarek Abbes, Associate Professor at ENET'Com, University of Sfax, for giving me the honor to report this thesis.

My special thanks are addressed to Mr. Mohamed Adel Alimi, Professor at ENI-Sfax, University of Sfax, for agreeing to examine this work.

I would like to express my gratitude to my supportive supervisor, Mr. Hafedh Trabelsi, Professor at ENI-Sfax, University of Sfax, for his enthusiasm for the work and for his support, encouragement, and patience. He continuously encouraged me and was always willing to assist in any way throughout my Ph.D. study.

بسم الله الرحمن الرحيم

" فَرِحِينَ بِمَا آتَاهُمُ اللَّهُ مِن فَضْلِهِ وَيَسْتَبْشِرُونَ "

صدق الله العظيم

# *List of publications*

## Articles Published in international journals

1. M. K. Jabbar and H. Trabelsi, “A novelty of Hypergraph Clustering Model (HGCM) for Urban Scenario in VANET,” *IEEE Access*, 2022.

2. M. K. Jabbar and H. Trabelsi, “Clustering Review in Vehicular Ad hoc Networks: Algorithms, Comparisons, Challenges and Solutions.,” *Int. J. Interact. Mob. Technol.*, vol. 16, no. 10, 2022.

3. M. K. Jabbar and H. Trabelsi, “EtHgSC: Eigen Trick-Based Hypergraph Stable Clustering Algorithm in VANET,” *J. Electr. Comput. Eng.*, vol. 2023, p. 6327247, 2023.

## Communications in international conferences

1. M. K. J. Alsabah, H. Trabelsi, and W. Jerbi, “Survey on Clustering in VANET Networks,” in *2021 18th International Multi-Conference on Systems, Signals & Devices (SSD)*, 2021, pp. 493–502.

2. M. K. Jabbar and H. Trabelsi. “A Review on Clustering in VANET: Algorithms, Phases, and Comparisons.” in *2022 19th International Multi-Conference on Systems, Signals & Devices (SSD)*, 2022, pp. 444–451.

3. M. K. Jabbar and H. Trabelsi, “A Betweenness Centrality Based Clustering in VANETs,” in *2022 15th International Conference on Security of Information and Networks (SIN)*, 2022, pp. 1–4.

# *Table of Contents*

# *Abstract*

A vehicular ad hoc network (VANET) is a dynamic and constantly changing topology that requires stable clustering to prevent connection failure. This thesis brings original contributions, mainly to VANET clustering algorithms. This is our experiment with the hypergraph theory at different clustering schemes and cluster head (CH) selection parameters. In order to guarantee cluster stability, two clustering approaches based on the hypergraph theory are designed as a complete solution for VANET challenges. Each approach is bifurcated into two parts; cluster generation and CH selection. The first approach introduces a formulation of VANET through hypergraph spectral clustering. Hypergraph partitioning through the tensor trace maximisation (TTM) method is presented. Then, the Eigen-trick method is used to calculate the modified Laplacian value in TTM to improve the clustering in the second designed approach. The Eigen-trick considers the transformable connection between the vertex Laplacian and the hyperedge Laplacian, which can speed up the solution of eigenproblems without losing information. Also, it provides an approach for reducing the computational complexity of the clustering.

In the two approaches, the CH is selected using different parameters, considering the criteria for maintaining a stable connection with the maximum number of neighbours. Also, two CH selection schemes are presented to select the most stable vehicle as a head. Strong connectivity and a stable link lifetime are obtained using these schemes.

The proposed approaches demonstrate a considerable improvement in terms of stability and network performance compared with other techniques in the literature.

The designed approaches are tested on a real map of Baghdad city with the help of an open street map (OSM) and a simulation of urban mobility (SUMO) to generate realistic traffic. SUMO supports the traffic control interface (TraCI) for MATLAB interfacing. The network is triggered from MATLAB through the TraCI API an data is recorded at every simulation second.

**Keywords:** VANET, Clustering algorithm, Hypergraph, Tensor trace maximisation, Laplacian value, Cluster head, Stability, Relative speed, Neighbouring degree, Trust, Time to leave estimation, Long short term memory, Eigen-trick, Grey relational analysis.

# *List of Figures*

## *Chapter 1*



## *Chapter 2*



## *Chapter 3*

## *Chapter 4*

# *List of Tables*

## *Chapter 1*



## *Chapter 2*



## *Chapter 3*



## *Chapter 4*

# *List of Notations*

| Notations | Explanation |
|---|---|
| ITS | Intelligent Transportation System |
| MANET | Mobile Ad Hoc NETwork |
| VANET | Vehicular Ad Hoc NETwork |
| WAVE | Wireless Access in Vehicular Environment |
| OBU | On-Board Unit |
| RSU | Roadside Unit |
| V2V | Vehicle to vehicle |
| V2I | Vehicle to infrastructure |
| V2R | Vehicle to roadside unit |
| FCC | Federal Communication Commission |
| ETC | Electronic Toll Collection |
| DSRC | Detected Short Range Communication |
| BSS | Basic Service Set |
| CCH | Control Channel |
| SCH | Service Channel |
| CH | Cluster Head |
| CM | Cluster Member |
| GW | Gateway node |
| IoV | Internet of Vehicles |
| LTE | Long-Term Evolution |
| PDR | Packet Delivery Ratio |
| E2E Delay | End to End Delay |
| SUMO | Simulation of Urban Mobility |

| | |
|---|---|
| OSM | Open Street Map |
| TraCI | Traffic Control Interface |
| API | Application Programming Interface |
| TTM | Tensor Trace Maximization |
| $\Theta$ | Vehicle direction |
| $V_{vehi}$ | Vehicle speed |
| $\psi_{vehi}$ | Relative speed |
| $\eta$ | Neighbourhood Degree |
| $\mathfrak{E}$ | Eccentricity |
| $\mathfrak{t}$ | Trust |
| PU | Primary user |
| SU | Secondary user |
| $Lv$ | Vertex Laplacian |
| $Le$ | Hyperedge Laplacian |
| PGRP | Predictive directional greedy routing protocol |
| $\theta'_{vehi}$ | Predicative direction |
| $(X'_{vehi}, Y'_{vehi})$ | Vehicle predicted coordinates |
| GRG | Grey relational grade |
| $\gamma(Y_{0k}, Y_{ik})$ | Grey relational coefficient |
| g(v) | Vehicle grade |
| $Dig$ | Diagonal matrix |
| GRA | Grey Relational Analysis |
| $L$ | Fiedler vector; $L_e$hyperage Laplacian and $L_v$ vertex Laplacian |
| HGCM | HyperGraph Clustering Model |
| EtHgSC | Eigen trick-based Hypergraph Stable Clustering |

| | |
|---|---|
| $f_i(t)$ | Vehicle feature |
| $m_i(t)$ | Metric tuple with relative speed, neighbours, eccentricity and trust |
| $c_{ij}$ | Node connectivity |
| $U$ | Set of eigenvectors; $U_v$ for vertices and $U_e$ for hyperadges |
| $t$ | Total simulation time |
| $N$ | Maximum vehicles' number |
| $Vehi_{ID}$ | The ID of vehicle |
| $R_{vehi}$ | Transmission range of vehicles |
| $C_{num}$ | No. of clusters formed in the network |
| $tr(B_k)$ | Between-group dispersion matrix |
| $tr(Z_k)$ | Within-cluster dispersion matrix |
| $N(\mu, \sigma^2)$ | Gaussian probability distribution function |
| $C_B$ | Betweenness centrality |
| $\sigma_{st}$ | Total number of the shortest path |
| $RSU_{Loc}$ | Location of auxiliary facilities placed |
| $C_{optimal}$ | Optimal number of clusters |
| $Vehi_{num}$ | Number of vehicles at each cluster |
| $s$ | Calinski–Harabasz index |
| $tr(Z_k)$ | Within-cluster dispersion |
| $tr(B_k)$ | Between-cluster dispersion |
| no_lane | Lanes' number on the map |
| $R_{RSU}$ | RSU transmission range |
| (δ) | Rewarded or penalised with an absolute value of 0.01 |
| $S_{thr}$ | The threshold of speed |

| | |
|---|---|
| $V_{avg}$ | Average speed of vehicles in a cluster |
| $CH_{score}$ | CH score |
| $I$ | Incidence matrix |
| $L_{th}$ | Lane length |
| $d_{vehi}$ | Distance covered by a vehicle on the road |
| $T_{leave}$ | Time to leave |
| $T'_{leave}$ | Estimated time to leave |
| $\lambda$ | Eigenvalues |
| $h$ | Channel |
| $\gamma$ | Signal-to-noise ratio |
| $P_d$ | Probability of detection |
| $P_f$ | Probability of false alarm |
| $T(Y)$ | Test statistics |
| $\mathbb{Q}(.)$ | Complementary distribution function |
| $\chi^2$ | Chi-square probability distribution function |

# Chapter 1

# General Introduction

## 1.1 Overview of Vehicular Ad-hoc Network

The Intelligent Transportation System (ITS) that includes all types of communications between vehicles, is an important next-generation transportation system. ITS provides many facilities to the passengers, such as safety applications, assistant to the drivers, emergency warnings, etc. With the rapid development of automotive manufacturing, vehicles are becoming more and more intelligent and powerful. Vehicular Ad Hoc NETwork (VANET) is a self-organizing network formed by a collection of moving vehicles. VANET is a subset of Mobile Ad Hoc NETwork (MANET) [1]. MANET is a network without a fixed and self-configuring infrastructure of mobile nodes connected via wireless communication. When the mobile nodes in MANETs are replaced by vehicles and start to navigate fixed routes, such as roads, the network becomes a VANET. In contrast to MANET, VANET features a hybrid network design, sufficient energy, computational power, and nodes with increased mobility over limited routes [2].

Beginning in the early 1990s, people began to pay more and more attention to VANET technologies, and in recent years, it has become a significant concern. VANETs have shown promise in improving driving efficiency and traffic safety. Drivers can prevent unforeseen accidents at their blind spots, such as corners or other challenging road situations, by sharing pertinent traffic information with a group of close vehicles [3]. In emergency situations, vehicles like police cars and ambulances may receive a greater priority, allowing motorists nearby to be promptly alerted. Real-time traffic data can assist drivers in planning their routes and travel times to reduce congestion and increase traffic efficiency. A better-organized routing strategy helps reduce energy consumption and save resources. The primary benefit of VANET is to help vehicles communicate with one another and share information, which helps reduce traffic and improve road safety [4].

VANET uses vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communications to increase driving efficiency and safety. Effective communication is made possible for usage in automobiles by dedicated short-range communications technology, which specifically refers to a suite of standards for Wireless Access in Vehicular Environments (WAVE) and supports both V2V and V2I communications [5].

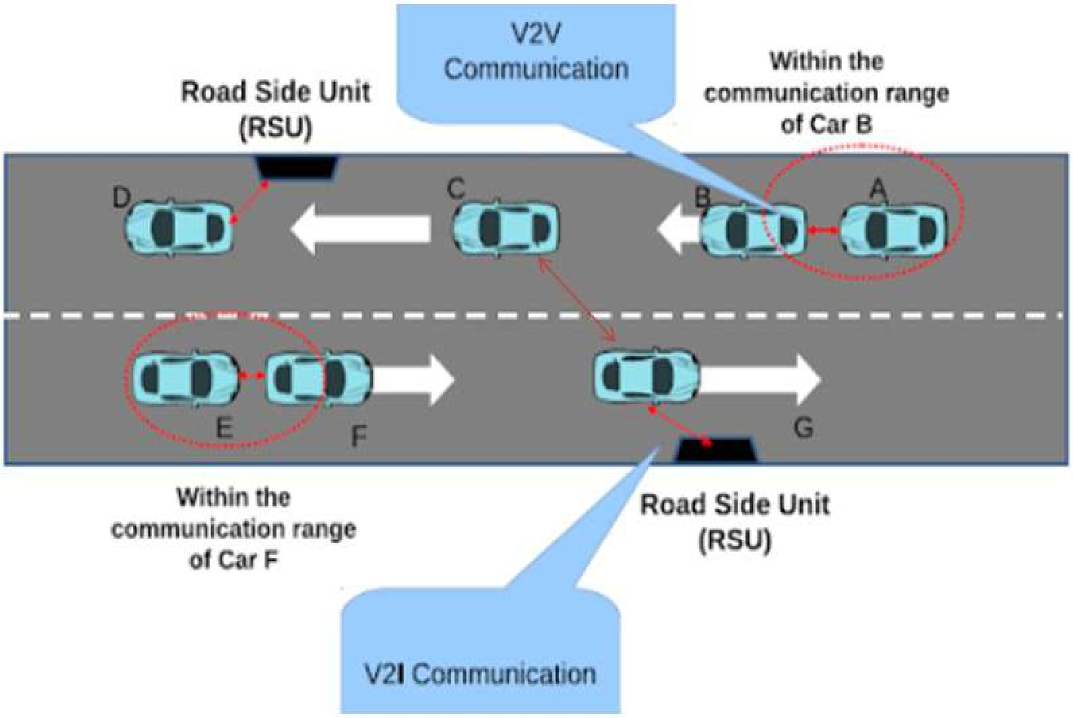



Figure 1.1.Vehicular Ad Hoc Network (VANET) [5].

VANET has two critical elements; Roadside Units (RSUs) and On-Board Units (OBUs). The RSUs are placed alongside the road, where they save all vehicle information and forward it to other OBUs. RSU is mounted along a road or pedestrian passageway. It may also be mounted on a vehicle or hand-carried, but it may only operate when the vehicle or hand-carried unit is stationary. The VANET network is shown in Figure 1.1 [5].

The RSUs can control the entire information transmission task in the vehicles or OBUs. On the other hand, OBUs are the devices placed in the dynamic vehicles and support information exchange among the vehicles and RSUs [1].

In the rest of this chapter, the characteristics and challenges of VANET, as well as VANET's communication modes and the associated technology, are introduced. The most important VANET applications are also presented. Furthermore, we identify the challenges in VANET and address the motivations and objectives of this thesis. At last, we present the contributions and organisation of this thesis.

### 1.1.1 VANET Characteristics

Compared with MANET, the specific characteristics and challenges of VANET have attracted researchers and industries from different fields to investigate VANET applications, technologies, protocols, and standards. VANET inherits similar features from MANET; however, it shows some unique characteristics, leading to some new challenges [6], [7].

- Predictable vehicle mobility: Road topologies, traffic signals, traffic conditions, and road signs are all factors that limit the movement of vehicles. As a result, some ITS applications can benefit from a predictable vehicle trajectory.
- Various vehicle mobility patterns: Varied mobility patterns are a feature of VANET. Cars, trucks, and motorcycles typically travel at different speeds. In the meantime, the speed limits vary depending on the traffic conditions.
- Highly dynamic network topology: The network architecture frequently changes as a result of the rapid vehicular traffic, particularly on highways. When one vehicle passes another quickly due to a higher speed, the inter-vehicle connections may become unstable. As a result, these vehicles' information exchange is unreliable, which could further compromise traffic safety. Enhancing information transmission reliability in highly dynamic network topologies is a major area of research, particularly when sending emergency messages.
- Unlimited network scale: The network scale of VANET may be extremely large, such as in scenarios involving extremely congested urban areas. The VANET's restricted transmission range can only provide a short-range vehicle communications without a central controller, which is insufficient to enable some VANET services. Instead of this, the hierarchical network architecture can address the issue of network scalability.

### 1.1.2 Wireless Communication Modes in VANET.

Increasing the efficiency of transportation and enhancing driver safety will be made possible by enabling information sharing between vehicles and infrastructures. The vehicle's sensors and other equipment, including the radar, Global Position System (GPS), and telemetry sensors, can be used to gather data. The RSUs may keep information from neighbouring vehicles or the database. These data can be transmitted to fulfill the needs of any additional nodes. The best possible communication between vehicles, vehicles, and RSUs is ensured using VANET communication. There are three types of communication modes in the VANET [6], [8].

### *A. V2V Communications*

The V2V communications model in VANET refers to vehicle-to-vehicle communications, where OBU-equipped vehicles are capable of direct radio communication within their radio ranges. Applications require data transmission between vehicles within the communication range. Due to the flexibility with which V2V communications can be implemented without the aid of infrastructures, the often exchanged data assists drivers in operating their vehicles safely and efficiently. Data transmission between vehicles is reliable and has a low latency with this type of communication [6].

But the majority of V2V applications rely on communicating with nearby vehicles via vehicular networks to exchange data. However, V2V communications can be exceedingly unreliable if vehicles are unable to send and receive messages (for instance, if OBUs are not equipped or malfunction). Additionally, users cannot access resources on external networks, such as the Internet, through V2V communications [8].

### *B. V2I Communications*

The term "V2I" stands for "vehicle-to-infrastructure communication"; it entails the installation of infrastructure along roads as well as a variety of applications from infrastructure to cars that might improve service quality. V2I communications offer trustworthy, secure, and mobile service applications via the network for cars. Vehicles communicate with RSUs using V2I in order to send and receive information or requests. The V2I has also named a Vehicle to RSU (V2R) if communication occurs between vehicles and an RSU. The RSUs placed along the roads can assist with traffic services for vehicles, such as providing details about the state of the roads or other locations, or providing information about the token collection and parking lot availability, etc [8].

However, achieving continuous connectivity for cars will require more power and end-to-end latency in V2I than in V2V, and the installation of RSUs and other infrastructures is also an expensive strategy [6].

### *C. V2X Communications*

Vehicles and other terminals, such as RSUs, mobile devices, traffic signal towers, etc., can communicate with one another and with each other using V2X communications.

V2V and V2I collaborate in V2X communications to ensure the quality of service of vehicular networks. For instance, pure V2V communications cannot maintain a stable connection for vehicles in low traffic density, thus the cooperation of V2I communications helps to avoid connection loss and enhance the link quality. Traditional V2V models' service areas can be expanded thanks to V2X communications, which also increase V2I's effectiveness and lower costs [8].

Vehicles cannot be served via pure V2I communication when they are outside of an RSU's coverage area. But, with some additional assistance from V2V communications, the system is able to provide for vehicles outside of these specific ranges. Additionally, the transmission of data via V2V communications can assist in lowering the cost of rebroadcasting from RSUs to reach those cars that are too far away to obtain high-quality services. Consequently, the robustness and efficiency of data services can be further improved by the cooperative communication between V2V and V2I [6].

The V2X process in this thesis consists of V2V communications, which are critical in exchanging data between the cluster heads and other cluster members. V2I connects the cluster head to the RSU. The RSU acts like a gateway inside a router, collecting connections from one or several vehicle clusters.

## 1.1.3 Detected Short Range Communications (DSRC)

Dedicated Short-Range Communications for Wireless Access in Vehicular Environments (DSRC/WAVE) have been created to facilitate vehicular communications throughout the development of ITS. Different spectrums and standards have been assigned by Europe, the United States, and Japan for vehicular communications [9].

The Federal Communications Commission (FCC) of the United States assigned 75 MHz (between 5.850 GHz and 5.925 GHz) of spectrum for DSRC in vehicular environments in 1999. A few years later, the 5.9GHz DSRC was created by the European Telecommunications Standards Institute (ETSI) to assist ITS. Japan has made efforts to encourage Electronic Toll

Collection (ETC) systems, which have reduced vehicle emissions and saved time and money. Table1.1 displays the characteristics of DSRC in these three regions [10].

Table 1.1 DSRC Features.

| Feature | USA | European | Japan |
|---|---|---|---|
| **Radio Band** | 75MHz | 20MHz | 80MHz |
| **Radio Frequency** | 5.9GHz | 5.8GHz | 5.8GHz |
| **Data Rate** | 3-27 Mbps | 350 Kbps | 1-4 Mbps |

The IEEE 802.11p/1609 WAVE protocols, which are specifically created to satisfy the requirements of vehicular communications, serve as the foundation for DSRC. The communication in DSRC is described as being "Short Range" and occurs over distances of hundreds of meters (100m-1000m). Direct V2V and V2R communication can be supported by DSRC without the need for infrastructure. IEEE 802.11p takes into account the situation where nodes are moving quickly and operating in the DSRC frequency band. The connection between automobiles and RSUs in traffic scenarios only lasts for a brief time; Consequently, IEEE 802.11p specifies a method of sending messages between nodes without having to wait for the process to join a Basic Service Set (BSS). This reduction in latency may cause the data to be sent through other available channels [11].

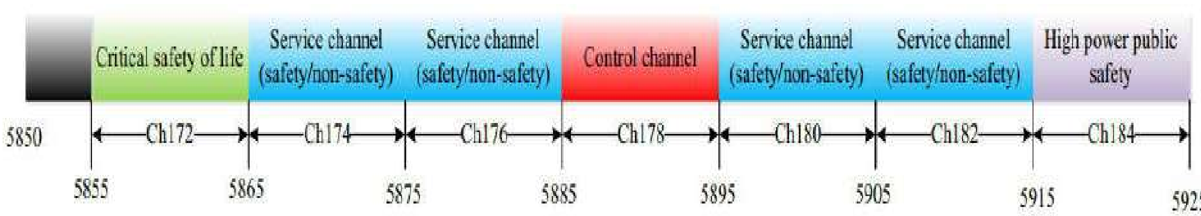



Figure 1.2 DSRC Channels [11].

According to IEEE 802.11p, the 75 MHZ spectrum is split into seven channels, each of which has a 10 MHz bandwidth (Figure 1.2). One of these seven channels (channel 178) is referred to as the control channel (CCH) and it is only used for urgent safety messages. The other two channels are Service Channels (SCH), one on each side of CCH; they only convey application and service data. Sometimes, two nearby SCH might combine to form a single 20 MHZ channel. The final two channels on each end are set aside for special delivery when necessary [11].

### 1.1.4 VANET Applications

Infrastructures alongside roads serve as access points or cache points for information about services. Various real-world criteria could be satisfied by using the interaction between cars or between vehicles and RSUs. Three types of VANET applications can be distinguished, including road safety, traffic efficiency, and entertainment applications [12].

A. ***Road Safety***

In order to prevent some potential problems, vehicles communicate with one another, especially when they are at intersections or on a road with a lot of traffic. Vehicles can be alerted when a vehicle in their blind spot is heading toward them, and they can also be startled when a sudden break occurs in front of them even when it is still a few vehicles away. When an emergency occurs, such as ambulances or fire trucks passing by, vehicles could be warned in advance to move aside. Cooperative message transfer, real-time information assistance, and traffic control notification are the core components of traffic safety applications.

***B. Traffic efficiency***

The effectiveness of the transportation system as a whole is the main emphasis of these applications. As an illustration, you may tell distant vehicles to switch to a road with less traffic density in order to prevent traffic congestion. Typically, this type of application needs a wide affected range. The transmission latency and accuracy are less strict when compared to safety applications.

***C. Entertainment applications***

Various communication technologies are used to support these applications. These applications offer information entertainment and services such as information sharing, internet access, streaming audio, video, etc., to drivers.

## 1.2 Thesis Motivations and Objectives

Among the ITS applications mentioned above, the most urgent and challenging one is the safety-related application. For this kind of application, information dissemination usually requires low latency, high accuracy, and high reliability. At the same time, VANET has a scalability problem that cannot be solved by flat network architecture without a central controller. Thus, researchers have proposed a hierarchical network architecture to solve this problem. In such hierarchical network, vehicles are virtually organised into different groups called clusters. The objective of a clustering algorithm is to partition a network into some subnetworks, each of which has some similar attributes according to an appropriate metric. Designing stable clustering algorithms for vehicular networks is the major purpose of this thesis.

Communication overhead has the issue of scalability and stability in a dynamic environment. VANET is a dynamic and constantly changing topology, so it requires stable clustering to prevent connection failure.

*Our first objective in this thesis is to design clustering approaches for VANET as a complete solution to this challenge in order to guarantee cluster stability using hypergraph theory.*

The formation of stable clusters is a big challenge in designing any clustering algorithm in VANET due to swift changes in vehicle communication links.

*The hypergraph approach is used to comprehend this problem, so the formulation of VANET through the hypergraph is introduced in our work. Then, based on the hypergraph spectral clustering through the tensor trace maximisation method, an optimal number of clusters is formed.*

The work on the real scenario is mostly limited to the highway; the urban scenario analysis is very limited and few. Designing a clustering algorithm for an urban environment is more complicated than designing it for a highway, due to the large number of intersections and the varied speed of vehicles as a result of congestion.

*Our clustering approaches are designed for the urban scenario and tested on the real map of Baghdad's region.*

The cluster head maintains the communication between the cluster vehicles and RSU. The longer stability of the cluster head is required. The stability of the cluster head will be higher if it is in communication link with the neighbouring vehicles for a longer time. Sustaining a cluster head for a long period is difficult.

*So, in our work, some parameters to select the most stable cluster head are introduced. These parameters are:*

- *Vehicle's Relative speed: How close a vehicle's speed to its neighbour's is determined using this parameter.*
- *Neighbouring degree: it is the total number of neighbouring vehicles*
- *The eccentricity which is calculated by an ever-evolving hypergraph and eliminates the need for re-clustering for the dynamic traffic density.*
- *The trust score is inspired by the deep learning-trained spectrum sensing approach. The trust values for the primary users are especially to be evaluated using an adaptive spectrum sensing. Long short-term memory (LSTM), a deep recurrent learning network, is trained for the probability of detection with various signal and noise conditions.*
- *The vehicle's time to leave estimation is calculated with the help of the predicted vehicle position using the Predictive directional greedy routing protocol.*

To increase the clustering stability, two cluster head selection schemes are introduced in this thesis. By using these schemes, strong connectivity and a stable link lifetime are obtained. These two schemes are:

- *The scheme of a cumulative multimetric is first presented in Chapter 3.*
- *A relational analysis is developed using a grey relational analysis model between four cluster head selection parameters instead of knowledge based weightage. This scheme is presented in Chapter 4.*

In summary, this work is our experiment with the use of hypergraph theory at different clustering approaches and cluster head selection parameters.

## 1.3 Thesis Organisation

This thesis brings original contributions, mainly to VANET clustering algorithms. In this thesis, a formulation of VANET through a hypergraph is introduced. It is our experiment with the hypergraph theory. The hypergraph-based spectral clustering method is applied in different clustering approaches. Also, two cluster head selection schemes are presented to increase the clustering stability. In this thesis, we first provide an overview of the existing clustering algorithms in the literature using various terms. Secondly, to obtain stable clustering, we propose an entirely dynamic approach to the clustering formation and maintenance of a VANET structure in an urban scenario. This approach uses a hypergraph spectral clustering model for cluster formation, and the cluster head stability is governed by a cumulative multimetric factor. Thirdly, we try to increase the clustering stability by improving the hypergraph algorithm using the Eigen-trick method, which can speed up the solution of eigenproblems without losing information. The cluster head stability is increased by using a new scheme for selecting the stable cluster head with the help of the grey relational analytical model. The organisation of this thesis is listed in what follows:

- **Chapter 1: General Introduction.**

  This chapter introduces the context and provides a general overview of VANET. VANET characteristics and challenges, communication modes, technologies, and applications are presented. Furthermore, we address the motivations and the desired objectives of this thesis. Finally, we introduce the organisation of this thesis.

- **Chapter 2: State-of-the-Art of Clustering in VANET.**

  In this chapter, we give an overview of the existing clustering algorithms in VANETs. A complete survey on clustering in VANETs is provided based on the clustering process. The clustering process in most algorithms is explored in the aspects of cluster head selection, cluster construction, and cluster maintenance. Then, the existing algorithms are compared in terms of transmission range, vehicle density, vehicle velocity, hop count, and traffic scenario. In addition, a comprehensive analysis of clustering algorithm performance evaluation methods is presented, followed by performance metrics, goals, and simulation tools for each clustering algorithm in the literature. Finally, some of the most important issues in the literature are summarised.

- **Chapter 3: HyperGraph Clustering Model (HGCM) in VANET.**

In this chapter, a newly developed vehicular-hypergraph-based spectral clustering model is introduced. It is called the HyperGraph Clustering Model (HGCM), which aims to improve the cluster's stability. Clusters are formed using the hypergraph through the tensor trace maximisation and the construction of the hypergraph is designed using the distance proximity amongst the vehicles in the network. The network's performance, especially in an urban scene, can be improved by installing auxiliary facilities, such as RSUs. Here, an evolving graph structure of the traffic is conceived using betweenness centrality. The cluster head stability is governed by a cumulative multimetric factor inclusive of relative speed, eccentricity, neighbourhood, and spectrum sensing based on cooperative trust. Trust calculation is performed using deep learning-trained spectrum sensing as a model. The scheme of a cumulative multimetric is introduced through which strong connectivity and stable link lifetime are maintained. The proposed HGCM is tested for various vehicle densities in a real area in Iraq's capital, Baghdad. Compared with individual measures and other techniques in the literature, our cumulative approach significantly improves the clustering stability. The proposed approach also improves the network performance in terms of packet delay and throughput.

- **Chapter 4: EtHgSC: Eigen trick-based Hypergraph Stable Clustering in VANET.**

This chapter is dedicated to improve the hypergraph-based spectral clustering algorithm using the Eigen-trick method. To improve clustering, the Eigen-trick is used to calculate the modified Laplacian value in the tensor trace maximisation. The Eigen-trick improves the clustering efficiency by utilising higher-order information in eigenvalues. It considers the transformable connection between the vertex Laplacian and the hyperedge Laplacian, which can speed up the solution of eigenproblems without losing information.

We introduce a new scheme using the improving hypergraph algorithm for cluster generation. The proposed scheme is named Eigen trick-based Hypergraph Stable Clustering (EtHgSC). The slow dragging and high-speed vehicle management, along with the direction change due to junctions and hybrid velocities, have been taken into consideration in this approach. So, for cluster head selection; in addition to relative speed, neighbouring degree, and eccentricity, the vehicle's time to leave is introduced by using estimating the next vehicle's position. The predicted vehicle direction at the next instant is calculated with the help of the predicted vehicle position by the predictive directional greedy routing protocol to get the necessary estimated time to leave. A relational analysis is developed using grey relational analysis between four cluster head selection parameters instead of knowledge-based weightage.

There is a significant improvement in terms of stability using the EtHgSC model. Our proposed scheme has the highest stability in a comparison with the state-of-the-art techniques, which are the most widely used (High citations) in the literature.

- **Chapter 5: Conclusion and Future Works.**

This chapter concludes this thesis and points out some limitations of the work. Meanwhile, limitations of the thesis will be further considered for future study and research.

# Chapter 2

# State-of-the-Art of Clustering in VANET

## 2.1 Introduction

One of the most important and investigated ITS applications is the safety-related application. To support these services, information should typically be sent with high accuracy and minimal latency. However, VANETs have a highly dynamic topology, with a high relative speed of vehicles and frequent discontinuities in the networks that cause scalability and unstable connection problems. Also, a single vehicle cannot keep global information for a vast and dynamic network like VANETs due to its limited transmission range. These shortcomings lead to topology management in VANETs, that can be done through clustering [13].

One of the most effective unsupervised methods for grouping information is clustering. It is a common VANET technology that offers an appealing approach for simplifying and optimising network functions and services. When compared to the traditional flat structure Figure 2.1 (a), it has dramatically improved performance in a variety of applications. The clustering structure for VANET has more benefits than the flat structure. These benefits include; reducing the overhead, reducing latency, increasing network stability, enhancing the network's scalability and reliability, reducing broadcast storm problems, and mitigating redundant data transmission.

Clustering is a technique for organising network nodes into small groupings called clusters. Typically, vehicles in close proximity are grouped together in a cluster based on various key parameters and metrics. The vehicles present in the cluster are known as [4],[5]:

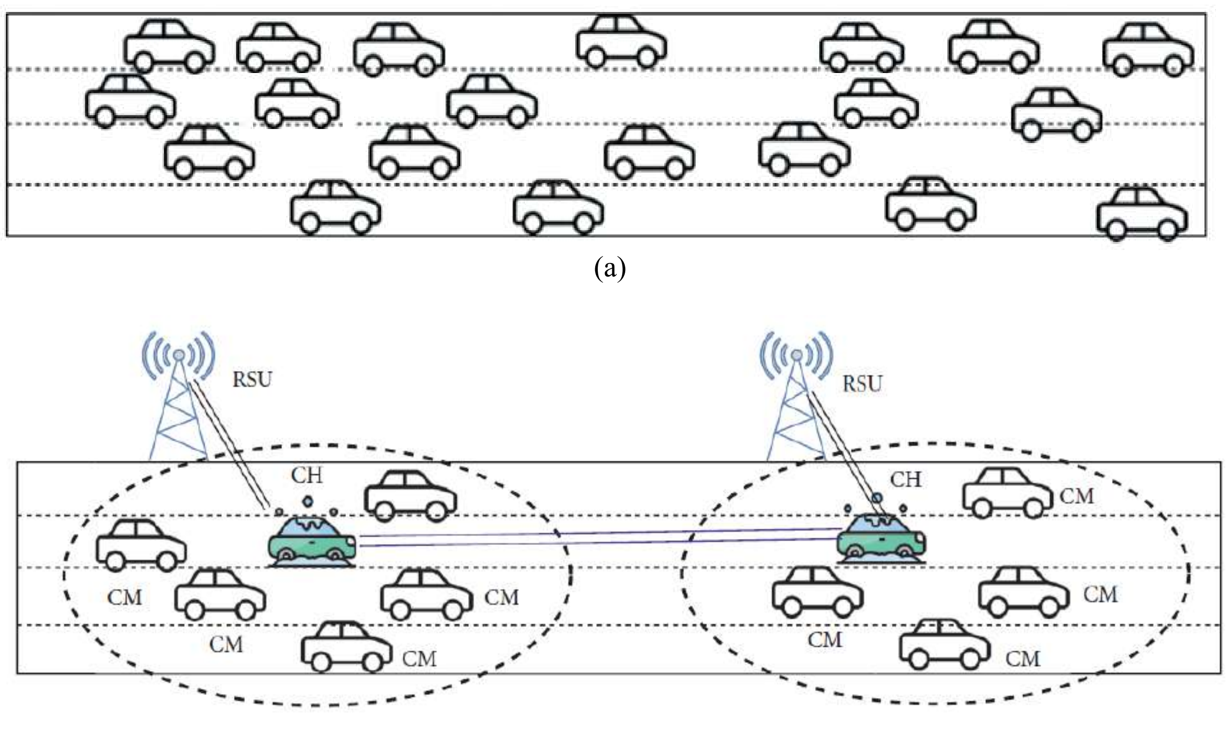
1. Cluster Head (CH): This is the node that is the coordinator or head of the cluster. The CH is selected according to different criteria, and its main task is to allow cluster members to communicate and share information with other members and CHs. Each CH in the cluster is responsible for communicating with its members, the RSU, and the CHs of other clusters.

2. Cluster Member (CM): The remaining nodes in the cluster are the CMs. These nodes exchange information by broadcasting messages to each other.

3. Gateway Node (GW): This node helps to communicate with RSU; it doesn't need to be present in every cluster.

Figure 2.1 (b) illustrates the VANET's cluster-based communication structure. A CH is chosen by defining the network's parameters, and the remaining vehicles are treated as CMs. Internal cluster communication is handled entirely by the CH. There are two specific routing that divide a cluster's internal communication; intra-cluster communication and inter-cluster communication. The formation of a cluster decreases the performance of high load on the VANET nodes [5]. The communication efficiency of VANETs can be enhanced through vehicular node clustering if the clusters are reliable and possess longevity. So, the cluster stability is important for the VANET's reliability and scalability, as it guarantees minimal intra- and inter-cluster communication, lowering the overhead associated with these issues [5].

An efficient clustering technique minimises the overhead of reclustering and facilitates network management. To implement a robust clustering algorithm, there are problems that need to be solved: Clustering methods need to be designed, and metric has to be used to select the CH [14].




(b)
Figure 2.1. VANET (a) Flat Structure and (b) Clustering Structure.

To achieve the best information of communication in VANET, the most recent clustering algorithms are presented. We also concentrate on VANET's intelligent clustering algorithms. This leads us to explore different clustering strategies.

The following is a list of this chapter's main contributions:

- An overview of the development of clustering algorithms in VANETs from 2008 to 2022, which have been observed and studied, is provided. Also, most of these algorithms have never been summarised in previous research.
- The existing clustering techniques are summarised and classified in terms of clustering procedure: Cluster formation, CH selection, and cluster maintenance.
- The clustering techniques in the literature are compared in terms of density, mobility, traffic scenario (environment), topology, and transmission range.
- A comprehensive analysis of the most common parameters used for evaluating the performance of clustering algorithms is introduced, including cluster performance parameters and network performance parameters. Also, simulation tools for each clustering algorithm are presented.
- Some of the most important issues in the literature are summarised, and then our proposed solution is introduced, which will be presented in detail in the next chapter.

## 2.2 Clustering Algorithms History

In the early 1990s, the clustering techniques for VANETs began to be developed and were expanded after 2005 [13].

Researchers discovered that prior clustering algorithms in MANETs were no longer appropriate for VANETs due to their predictable mobility and specified route topology. Additional control overheads may be imposed due to the time it takes to complete the clustering phases. As a result, a good clustering method should construct a small number of clusters and dynamically maintain the cluster structure without creating significant network overhead. Some MANET clustering methods were developed to address the unique requirements of vehicular communications. Furthermore, most of the clustering algorithms were derived from the previous MANET clustering algorithms, including the Mobility Based Metric for Clustering (MOBIC) in [15], Weighted Clustering Algorithm (WCA) in [16], and Distributed and Mobility Adaptive Clustering (DMAC) in [17]. Several clustering techniques for VANETs have been proposed, particularly after 2010 as a result of the expansion and development of the VANET and also to meet the need of increasing the clustering stability and solving some problems. In Table 2.1, some VANET clustering algorithms are highlighted, which have been presented from 2008 to

2022. Also, the number of citations for each algorithm is highlighted. We note that the Hybrid Vehicular Multi-hop algorithm for Stable Clustering (VMaSC-LTE) has the highest citations, and the Passive Multi-hop Clustering algorithm (PMC) has the highest mean citations.

Table 2.1 Various Clustering algorithms with their number of Citations.

| Ref | Year | Algorithm | Abbreviation | Citation | Mean |
|---|---|---|---|---|---|
| [18] | **2008** | Position-based Prioritized Clustering | PPC | 179 | 12.8 |
| [19] | **2009** | Robust Mobility Adaptive Clustering | RMAC | 77 | 5.9 |
| [20] | **2009** | Affinity Propagation | APROVE | 209 | 16 |
| [21] | **2009** | Density Based Clustering | DBC | 80 | 6.15 |
| [22] | **2010** | Aggregate Local Mobility | ALM | 133 | 11.1 |
| [23] | **2010** | Cluster-Based Directional Routing Protocol | CBDRP | 74 | 6.2 |
| [24] | **2011** | Vehicular clustering based on the Weighted Clustering Algorithm | VWCA | 158 | 14.3 |
| [25] | **2011** | Adaptive Service Provider Infrastructure | ASPIRE | 14 | 1.3 |
| [26] | **2011** | Zhang et al. [26] | - | 141 | 12.8 |
| [27] | **2012** | Spring-Clustering | Sp-Cl | 58 | 5.8 |
| [28] | **2012** | Stability-Based Clustering Algorithm | SBCA | 63 | 6.3 |
| [29] | **2012** | Threshold Based algorithm | TB | 207 | 20.7 |
| [30] | **2012** | Fuzzy Logic Based clustering Algorithm | FLBA | 82 | 8.2 |
| [31] | **2012** | Mobility-Aware Clustering Algorithm based on Destination positions | AMACAD | 78 | 7.8 |
| [32] | **2012** | Trust dependent Ant Colony Routing | TACR | 54 | 5.4 |
| [33] | **2013** | Agent Learning–based Algorithm | ALCA | 87 | 9.6 |
| [34] | **2013** | Vehicular Multi-hop algorithm for Stable Clustering | VMaSC | 98 | 10.9 |
| [35] | **2014** | Arkian et al. [35] | - | 52 | 6.5 |
| [36] | **2015** | Distributed Multi-hop Clustering based on Neighbourhood Follow | DMCNF | 103 | 14.7 |
| [37] | **2015** | Adaptive Weighted Clustering Protocol | AWCP | 61 | 8.7 |
| [38] | **2015** | Aggregate Relative Velocity | ARV | 35 | 5 |
| [39] | **2015** | Hybrid Vehicular Multi-hop algorithm for Stable Clustering | VMaSC-LTE | 365 | 52.1 |
| [40] | **2016** | Mobility-aware and Single-hop Clustering scheme | MOSIC | 11 | 1.8 |

| | | | | | |
|---|---|---|---|---|---|
| [41] | **2016** | New Clustering Algorithm Based on Agent Technology | NCABAT | 11 | 1.8 |
| [42] | **2016** | Clustering-Based VANET Routing algorithm Protocol | CBVRP | 36 | 6 |
| [43] | **2017** | A Mobility-based Scheme for Dynamic Clustering in VANETs | MoDyC | 126 | 25.2 |
| [44] | **2017** | K-Mean and Floyd-Warshall algorithms | KMFW | 21 | 4.2 |
| [45] | **2017** | Cluster-Based Life-Time Routing | CBLTR | 88 | 17.6 |
| [46] | **2017** | Stable Clustering algorithm for vehicular ad hoc networks | SCalE | 23 | 4.6 |
| [47] | **2018** | Passive Multi-hop Clustering Algorithm | PMC | 221 | 55.25 |
| [48] | **2018** | Link Reliability-based Clustering Algorithm | LRCA | 37 | 9.25 |
| [49] | **2018** | Unified Framework of Clustering approach | UFC | 73 | 18.25 |
| [50] | **2018** | Enhanced Control Overhead Reduction Algorithm | ECORA | 3 | 0.75 |
| [51] | **2018** | Normalized Multi-Dimensional Parameter based Affinity Propagation Clustering | NMDP-APC | 7 | 1.75 |
| [52] | **2019** | Saleem et al. [52] | - | 17 | 5.7 |
| [53] | **2019** | Double-Head Clustering | DHC | 46 | 15.3 |
| [54] | **2019** | Enhanced Weight-based Clustering Algorithm | EWCA | 30 | 10 |
| [55] | **2019** | Hybrid Clustering Algorithm based on Roadside | HCAR | 21 | 7 |
| [56] | **2019** | Center-Based Clustering algorithm | CBSC | 47 | 15.6 |
| [57] | **2019** | Mobility Based Clustering Algorithm | MBCA | 6 | 2 |
| [58] | **2019** | Cluster-based VANET oriented Evolving Graph | CVoEG | 35 | 11.6 |
| [59] | **2019** | Fuzzy-based Cluster Management Scheme | FCMS | 31 | 10.3 |
| [60] | **2019** | Probabilistic- Direction-Aware Cooperative Collision Avoidance | P-DACCA | 23 | 7.6 |
| [61] | **2019** | Chain-Branch-Leaf | CBL | 14 | 4.6 |
| [62] | **2019** | Abbas et al. [62] | - | 9 | 3 |
| [63] | **2019** | Moore et al. [63] | - | 4 | 1.3 |
| [64] | **2020** | Grasshoppers' Optimization-based node clustering Algorithm | GOA | 22 | 11 |
| [65] | **2020** | Collaborative Clustering Approach for the Internet of Vehicles | CCA-IoV | 4 | 2 |
| [66] | **2020** | Adaptive Clustering with Optimal Stability | ACOS | 0 | 0 |

| | | | | | |
|---|---|---|---|---|---|
| [67] | **2020** | Diverted Path Approach | DPA | 9 | 4.5 |
| [68] | **2021** | Junction-based Clustering for VANET | JCV | 4 | 4 |
| [69] | **2021** | Maan et al. [69] | - | 5 | 5 |
| [70] | **2021** | Khan et al. [70] | - | 4 | 4 |
| [71] | **2021** | K-Means clustering method, a new routing protocol | KMRP | 9 | 9 |
| [72] | **2022** | Region-based Collaborative Management Scheme | RCMS | 2 | 2 |

## 2.3 Clustering Process in VANETs

The cluster establishment in VANET communication process is the most important part. There are two phases to complete this process:

• First phase- (Cluster Generation): which has two major components; the cluster formation process and the CH selection process; during this phase, nodes send advertisement messages to pick the primary CH and CMs, and subsequently regular data packets are transmitted between them. In order to create a stable cluster, there may be a few techniques added between the advertisement message transmission and CH selection.

• Second phase- (Cluster Maintenance): Stable cluster merging, selection of secondary CH, re-clustering, and cluster splitting are some of the maintenance methods.

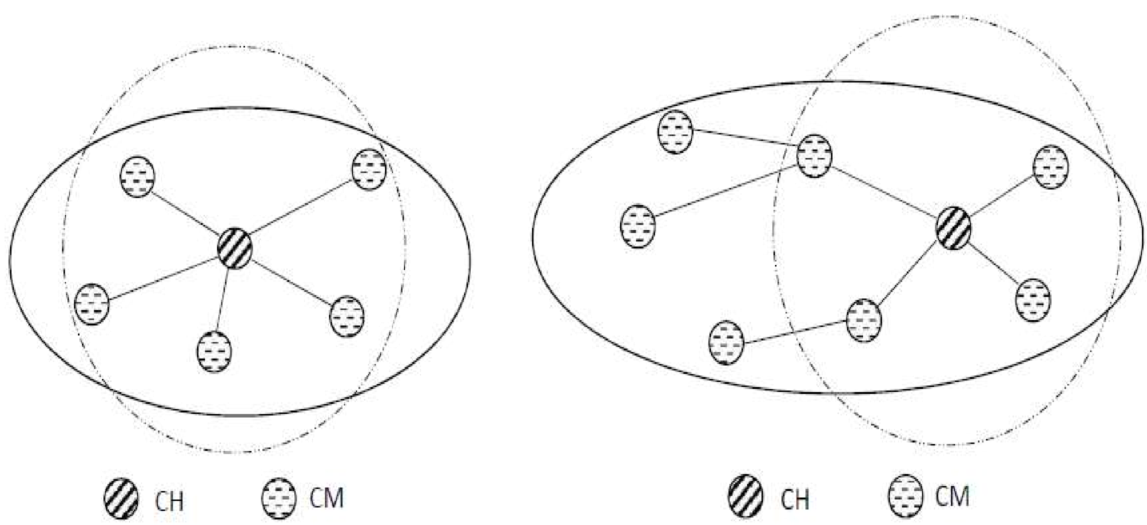
The cluster formation and maintenance work together throughout the whole operation period. The formation phase provides a cluster based on the current and predictable network topology, while the maintenance phase reduces the impact caused by unpredictable network changes. As mentioned previously, the clustering formation method and the metrics used for selecting the CH are the two major issues that need to be considered for designing an efficient clustering algorithm [14].
Few researchers in the literature have discussed these phases separately. This section provides an overview of the algorithms and criteria used in each clustering step, including cluster formation, CH selection, and cluster maintenance.

### 2.3.1 Cluster Generation Phase

This phase goes through two processes to complete the generation of clusters; Cluster formation process and CH selection process. To complete the clustering generation phase, some algorithms form the clusters and then select the CH and others vice versa.

#### 2.3.1.1 Cluster Formation

Cluster formation can help reduce the network load and increase overall efficiency. Each cluster has two or more members and selects its CH, which is responsible for faster communication with a minimum delay. Cluster formation in VANET has various types and categories: in terms of topology: single-hop vs. multi-hop, in terms of the environment: highway vs. urban, in terms of models: center-based vs. distributed-based, in terms of density: dense vs. sparse, in terms of speed: high speed vs. slow speed. Information used for VANET clustering can be topology information and mobility information such as speed and acceleration. This section classifies the clustering algorithms on the basis of topology Figure 2.2. That means a cluster structure in VANETs can be modeled according to the hop distance that separates the CH and its members, transmission range, and cluster radius.




(a) (b)

Figure 2.2. Cluster Topology a) Single hop and b) Multi-hop.

### *a. Single-hop Clustering Algorithms*

It is the algorithm that creates clusters with a single-hop distance between each node and its CH. That means every node connects directly with the CH [13]. Many clustering algorithms form directly single-hop clusters based on the vehicle transmission range or the limited cluster radius. Some of the single-hop clustering algorithms are:

In [19], the authors presented RMAC algorithm, which identified a one-hop neighbour and selected a CH using the relative node mobility metrics of speed, location and travel direction. An evolving structure was created via neighbourhood analysis using the node precedence algorithm. This approach facilitates geographic routing by producing accurate neighbour information. However, frequent change of CH is the issue, which was not dealt with in this algorithm.

With the same mobility concept, another single-hop schemes were presented in [20] and [43]. APROVE algorithm was proposed using the affinity propagation technique in a distributed manner [20]**.** The authors claimed the existence of clusters with high stability.

One of the major issues in the VANET is the security because of its impact on the performance of the network, it is done in VWCA in [24], and ALCA in [33]. VWCA algorithm also improves the connectivity, and stability performance; it is a single-hop clustering algorithm. The connectivity can be increased using the adaptive transmission range algorithm (AART) which is based on detected short-range communication standards. The AART helps to extend the transmission range dynamically from 100 m to 1000 m based on the vehicles' density.

Koulakezian in [25] designed ASPIRE architecture by using the concept of vehicular mobility in a highway scenario. The author allowed vehicles to connect to the network through regular mobile IP nodes, thereby increasing the connectivity and decreasing the overhead by caching in clusters.

The stability of the CH in VANET was enhanced by designing the SBCA algorithm in [28]. This algorithm provides a more stable structure according to vehicle mobility and the number of neighbours. This algorithm achieves the extension of the CH lifetime. The cluster formation procedure does not take the vehicle's direction into account, which has an impact on the VANET system's performance.

The fuzzy logic with software-defined networking to improve the cluster stability was used in [30] and [69], but Fuzzy logic usage imposes a knowledge constraint to define the optimal rules.

The authors in [30] suggested a distributed and dynamic CH selection criterion in order to dynamically organise the network into clusters. The FLBA method contains a learning mechanism that predicts the future positions and speeds of all CMs using a fuzzy logic inference system.

AMACAD is a single-hop clustering algorithm for VANETs; it was proposed in [31]. This algorithm takes into account the destination of the vehicles to arrange the clusters and implements an efficient message mechanism to respond in real-time and avoid global re-clustering.

In [40], the authors proposed the MOSIC algorithm. Gauss Markov mobility (GMM) model is used by this proposed for mobility prediction that makes a vehicle able to prognosticate its mobility relative to its neighbours.

NCABAT was introduced in [41], it is a single-hop clustering algorithm. The objective of this algorithm is to describe the agent properties of vehicles with the purpose of improving traditional schemes in terms of performance.

Some algorithms, like KMFW in [44] and KMRP in [71], used a K-mean method and integrated it with other algorithms to form stable clusters. The K-mean method is one of the simplest unsupervised clustering methods.

LRCA algorithm was proposed in [48] to grant reliable and efficient data transmission in urban VANET. LLT-based neighbour sampling scheme is used to select a group of vehicles with stable neighbour vehicles. In this proposed, the routing approach is prepared to support infotainment services in VANET, which are not strict in delay constraints. Better clustering stability in terms of long CH duration, long CM duration, and low rate of CH change was acquired using LRCA.

The authors in [53] presented a robust DHC method for VANET. It is a single-hop clustering algorithm that aims to increase cluster stability and minimise the number of clusters in the network under different scenarios and conditions.

EWCA was proposed in [54], it is a single-hop clustering algorithm. In this algorithm, the vehicles with the same road ID and within the transmission range of their neighbours are suitable for the cluster formation process. In an emergency message transmission case, this technique reduces the formation of unstable clusters and enhances the clustering stability. The simulation results have proven that the cluster stability and overhead latency reduction are superior.

HCAR was proposed in [55], it is a single-hop clustering method. The RSUs were used to design it for the Internet of Vehicles (IoV). The HCAR algorithm is centralised in the distributed RSUs. After the RSUs have been managed, the clusters are formed using a graph theory-based approach. The weight mechanism is used to choose secondary CH to address the problem of CH unavailability. This proposed improves the stability of CH.

The authors in [56] proposed the CBSC algorithm to help self-organized VANETs form stable clusters and decrease the status change frequency of vehicles on highways. Also, a new CH selection algorithm was presented to minimise the impact of vehicle motion differences. In this scheme, two metrics are introduced to enhance VANET's security. The results showed that the proposed technique has a low packet loss rate and high stability.

The authors in [61] introduced the Multi-Point Relay (MPR) and CBL algorithm. MPR aids in reducing redundant transmission throughout the network. In CBL algorithm, the vehicles that move in the same direction and with lower speed are suitable for building a backbone (chain). This algorithm increases the CH lifetime and creates a stable cluster.

In [65], the authors presented a new collaborative clustering algorithm CCA-IoV based on node score, in which the CH is chosen from a vehicle with a high node score. The distance, degree of a node, average relative speed, link stability, and average relative acceleration are the parameters used to determine the node score. To improve cluster stability, they also presented a vice CH; however, the mechanism by which a vice CH increases cluster stability is not adequately understood.

In [68], the authors proposed a robust and dynamic mobility-based clustering scheme called JCV. It is a single-hop clustering algorithm and clusters were generated by considering the moving direction of a vehicle at the junctions, vehicles' density, and transmission range. The CH was selected by relative position and time spent on the road. This technique provided high stability, preventing clusters from breaking at the junction frequently.

Single-hop clustering algorithms provide more reliable intra-cluster communication and highly effective coordination to CHs. The coverage area of this type of cluster is small, which leads to high maintenance overhead and a large number of clusters.

Also, in single-hop clustering algorithms, when the density of vehicles is low, the vehicle may not find any member and stay single, so it cannot form a cluster. This case should be avoided because the cluster performance will be decreased. To avoid this problem, the minimum number of vehicles can be limited in a cluster. We can summarise that the single-hop algorithms provide good cluster stability and low latency, but clustering coverage requires further improvement.

### *b. Multi-hop Clustering algorithms*

Clusters are generated with multi-hop distance, where every node is at a distance of at most multi-hop from its CH. Some multi-hop clustering algorithms are presented in this section.

In [18], the authors proposed a multi-hop clustering technique known as PPC in which cluster structure is determined by the geographic location information and priorities assigned to vehicles. Each node broadcasts information related to itself and its neighbours. In this approach, the CH election process is similar to the computation of minimum dominating sets used in graph theory. A node with the highest priority is selected as a CH.

DBC was introduced in [21], it is a multi-hop clustering model using YATES algorithm to achieve stability in clusters. In this proposed the cluster formation is based on a complex clustering metric that takes into account the density of the connection graph, link quality, and traffic conditions.

The ALM algorithm was proposed in [22]. This algorithm followed a similar cluster formation process and CH selection. Each vehicle calculates its overall mobility variance neighbours. Lower variance equates to less mobility and greater stability. As a result, a car with less variance than its neighbours is a good candidate for a CH. Also, the proposed algorithm avoids clusters from merging frequently in order to improve cluster stability.

Some algorithms were introduced to solve the problem of fast data transmission and link connectivity, like the CBDRP in [23]. It is a multi-hop clustering algorithm, and the clusters are formed based on moving direction and radio communication. This approach reduces latency and increases packet delivery ratio and link stability.

The same ideology of mobility in [25] was reintroduced using the concept of multi-hop, and a new clustering scheme based on multi-hop was presented by Zhang et al. [26].

In [27], the authors proposed a distributed clustering algorithm that forms stable clusters based on force-directed algorithms; it is named as Sp-Cl scheme. Every node applies to its neighbours a force according to their distance and their velocities. Vehicles that move in the same direction or toward each other apply positive forces while vehicles moving away apply negative forces. This work also proposed mobility metric based on forces applied between nodes according to their current and future position and their relative mobility.

A multi-hop clustering approach was also presented in [29] called TB with the goal of maximising the stability of the network topology and decreasing network dynamics. The speed difference among vehicles as well as the position and the direction were taken into account in this proposed during the cluster formation process. This algorithm increased the stability of the network by increasing CH lifetime and minimising vehicle transitions between clusters.

Some algorithms like TACR in [32] improve the routing overhead by using a new technique to select the CH by taking into consideration the trust value and the updated position of vehicles.

VMaSC was proposed in [34], this algorithm is used to construct stable multi-hop clusters with a minimum number of CHs in VANET. The CH is selected based on the node with the least mobility via multi-hops, which is determined as a function of the speed differential between neighbouring nodes. Also, the VMaSC was improved in [39] by integrating IEEE 802.11p-based multi-hop clustering and the Long-Term Evolution (LTE).

The speed difference between neighbouring nodes was taken into account to obtain a stable clustering structure [35].

In [36], DMCNF was proposed to solve the network weaknesses that occur as a result of a dynamic topology. It allows cars to choose their targets in a distributed manner among one-hop neighbours on a regular basis. The selection of CH is determined by the vehicle's relation to its neighbours. The cluster's stability is improved by this algorithm.

AWCP was introduced by taking into account the speed information, direction, position, and highway ID to select the most stable vehicles among current vehicles to operate as CHs [37]**.** This algorithm uses highway ID information to maximise cluster structure stability. This technique improves cluster lifetime and minimises communication overhead.

In [42], for desert and rugged situations, a VANET-based clustering routing protocol was introduced. The source and destination vehicles work to keep the stability of cluster architecture. The designed algorithm's tasks are CH selection, cluster structure formation, and routing protocols.

The lack in clustering algorithms performance in VANET in terms of stability and reliability was solved by the PMC algorithm [47]**,** which is a multi-hop algorithm. Clustering is presented in this approach based on the priority neighbour following strategy. This strategy can improve the stability of clusters and reduce the cost of clustering effectively.

One of the major issues in VANET is cooperative collision avoidance (CCA) which has an impact on cluster stability. In various two directions of real traffic scenarios, a probabilistic direction aware (PDA) algorithm was proposed for dominant CCA [60]. Cluster formation is handled using a modified K-medoids method that integrates the Hamming distance metric for direct knowledge. The distance and speed of nodes are used to calculate a collision's probability between the vehicles. The benign factor is used to determine the vehicles' optimal safe speed, which is compared to the threshold range and delivers a collision warning. The communication overhead and collision latency are decreased.

Grasshopper optimisation tunes the network parameters for enhanced stability [64]. But, the use of an optimisation algorithm in VANET is not feasible due to the frequently changing topology.

Other algorithms proposed to increase the vehicular communication reliability [51],[67]**.**

Multi-hop clustering algorithms can reduce the number of clusters, expand the cluster coverage area, and enhance cluster stability. We can summarise that the multi-hop algorithms provide high clustering coverage, and good cluster stability, especially with regard to the number of CMs re-affiliations, CH changes, and cluster lifetime. However, multi-hop cluster formation is more complex, which will take a long cluster formation time, and this may cause a delay in transmitting the information. In addition, if the connection is multi-hop, then the data loss may be high as the carrying vehicle may change its direction or speed, also, the cluster overhead requires more improvement.

Also, according to some simulation results, the cluster performance degrades when the number of hops is more than three [34]. This means when the hop count increases, the cluster performance will decrease.

The authors in [38], [45], [46], [50], [52], [57], [58], [59], [62], [63], [66], and [70] didn't mention whether the connection between their CHs and members is single-hop or multi-hops, each of these algorithms is formed according to different criteria and presented to achieve a specific goal.

In [38], a hybrid backbone-based clustering algorithm for VANETs was proposed. The concept of the number of links and vehicular mobility is used for cluster formation and CH selection. During cluster formation, nodes with a relatively higher degree of connectivity initially form a backbone that is designated as leadership. The leadership then participates in CH election and efficient cluster re-organization using an aggregate relative velocity of vehicles in the leadership.

The authors in [45] presented CBLTR algorithm to enhance the overall clustering protocol performance. This algorithm increases the throughput and the route stability.

Other algorithms, like the ECORA in [50] proposed aimed to reduce the routing overhead. For CH selection, this algorithm used CBLTR protocol. ECORA reduces the control overhead by limiting frequent CH advertising on the highway.

In [57], the MBCA was used to solve the problem of multimedia broadcasting content in a hybrid VANET topology. Cluster formation and CH selection are based on mobility measurements, which are utilised to determine the vehicles' relative speeds. The cluster's stability is improved using the handshake process.

The authors in [52] and [59] used the fuzzy logic scheme to improve the network's reliability and stability. FCMS was implemented for detecting a reliable vehicle. For clustering in VANET, FCMS1 and FCMS2 models are compared. Three input factors are for FCMS1. Vehicle trustworthiness (VT) is the fourth input factor in the FCMS2 model. The FCMS model improves the cluster stability and security.

In [58]**,** the authors presented the CVoEG model to improve the reliability and scalability of vehicular communications. The clusters are formed based on graph theory, and the vehicles are divided into an optimal number of clusters (ONC) by using the Eigen gap heuristic. A vehicle with a maximum Eigen-centrality score will be selected as a CH.

Other algorithms like in [62] and [63] proposed to achieve a reduction in delay.

ACOS algorithm [66] and SCalE [46] were proposed in to maximise cluster stability in vehicular networks. ACOS scheme is based on a heterogeneous vehicular network architecture, which allows the coexistence of DSRC and cellular networks for vehicular communications. The scheme uses the predicted driving behaviour of vehicles over a time horizon to maximise the clusters' lifetime.

Another algorithm was presented by Khan et al. in [70]**;** it is a heuristic clustering algorithm. The authors used connectivity-based CH selection and calculated the eccentricity for it. The highest eccentric vehicle in a cluster was assigned as the CH. This algorithm forms stable clustering and increases the probability of connectivity of the elected route.

#### 2.3.1.2 CH Selection

The network's robustness and scalability are strongly influenced by CH stability. The CH selection process is one of the most important parts of clustering. The stable CH guarantees that intra- and inter-cluster communications are kept. Each CH in the cluster is responsible for communicating with its members, RSU, and CHs of other clusters. The nodes' interaction with other CH and their neighbours influences the selection of CH. To improve VANET stability, a reliable vehicle can only be a CH. If a metric can resist unnecessary changes while maintaining connectivity, it is robust. A suitable metric can reflect the features of nodes and improve the performance of the clustering algorithm. The researchers considered various metrics for selecting the CH, such as received signal strength, relative speed, position, direction, and link lifetime. Each metric has some sort of precedence over the others because there isn't a single metric that can be the winner in all scenarios. Many clustering approaches are relying on a combination of multiple metrics rather than a single metric for selecting the CH. Table 2.2 shows the CH selection metrics for each clustering approach in the literature.

Table 2.2 CH Selection Metric for the Algorithms in Literature

| **Year** | **Algorithm** | **CH selection metric** |
|---|---|---|
| **2008** | PPC | ID, travel time, and relative velocity |
| **2009** | RMAC | Speed, location, and direction |
| **2009** | APROVE | Distance and speed |
| **2009** | DBC | SNR, distance, and velocity |
| **2010** | ALM | Lowest Variance |
| **2010** | CBDRP | Moving direction |
| **2011** | VWCA | Distrust Level, degree, velocity, and direction |
| **2011** | ASPIRE | Network parameters |
| **2011** | Zhang et al.[26] | Relative mobility |
| **2012** | Sp-Cl | Relative velocity and distance |
| **2012** | SBCA | Relative Speed and RSS |
| **2012** | TB | Distance and relative velocity |
| **2012** | FLBA | Weighted stabilization factor |
| **2012** | AMACAD | Destination , distance, and relative velocity |
| **2012** | TACR | Position and trust value of vehicles |
| **2013** | ALCA | Velocity |
| **2013** | VMaSC | Average speed |
| **2014** | Arkian et al. [35] | Speed, and neighbours |
| **2015** | DMCNF | The propagation delay ratio and number of the following car |
| **2015** | AWCP | Highway ID, direction, position, and speed |
| **2015** | ARV | Relative velocity |

| | | |
|---|---|---|
| **2015** | VMaSC-LTE | Average speed |
| **2016** | MOSIC | Relative speed, relative distance, and relative mobility |
| **2016** | NCABAT | Lowest ID |
| **2016** | CBVRP | Velocity and location |
| **2017** | MoDyC | Relative position, moving direction, and link lifetime |
| **2017** | KMFW | Average distance |
| **2017** | CBLTR | Speed, life time, distance from threshold |
| **2017** | SCalE | Vehicle intentions |
| **2018** | PMC | Speed, Neighbours, Link lifetime , and Position |
| **2018** | LRCA | link reliability |
| **2018** | UFC | Relative position, relative velocity, and link lifetime |
| **2018** | ECORA | Life time |
| **2018** | NMDP-APC | Position and speed |
| **2019** | Saleem et al.[52] | Network Connectivity Level, speed, trust value, lane weight, distance |
| **2019** | DHC | Signal Strength, Relative Speed, Link Lifetime |
| **2019** | EWCA | Speed and Position |
| **2019** | HCAR | Lowest ID |
| **2019** | CBSC | Position and relative Speed |
| **2019** | MBCA | Relative velocity and direction |
| **2019** | CVoEG | Link Lifetime, position, and relative Speed |
| **2019** | FCMS | Relative speed, degree, security, and trustworthiness |
| **2019** | P-DACCA | Distance |
| **2019** | CBL | Direction and relative Speed |
| **2019** | Abbas et al. [62] | Avg distance, speed, time duration |
| **2019** | Moore et al.[63] | Speed, node degree, position |
| **2020** | GOA | Grasshoppers' optimization-based node |
| **2020** | CCA-IoV | Average relative speed, link stability, average relative acceleration, and distance |
| **2020** | ACOS | Driver behavior |
| **2021** | JCV | Movement at the junction, relative position, time, and degree of a node |
| **2021** | Maan et al. [69] | Network connectivity level, lane weight, average velocity, and average distance |
| **2021** | Khan et al. [70] | Link connectivity |

| | | |
|---|---|---|
| **2021** | KMRP | Velocity, free buffer size, and node degree |
| **2022** | RCMS | Using SRP model, relative distance, and speed |

### 2.3.2 Cluster Maintenance Phase

Because of VANETs dynamic topology, severe packet loss occurs due to frequent vehicle re-connection and disconnection. The cluster maintenance process ensures strong connectivity by reducing frequent vehicle re-clustering and also achieves a stable link lifetime through CH. Additionally, a successful cluster maintenance method is necessary to prevent unnecessary cluster re-formations. Cluster maintenance involves vehicle joining, vehicle leaving, cluster merging, selecting secondary CH, and other cluster maintenance methods [4]. Some algorithms neglected this phase and did not mention it. The maintenance methods that have been used in the literature are presented in this section.

In the vehicle joining and vehicle leaving processes, the CH sends frequent signals, and if it receives any signal from a vehicle, this new vehicle is assigned to that cluster and becomes the CM of that particular cluster. Then the CH will update its local database. When the CH loses the connection with a member vehicle, the information for that member is deleted from the CH's local database. AWCP, CCA-IoV, DMCNF, and SCalE algorithms have used this method.

The second method is the cluster merging process; it is more complex than the first one. Cluster merging takes place when two or more clusters can be represented by a single merged cluster, which can minimise the clusters' number and improve the clustering efficiency. The conditions of the cluster merging are different for each algorithm. For example, in the ALM algorithm, cluster merging occurs if two CHs are in each other's transmission range. Whereas, some algorithms like PPC used a distance threshold to control cluster merging, where the cluster merging occurs between two CHs if their distance is less than the dismiss threshold. In the VMaSC algorithm, the averaged relative speed of the two neighbouring CHs; referred to as AVGREL-SPEED, is compared. The CH with the higher average relative speed relinquishes his CH job and becomes a CM for the CH with the lower average relative speed. Also, the PMC algorithm uses the cluster merging method in the cluster maintenance phase, the CH node sends merge request packets to other neighbouring CH to request cluster merging. If one of these two CHs has smaller following vehicles and high relative speed, the merging process is performed.

Other algorithms addressed the two processes (cluster merging and vehicle leaving or joining) in the cluster maintenance phase, like TB, SP-CI, LRCA, MoDyC, UFC, and JCV.

In CVoEG, a couple of vehicles may cause out-of-range communication due to small variations in speed or direction, which can change the shape of the cluster. Cluster splitting is a good solution to maintain the performance of the CVoEG model. Whereas, when two moving clusters get close to each other, the distance between their CHs becomes shorter, thus cluster merging is typically initiated by CH.

A selected secondary CH is another approach used in the cluster maintenance phase. The secondary CH is selected by the CH according to different criteria. It resolves the unavailability of CH to increase the clustering stability. Some algorithms, like EWCA, CBDRP, SBCA, MBCA, and HCAR have used this method.

Some algorithms used another cluster maintenance method, like the maintenance phase in the FLBA algorithm, which is adjustable to drivers' behaviour along the way and has a learning technique for predicting the future position and speed of all CMs using a fuzzy logic inference system.

## 2.4 Clustering Algorithms Comparison

A comparison between the exiting clustering algorithms is shown in Table 2.3 in terms of transmission range, vehicle density, vehicle velocity, hop count, and traffic scenario.

Table 2.3 Clustering Algorithms Comparison

| Year | Algorithm | Transmission Range | Vehicle Density | Vehicle Velocity | Hop Count | Traffic Scenario |
|---|---|---|---|---|---|---|
| **2008** | PPC | 250 m | 100 | - | Multi | Highway |
| **2009** | RMAC | 250 m | 25,50,75 | 22 and 36 m/s | Single | Highway |
| **2009** | APROVE | 250 m | 100 | 15, 25, 35, | Single | Highway |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | | | 40, 50 m/s | | |
| **2009** | DBC | 250m | 100-500 | 11-31 m/s | Multi | Urban |
| **2010** | ALM | - | 30-200 | 2,5,10 m/s | Multi | Random |
| **2010** | CBDRP | - | 60 | 25-35m/s | Multi | Highway |
| **2011** | VWCA | Dynamic 100-1000 m | 10- 350 | 19-33.3 m/s | Single | Highway |
| **2011** | ASPIRE | 250 m | 400 | 11.1 , 22.2, 33.3, 44.4 m/s | Single | Highway |
| **2011** | Zhang et al. [26] | 120 m | 100 | 10- 35 m/s | Multi | Highway |
| **2012** | Sp-CI | 80,125 m | 20-150 | 22-44m/s | Multi | Highway |
| **2012** | SBCA | 300m | 50-150 | 25-35 m/s | Single | Highway |
| **2012** | TB | 150-300 m , 800-1000 m | 400 | 19,25,30 m/s | Multi | Highway |
| **2012** | FLBA | 200 m | 0.05/-0.4/m | 22-33.3 m/s | Single | Highway |
| **2012** | AMACAD | 100-200 m | 50 | 11-31 m/s | Single | Urban |
| **2012** | TACR | 250,500 m and dynamic | 50,100,150 | - | Multi | Highway |
| **2013** | ALCA | 200 | 400 | 13-22 m/s | Single | Highway, urban |
| **2013** | VMaSC | 100-300 m | 100 | 10-35 m/s | Multi | Highway |
| **2014** | Arki et al. [35] | - | 90 | 16-33 m/s | Multi | Highway |
| **2015** | DMCNF | 100-300 m | 100 | 10-35 m/s | Multi | Highway |
| **2015** | AWCP | 1000 m | 25-200 | 33.3- 41.6 m/s | Multi | Highway |
| **2015** | ARV | 250 m | 100,300,500 | 10-30 m/s | - | Urban |
| **2015** | VMaSC-LTE | 100-300 m | 100 | 10-35 m/s | Multi | Highway |
| **2016** | MOSIC | 200 m | 100 | 10-35 m/s | Single | Highway |
| **2016** | NCABAT | 150 m | 60 | - | Single | Random |
| **2016** | CBVRP | 1000 m | 20-200 | 0-33 m/s | Multi | Desert |
| **2017** | MoDyC | 200 m | 100 | 10-40 m/s | Single | Highway |
| **2017** | KMFW | 100-1000 m | 20-250 v/km | 10-40 m/s | Single | Highway, Urban |

| | | | | | | |
|---|---|---|---|---|---|---|
| **2017** | CBLTR | 250 m | - | 3-17 m/s | - | Highway |
| **2017** | SCalE | 150,250,350m | 560 | 22 – 33 m/s | - | Highway |
| **2018** | PMC | 100-300 m | 100 | 10-35 m/s | Multi | Random |
| **2018** | LRCA | 200,500 m | 1500 | 10-30 m/s | Single | Urban |
| **2018** | UFC | 300 m | 200 | 10-35 m/s | Single | Highway |
| **2018** | ECORA | 250 | 400 | 3-17 m/s | - | Highway |
| **2018** | NMDP-APC | - | - | 25 m/s | - | Highway |
| **2019** | Saleem et al.[52] | 300 m | - | 5-10 m/s | - | Highway |
| **2019** | DHC | 300 m | 50-200 | 13.8- 30 m/s | Single | Highway, urban |
| **2019** | EWCA | 300 m | 50-150 | 30 m/s | Single | Highway |
| **2019** | HCAR | 100-300 m | 100 | 10-40 m/s | Single | Highway |
| **2019** | CBSC | - | - | 55.55 m/s | Single | Highway |
| **2019** | MBCA | 300 m | 600 | 30 m/s | - | Highway |
| **2019** | CVoEG | 200-1000 m | 50,1214 | 10-50 m/s | - | Highway |
| **2019** | P-DACCA | 150 m | Variable | 0-42 m/s | Multi | Highway |
| **2019** | CBL | - | - | 22 m/s | Single | Highway |
| **2019** | Abbas et al. [62] | 500 m | 20 | 11-33 m/s | - | Highway |
| **2019** | Moore et al. [63] | - | 180 | 35-37 m/s | - | Highway |
| **2020** | CCA-IoV | - | 100 | 10-35 m/s | Single | Highway |
| **2020** | GOA | Dynamic | 100 | 22-30 m/s | Multi | Highway |
| **2020** | ACOS | 500 m | 20-40vh/h | 26.5-32 m/s | - | Urban |
| **2020** | DPA | 200 m | - | 20 m/s | Multi | Highway |
| **2021** | JCV | 200 m | 100 | 10-35 m/s | Single | Highway |
| **2021** | Maan et al. [69] | 300 m | 100 | 5-10 m/s | Single | Highway |
| **2021** | Khan et al. [70] | 300 m | Depends on arrival rate | 16 -27 m/s | - | Highway |
| **2021** | KMRP | 250 m | 100,150,200,250, 300 | 16-33 m/s | Single | Highway |
| **2022** | RCMS | 250 m | 1200 | 10-30 m/s | Multi | Urban |

From Table 2.3, we can see that each algorithm used different assumptions (density, mobility, traffic scenario, transmission range, and model). So, it is challenging to fairly compare these algorithms.

## 2.5 Performance Evaluation and Simulation Tools

### 2.5.1 Performance Evaluation Parameters

Any clustering algorithm's performance can be assessed and evaluated using a variety of parameters; Cluster performance and network performance are the two most common metrics used for evaluating the performance of clustering algorithms:

1. Cluster Performance Parameters represent how well clustering techniques perform and mirror how stable the network's backbone nodes are. The overall cluster performance and stability are described using these parameters. There are many cluster performance parameters; some of these parameters are [13], [73]:

- Cluster number: It refers to the number of clusters that form during network operation. The clustering algorithm is more efficient when there are few cluster numbers.

- CH Stability/CH Lifetime: It is the maximum period of time that a vehicle can spend performing the head role in a cluster. It is computed by dividing the overall lifetime by the time spent in the head’s role.

- CM lifetime: It is the maximum period of time a node may be CM. To get its average, we divide the total lifetime of the CM by the total number of state changes from CM to another state.

- CH change rate: It is the average CH’s number change per time.

- Cluster change rate: Average clusters’ number changes for each vehicle in a unit of time.

- Cluster size: Vehicles’ number in one cluster.

A good and stable clustering algorithm should have a large cluster size, high CH and CM lifetimes, few cluster numbers, and low cluster and CH change rate. However, these parameters only can't describe communication links' details between vehicles in the network.

2. Network Performance Parameters: The overall network performance is described by these parameters, which include [4], [73]:

- Throughput: It is the number of bits transmitted per second in any network. The higher value of throughput provides better performance of the network designed.

- Packet loss ratio or collision ratio: The rate of packets' loss during the transmission process

- Packet Delivery Ratio (PDR): It is the ratio of the number of packets received by the destination to the total number of packets.

- Overhead: The average number of control messages received by the vehicle.

- End to End Delay (E2E Delay), Packet Delay (PD), or Latency: It is the time taken for transmitting a packet from a source to a destination.

All these parameters are utilised to estimate the context-based clustering approaches, like traffic prediction, routing, and information dissemination. A good and efficient clustering algorithm leads to large throughput, short E2E delay, low packet loss rate, high PDR, and small overhead.

### 2.5.2 Simulation Tools

The researchers have described numerous vehicular network simulators and traffic simulators and highlighted difficulties in simulating VANETs. NS2 is used to validate the effect of safety mobility in VANET. Our summary indicates that NS2 has been one of the most popular options for researchers. Due to their low complexity, NS3 and OMNeT++ are currently being utilised more and more, whereas C++, which is thought to be unreliable, it was only employed by a few researchers for network simulation [13]. NS2 has the same feature as Simulation of Urban Mobility (SUMO). SUMO is one of VANET's traffic simulators, and it can simulate urban mobility. For this simulator, the dataset that is most commonly used in literature is Open Street Map (OSM) and Google Map. Others used a mobility model generator for vehicular networks, which is called MOVE. It is a component of the SUMO traffic simulator that integrates the JAVA platform. Its software is able to track files containing realistic vehicle movement data.

Researchers usually use modelling based on real-world observations to achieve higher levels of reliability and accuracy in simulation results [74].

MATLAB is a powerful tool for modelling system and analysing calculations. It has the ability to calculate complicated algorithms and provides quick results. MATLAB is more powerful than others in analytical. It can be presented in a variety of scheme graphs and is adaptable to any algorithm. But in another, many researchers used NS2, NS3, SUMO, etc. It depends on the skill and research goal of the researchers [75]. MATLAB also supports socket communication like OMNeT++ and helps us to develop the event-driven VANET scenario. NS2 and NS3 are good in network simulation; but unable to run in-depth analyses in an algorithm. SUMO can run in-depth analysis in the algorithm, but it needs a third party like Traffic Control Interface (TraCI) for MATLAB and uses it to make the program. TraCI is an Application Programming Interface (API) developed in MATLAB that enables communication between any MATLAB-written application and the SUMO simulator. It combines all advantages and compensates for the disadvantages of SUMO and MATLAB [76]. TraCI allows users to control SUMO objects such as vehicles, traffic lights, junctions, etc., through MATLAB, which means it is designed to get access to the traffic running on the road simulated. Table 2.4 presents a comparison between the most used simulators according to different criteria.

Table 2.4 Simulators Comparison

| **Criteria** | **NS2** | **NS3** | **OMNeT++** | **SUMO** | **MATLAB** |
|---|---|---|---|---|---|
| **Open source** | Yes | Yes | Open-source (for study and research purposes), Commercial (for industrial purposes) | Yes | Yes |
| **Language support** | C++ | C++, Phyton | C++ | C++ | C, C++, JAVA |
| **Scalability** | Yes | Yes | Yes | Yes | Yes |
| **Solve Complex Analytical** | No | No | No | No | Yes |
| **Realistic Model** | Yes | Yes | Yes | Yes | Yes |
| **Supported OS** | Linux, Unix, Windows | Linux, Unix | Linux, Unix, Windows, macOS | Linux, Windows, macOS | Linux, Unix, Windows |
| **Ease to use** | Hard | Moderate | Moderate | Hard | Easy |

Table 2.5 presents the evaluated parameters and the simulator tools used for each algorithm in the literature. Each clustering algorithm used different evaluated parameters to evaluate its performance and used different simulator tools in the implementation. The clustering algorithms in the literature are classified based on their objectives as shown in Table 2.5; Stability improvement, security improvement, scalability and reliability improvement, routing improvement, delay reduction, overhead reduction, and network performance improvement.

Table 2.5 Simulator tools and Evaluation Parameters for the Exiting Clustering Algorithms

| Ref | Algorithm | Goal | Simulator Tool | Evaluation Parameters |
|---|---|---|---|---|
| [18] | PPC | Stability improvement | NS2 | Mean cluster diameter, Cluster reconfiguration rate, and throughput. |
| [19] | RMAC | Routing improvement | NS2 | Cluster residence times, Node re-clustering time, Average error of estimatted location. |
| [20] | APROVE | Stability improvement | NS2 | CH lifetime, CM lifetime, CH change rate. |
| [21] | DBC | Stability improvement | VANET MobiSim and Java | Average cluster size, average number of clusters, average percentage of clustered nodes, average number of CH changes per node, average time which node spend being clustered. |
| [22] | ALM | Stability improvement | SUMO, SIDE/ SMURPH | CH lifetime, Status changes per Node, CH density. |
| [23] | CBDRP | Routing improvement | NS2 | Latency, PDR, Average Routing Overhead |
| [24] | VWCA | Security improvement | MATLAB | CH and CM lifetime, PDR. |
| [25] | ASPIRE | Stability improvement , overhead reduction | NS2, MOVE | CH change rate, CH lifetime, CM lifetime, cluster size, percent connectivity. |
| [26] | Zhang et at. [26] | Stability improvement | NS2 | CH lifetime, CM lifetime, CH |

| | | | | |
|---|---|---|---|---|
| | | | | change rate. |
| [27] | Sp-Cl | Stability improvement | - | Average cluster change, Number of clusters, and average cluster lifetime. |
| [28] | SBCA | Stability and network performance improvement | NS2 | Average cluster lifetime, overhead, and packet delivery. |
| [29] | TB | Stability improvement | C++ | Average cluster change, Average cluster lifetime |
| [30] | FLBA | Stability improvement | NS2, MOVE, SUMO | Average CH time, Average CM's dwell time, Average cluster size. |
| [31] | AMACAD | Stability and network performance improvement | Java | CH lifetime, Membership lifetime, Re-affiliation rate. |
| [32] | TACR | Routing improvement | - | Routing Overhead, CH Selection Time, Cluster Creation Time, and Probability of message Transmission. |
| [33] | ALCA | Security improvement | VANET MobiSim | Node participation time, Throughput, Efficiency, CH duration, Connectivity ratio |
| [34] | VmaSC | Stability improvement | NS3, SUMO | CH/CM Duration, CH Change Rate, Overhead, Number of Vehicles in SE state. |
| [35] | Arkian et al. [35] | Stability improvement | OMNeT++, SUMO | CH lifetime, cluster number |
| [36] | DMCNF | Stability improvement and overhead reduction | NS2, VanetMobiSim | Average CH/CM durations, Average number of clusters, Average CH change number, and average overhead. |
| [37] | AWCP | Stability improvement | NS2, JOSM, SUMO, | Average Cluster Lifetime, PDR, overhead. |

| | | | | |
|---|---|---|---|---|
| | | | MOVE | |
| [38] | ARV | Stability improvement | SUMO | Average Cluster-Head lifetime, Percentage of CHs. |
| [39] | VMaSC-LTE | Stability improvement | NS3, SUMO | CH/CM Duration, CH Change Rate, Overhead, Number of Vehicles in SE state. |
| [40] | MOSIC | Stability improvement | NS3 | Average CH/CM Duration, Average Number of clusters, Average Control Message Overhead, Average CH Changes Rate. |
| [41] | NCABAT | Routing improvement | JADE | Throughput, E2E Delay, and PDR. |
| [42] | CBVRP | Routing improvement | - | PDR, E2E delay, Number of cluster reconstruction, Routing cost. |
| [43] | MoDyC | Stability improvement | NS2, SUMO | CH duration,CM duration, cluster number, CH change rate, cluster efficincy, state change, number of CMs |
| [44] | KMFW | Stability improvement | NS2 | Persantage connectivity, CH duration, signal quality |
| [45] | CBLTR | Routing and Network performance improvement | MATLAB, SUMO | No. Hello Message , Avg No. of CH Changes, E2E Delay, Throughput. |
| [46] | ScalE | Stability improvement | MATLAB | CM lifetime. CH re-election, Reaffiliation |
| [47] | PMC | Stability improvement | NS2, VanetMobiSim | Average CH/CM Duration Time, Number of Average Cluster Head Changes, Clustering Overhead. |
| [48] | LRCA | Network performance and stability improvement | NS2, SUMO | CH/CM duration, CH change rate, PDR,E2E delay, overhead |
| [49] | UFC | Stability improvement | SUMO | CH and CM duration, Clustering efficiency, Number of initial CHs, |

| | | | | |
|---|---|---|---|---|
| | | | | CM disconnection rate, CM re-clustering delay, and CM re-clustering success ratio. |
| [50] | ECORA | Routing overhead reduction | MATLAB, SUMO | Control overhead message |
| [51] | NMDP-APC | Delay reduction, reliability improvement | MATLAB | Number of Iteration, number of Clusters |
| [52] | Saleem et al.[52] | Stability and reliability improvement | MATLAB | Cluster size, CH duration |
| [53] | DHC | Stability improvement | SUMO | CH/CM lifetime, Number of changed states, packet overhead, Cluster formation rate, CH Alienation. |
| [54] | EWCA | Stability improvement and overhead reduction | NS2, SUMO | Cluster stability, number of clusters, and E2E. |
| [55] | HCAR | Stability improvement | NS2, VANET MobiSim | CH lifetime, average overhead, and number of cluster |
| [56] | CBSC | Security and stability improvement | OMNeT++, SUMO | Average CH/CM Lifetime, Average Number of Re-affiliation Times, Packet Loss Rate. |
| [57] | MBCA | Stability improvement | OMNeT++, SUMO, and VIENS | Average CH duration, average CM duration, PDR, network delay, and overhead. |
| [58] | CvoEG | Scalibilty improvement | MATLAB, SUMO, MOVE. | Number of clusters, delay, reliability, throughput, PDR |
| [59] | FCMS | Security and stability improvement | - | Vehicle Remain or Leave Cluster. |
| [60] | P-DACCA | Collision avoidance | NS2 | Cluster stability, overhead, and collision probability. |
| [61] | CBL | Stability improvement | MATLAB, SUMO | No. of Relay nodes |

| | | | | |
|---|---|---|---|---|
| [62] | Abbas et al. [62] | Delay reduction, reliabiliry improvement | MATLAB, NS3, SUMO,MOVE | CDF, CH duration, cellular sum rate, throughput |
| [63] | Moore et al. [63] | Delay reduction, Scalibilty improvement | NS3 | Delays confidence interval, CCH and SCH utilization |
| [64] | GOA | Stability improvement | MATLAB | Number of clusters |
| [65] | CCA-IoV | Stability improvement | SUMO, NS2 | Avarege CH and CM duration, cluster number. |
| [66] | ACOS | Stability improvement | SUMO, NS2 | CH duration, CM duration, CH change rate. |
| [67] | DPA | Reliability improvement | MATLAB | V2V Reliability, V2I Reliability |
| [68] | JCV | Network performance and stability improvement | SUMO, CVANETSIM, JAVA | CH duration, CM duration, CH change rate, number of cluster, cluster participation, number of CM. number of EN, ratio of CM, EN duration, overhead, delay. |
| [69] | Maan et al. [69] | Stability improvement | MATLAB, SUMO | CH duration, average cluster size. |
| [70] | Khan et al. [70] | Stability and relaibility improvement | MATLAB, SUMO, MOVE. | Number of clusters, link connectivity, packet loss |
| [71] | KMRP | Stability and routing improvement | NS2 | Throughput, delay, PDR |
| [72] | RCMS | Network performance and stability improvement | OMNeT++, SUMO | Cluster lifetime, PDR, delay, overlap rate, reconstruction time |

## 2.6 Literature Problems

Some of the most important issues in this literature can be concluded as follows:

1. The high dynamics of the VANET environment make the formation and maintenance of stable clusters a big challenge issue. So, many approaches focused on the CH selection and gave less attention to the other phases. An efficient cluster formation, along with the annotated CH vehicle, is responsible for improving the network performance.
2. VANET is a continuously changing topology, which creates challenges in establishing a connection from one source to the destination vehicle. If the connection is unreliable, then the data loss may be high as the carrying vehicle (CH) may change its direction and speed. Accordingly, a stable cluster and a reliable CH reduce or limit information loss. Sustaining a CH for a long period is difficult.
3. Mobility and neighbourhood are the most metrics taken, and these metrics are lost in the urban scenario as the vehicle speed is low and there is huge congestion in peak hours. Thus, to successfully analyse the networks, it is necessary to look for new metrics or cumulative metrics.
4. The work on the real scenario is mostly limited to the highway; the urban scenario analysis is very limited and few (9 only out of 55). Designing a clustering algorithm for an urban environment is more complicated than designing it for a highway, due to a large number of intersections and the varied speed of vehicles as a result of congestion.
5. Some of the methods designed in the literature have overlooked the effect of dynamic change in these networks.
6. Researchers have made clusters and selected the CH by calculating the vehicles' behaviour in the network, velocities, moving directions, and positions in lanes. Fuzzy logic has been used, such as in [30], [52], [59], and [69], and other used heuristic algorithms, such as in [55] and [70]. Fuzzy logic schemes require tuned membership functions to decide for CH selection, which necessitate considerable experience and behaviour analysis of vehicles on a particular road. Owing to fast-changing topology and distributed VANET architecture, heuristic algorithms hardly make decisions due to several iterations in the calculation. They cannot cope with the changing frequency of a

VANET environment in a crowded city. However, we assume that in highways with low vehicle density, they may present good performance.

## 2.7 Conclusion

The primary focus of this chapter is to conduct an intensive survey of 55 clustering algorithms that have been observed and studied from 2008 to 2022. These algorithms have been summarised and classified in terms of clustering procedures: cluster formation according to hop count, CH selection, and cluster maintenance. Also, we have made comparisons between these algorithms based on transmission range, density, velocity, hop count, and traffic scenario. Finally, some of the most common metrics used for evaluating the performance of clustering algorithms and simulator tools have been presented. Through the study, it was noted that the performance evaluation metrics and simulation tools for each algorithm are very different, so it is very difficult to know which algorithm or method gives the most efficient performance.

With all the gaps in the literature studied in Section 2.6, our primary goal in the next chapter is to design a complete solution for VANET, especially for the urban scenario. Our first proposed scheme is presented; it is called the HyperGraph Clustering Model (HGCM). The designed approach has cluster formation, CH selection, and maintenance. The evolving nature of VANETs is meritoriously captured using the concept of hypergraphs, and the clusters are formed through the designed vehicular-hypergraph-based spectral clustering algorithm. The CH is selected considering the criteria for maintaining a stable connection with the maximum number of neighbours. Different metrics (relative speed, neighbouring degree, eccentricity, and trust score) are introduced for selecting the CH and using the cumulative multimeric scheme to increase the clustering stability. The designed approach is tested on a real map of Baghdad city with the help of OSM and SUMO to generate realistic traffic. SUMO supports the TraCI API for MATLAB interfacing. The network is triggered from MATLAB through the TraCI API and data is recorded at every simulation second.

***List of publications:***

1. "Clustering Review in Vehicular Ad hoc Networks: Algorithms, Comparisons, Challenges and Solutions.," *Int. J. Interact. Mob. Technol.*, vol. 16, no. 10, 2022.
2. "Survey on Clustering in VANET Networks," in *2021 18th International Multi-Conference on Systems, Signals & Devices (SSD)*, 2021, pp. 493–502.
3. "A Review on Clustering in VANET: Algorithms, Phases, and Comparisons." in *2022 19th International Multi-Conference on Systems, Signals & Devices (SSD)*, 2022.

# Chapter 3

# HyperGraph Clustering Model (HGCM) in VANET

## 3.1 Introduction

In mathematics, graphs are used to represent pairwise relationships between things; graph theory is the study of graph. In graph context, a network consists of vertices, also known as nodes, connected by edges, also known as links. An edge in a graph connects two vertices precisely. Graph theory is utilized to solve problems with a reasonably natural graph or network structure, such as those involving road network, machine learning, and communication networks. A network of vehicles is presented as a graph, in which a vehicular node is connected with two other vehicular nodes [58],[70], and [77]. This graphical representation may be suitable in sparse density, such as highways or minimally populated cities. On the contrary, in dense urban scenarios, a vehicle is always connected with more than two vehicles, and graph theory does not fit there. Hypergraphs are a suitable representation of dense vehicle networks. When a graph's edges (hyperedges) are permitted to connect more than two vertices, the resulting structure is known as a hypergraph. In other words, a hyperedge can link a group of vertices together because they are similar to one another in some characteristics.

### 3.1.1 Clustering Methods based on Hypergraph

In a graph and hypergraph, a cluster is named community, vehicle clustering can be categorized as a community detection problem, and this problem concentrates on grouping similar nodes together using the provided node attributes. Hierarchical clustering algorithms and partitioned clustering algorithms are the two main categories of hypergraph clustering techniques. On the basis of the existing cluster structure, hierarchical algorithms construct cluster. These algorithms build partitions sequentially, whereas partitioning methods only use one partition to separate nodes into clusters [14].

1. **Hierarchical Algorithms:** it is an effective technique for analyzing networks' general structure. It uses a hierarchical tree called a dendrogram to show the connections between the nodes in a graph. One major benefit of hierarchical clustering is that it partitions without regard to the number of clusters in a graph [14].

There are two methods for hierarchical clustering: agglomerative algorithms and divisive algorithms. Smaller clusters are iteratively combined into larger ones via agglomerative algorithms, which work "from the bottom up" and "from the top down" divisive algorithms divide large clusters into smaller ones [78].

The drawbacks of hierarchical algorithms;

1. To find the best partition among the sequences of partitions, we need to propose an extra criterion.
2. Hierarchical clustering does not scale effectively, which is another issue.

**2. Partitioning Algorithms:** it is the most basic and simplest type of cluster analysis, which divides the items' set into a variety of separate groups or clusters. The most popular partitioning techniques are:

***A. K- means clustering method***

James Macqueen first presented the K-means algorithm in 1967, yet it remains one of the most often used unsupervised clustering algorithms due to its simplicity, ease of implementation efficiency, and empirical success. A set of data is divided into K number of clusters by K-means based on their characteristics [78], Figure 3.1.

Naturally, the K-means clustering method takes substantially less time and produces results similar to those of hierarchical clustering. K-means clustering has this benefit over other methods in large-scale networks. Due to its simple implementation and fast convergence, K-means clustering is popular.

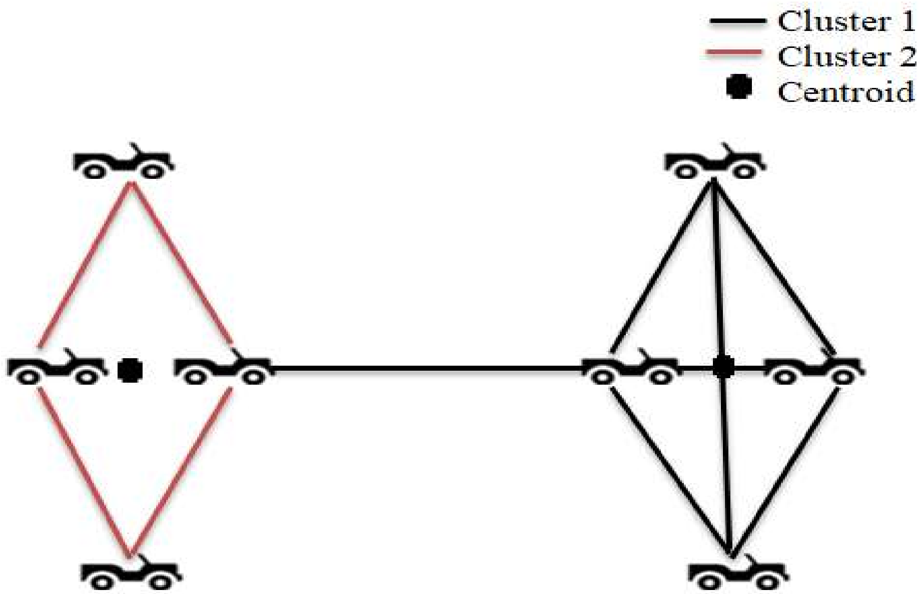



Figure 3.1 K-Means Clustering

However, there are drawbacks preventing K-means clustering from being used in dynamic and large-scale networks. First of all, the initial set of centers has a significant impact on K-means clustering. Varied initial centers produce different clustering outcomes, and as a result, different iterations are needed to reach convergence. The most common method is a random selection. It randomly chooses the initial cluster centers. Although this method is simple to implement, the outcome varies with different initial centers. The effectiveness of this algorithm cannot be guaranteed [79].

Secondly, how to determine the value of K is another issue. The number of the clusters is variable due to node mobility, particularly in a dynamic network.

Another drawback of this approach, it is not capable of disconnecting the non-linearly distinguishable clusters. To resolve this, spectral clustering is used based on eigenvectors of the matrix, by making an attempt to minimize the cut.

***B. K-Medoids Clustering Method***

Another partitioning clustering method is a K-medoids. *N* nodes are partitioned with a preassigned number; K in both K-means clustering and K-medoids clustering. K-means clustering technique and simple K-medoids are very similar. It appears that there may not be a need to study simple K-medoids clustering when we already have traditional K-means clustering [80].

In contrast to K-means clustering, K-medoids clustering is more resistant to outliers. The centroid of the cluster is what K-means refers to as the center. It indicates that the cluster's center may not be a real node. According to K-medoids clustering, a representative node known as a medoid should be at the center of each cluster, Figure 3.2. In other words, we might say that the cluster center should be a node that already exists and can be thought of as the middle of all the nodes in the same cluster [14].

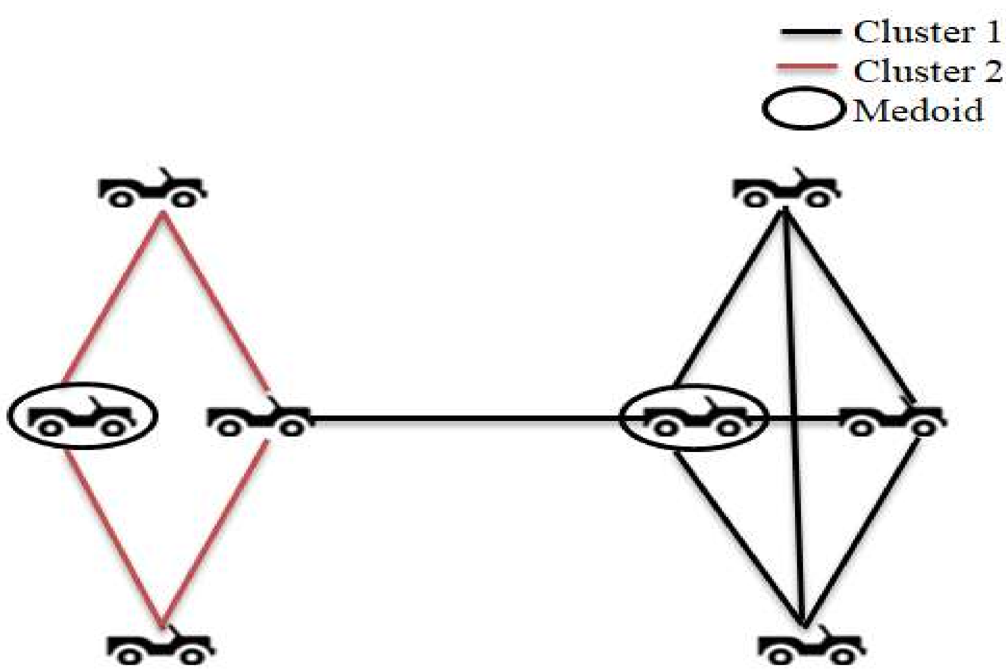



Figure 3.2 K-Medoids Clustering

***C. Spectral Clustering***

From graph theory, spectral clustering is derived. For spectral clustering, the Laplacian matrix ($L$) is a major tool. The original nodes are converted into a metric space using eigenvectors of $L$ as its coordinates. Combining with K-means clustering in the final step, spectral clustering can be explained as K-means clustering in a specific metric space [81].
The major drawback of spectral clustering limits the employment in large scale networks.

Spectral clustering has recently gained popularity as one of the most often used modern clustering techniques. It performs frequently better than conventional clustering methods like the K-means algorithm and is easy to implement. It can also be solved effectively by typical linear algebra software [78].

Partitioning clustering involves the formation of new clusters by merging or splitting the clusters instead of following a hierarchical order. Also, it is comparatively more reliable than hierarchical clustering.
In this chapter, a newly vehicular-hypergraph-based spectral clustering model using Tensor Trace Maximisation method (TTM) is introduced.

### 3.1.2 Why is VANET a Hypergraph Network?

As discussed in Chapter 2, To implement a robust clustering algorithm, there are problems need to be solved: a clustering methods need to be designed and metric has to be used to select the CH. Vehicle cluster generation is a populated concept, and various scholars have already addressed it. In our work, the clusters are generated using the hypergraph based spectral clustering method. The following are the key reasons to represent VANET as a hypergraph:

- VANET is a cooperative network where every decision depends on the information shared by neighbouring vehicles [82].
- In the graphical representation, a loss of information occurs in paired connections. Pairwise graph models don't have the representational capacity needed for tasks like analysis and learning to properly capture and display higher-order information. Hypergraphs, which depict interacting elements as nodes and hyperedges, are useful for capturing higher-order interactions in these systems.

- All vehicles in the network act as either sources, destinations or routers, depending on where they are in the network's hierarchy. One of the primary responsibilities of these nodes is to disseminate the data throughout the network. Vehicular mobility and the communication link amongst vehicles that is constantly breaking and reconnecting cause such networks to grow in nature. The relationships amongst nodes in more diversified vehicular networks are more difficult to understand because of the networks' ever-expanding character. As a result, complex networks now utilise super networks. Hypergraph-based and network-based are super networks that exist in the literature [82]. Hypergraphs pay more attention to the dynamic evolution process, making it possible to conduct a dynamic analysis of complicated networks.
- The hypergraph theory has the benefit of ensuring the homogeneity of points and edges. This makes it easier to express how nodes and edges relate to one another. Consequently, a hypergraph, in which one vehicle can communicate with numerous vehicles, can be used to modulate the representation of networks.

### 3.1.3 Proposed Contributions

A novel approach for clustering formation and maintenance of a VANET structure in an urban scenario is introduced in this chapter. It is called the hypergraph clustering model (HGCM).The CH stability is governed by a cumulative multimetric factor inclusive of relative speed, eccentricity, neighbourhood and spectrum sensing based on cooperative trust. The contributions of this proposed are listed below:

- A formulation of VANET through a hypergraph is introduced. The construction of the hypergraph is designed using the distance proximity amongst the vehicles in the network.
- Practical and optimal partitioning of the hypergraph through tensor trace maximisation (TTM) is proposed. A high order has all the edges but with negligible weights. Thus, the adjacency matrix is nearly sparse, and the overall computational complexity is effectively reduced.
- Optimal clusters are selected in accordance with the Calinski–Harabasz concept. This method is an external criterion for selecting optimal clusters. Hence, the information is

independent, and the structure of the information is inherited. Such a method is also preferred for convex clustering.

- The network's performance, especially in an urban scene, can be improved by installing auxiliary facilities, such as RSUs. Here, an evolving graph structure of the traffic is conceived using betweenness centrality.
- The challenge of classification is used to reinvent spectrum sensing. The method proposed for sensing is long short-term memory (LSTM), which is extensively trained for all signal types, including noise. Thus, it can sense an untrained signal and classify a vehicle as primary or secondary.
- The scheme of a cumulative multimetric for selecting a CH is presented, through which strong connectivity and stable link lifetime are maintained. Extensive simulation and comparison of cumulative CH selection scheme with individual measures are presented to show its supremacy in terms of stability. The stability of the CH enhances the routing performance of the designed approach in terms of PD and throughput.

## 3.2 Proposed Model

A multilane road structure in an urban scenario is considered. Fluctuating density of building infrastructure and vehicular mobility with total number of vehicles $N$ are infused into the real map scenario with corresponding speed and locations. Each OBU equipment has the same transmission range $R_{vehi}$. The same communication module is taken to communicate with auxiliary facilities, each having a transmission range $R_{RSU}$. Communication amongst vehicles is carried out using the V2V protocol. GPS unit and IEEE 802.11p radio equipment are embedded inside the OBU. By contrast, V2I is used for communication between vehicles and RSU. Every car in the network functions as a node (V), acting as a source, destination, or router.

The main task of these nodes is to broadcast information within the network. The vehicles are said to be one-hop neighbours if the distance between them is less than or equal $\boldsymbol{R_{vehi}}$ and multihop if the distance between them is greater than $\boldsymbol{R_{vehi}}$. In this section, the formulation for designing VANET as a hypergraph-partitioning problem is discussed in detail. The complete work can be divided into four major steps:

1. Neighbouring vehicle identification and adjacency matrix generation

2. Hypergraph-based spectral clustering algorithm for cluster formation

3. RSU deployment and cluster members allotment

4. Stable CH selection

The flow diagram of the complete work is shown in Figure 3.3.

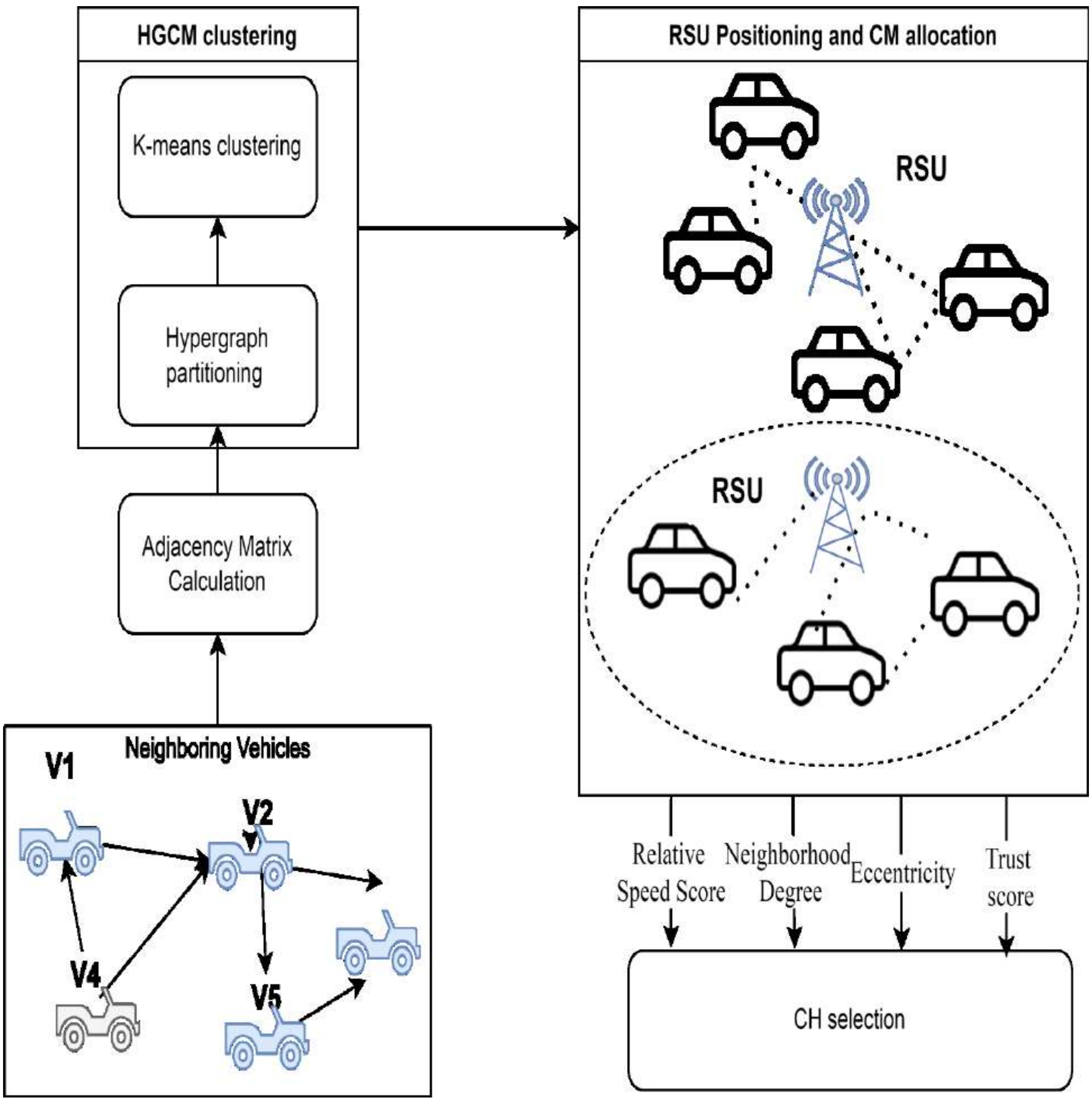



Figure 3.3 Flow diagram of the proposed HGCM Scheme

The pseudo code of the suggested work is jotted down in Algorithm 3.1.

**Algorithm 3.1: Pseudo code of the proposed scheme for stable CH detection**

**Input:** Vehicle-to-vehicle transmission range ($\boldsymbol{R_{vehi}}$), RSU-to-vehicle transmission range ($\boldsymbol{R_{RSU}}$).

1. For i=1:vehicles
    a. All neighbouring vehicles within the vehicle transmission range are identified
2. End for
3. Determine the similarity matrix based on how close the vehicles are to one another in terms of distance
4. Generate an adjacency matrix
5. Create Eigen values for each vehicle by using hypergraph-based TTM
6. Use Calinski-Harabasz and k-mean clustering to determine the ideal number of clusters.
7. The centrality index gives the RSU deployment location
8. Simulate the VANET for different vehicle densities in the network and collect the vehicle moving angle, locations and other information until every vehicle leaves the network.
9. For i=1: clusters
    a. For j=1: vehicles in cluster
        i. Calculate the CH parameters: neighbourhood degree, relative speed, trust score and eccentricity
        ii. Combine these matrices to obtain a single score for each vehicle
        iii. Highest-scorer vehicle is termed as the CH
    b. End for
10. For i= 1: vehicles
    i. Selected CH sends frequent polling signals
    ii. If ( the distance between the CH and vehicle <= CH transmission range) then
        a. The CH receive signal in return within a stipulated time period
        *b. CH Adds the vehicle in the RSU List & CH Local List*
        Else
    a. The vehicle does not reply within the speculated period time_span
    b. This vehicle is considered to be disconnected and leave the cluster

c. *CH Removes the vehicle from the RSU List & CH Local list*

End if

End for

11. End for

### 3.2.1 Formulation as hypergraph partitioning

A connected weighted hypergraph is a three-tuple of $\mathcal{H} = (\boldsymbol{V}, \boldsymbol{\mathcal{E}}, \boldsymbol{W})$, where each edge $\boldsymbol{\mathcal{E}}$ links a subset of vertices $\boldsymbol{V}$ in the hypergraph and may be linked with non-negative weight. This structure is composed of n vertices, $\boldsymbol{V} = \{\boldsymbol{v_i} | \boldsymbol{i} = 1,2,\ldots \text{n}\}$, where each vertex is a vehicle in our study, $\boldsymbol{\mathcal{E}} = \boldsymbol{c_{ij}} : \{< \boldsymbol{v_i}, \boldsymbol{v_j} > | \boldsymbol{v_i} \in \boldsymbol{V} \wedge \boldsymbol{v_j} \in \boldsymbol{V} \wedge (|\boldsymbol{d_{ij}}| \leq \boldsymbol{R_{vehi}})\}$ is an edge set, and $\boldsymbol{W} \to [\boldsymbol{0}, \boldsymbol{1}]$ is the weight associated with each edge. The distance of each vehicle constitutes the edge of each hyperconnection. The hyperedge connection is defined in Definition 1.

***Definition 1:*** Two vehicular nodes $v_i$ and $v_j$ at time $t$ are said to be connected if

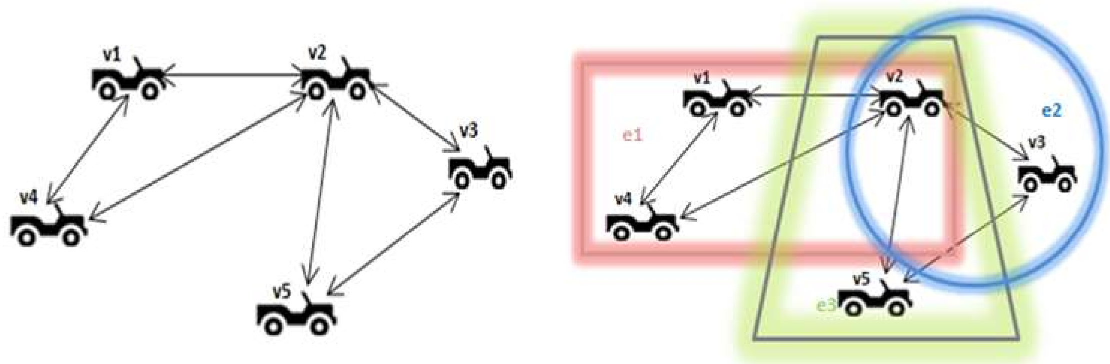
$$c_{ij} = \begin{cases} 1 & d_{ij} \leq R_{vehi} \\ 0 & d_{ij} > R_{vehi} \end{cases} \tag{3.1}$$

The connection between two vehicles is established using the distance proximity formulation. Here, the connection is established if the distance between the two vehicles $d_{ij}$ is less than or equal the transmission range of the vehicle nodes.

### 3.2.2 HGCM Generation Model

The cluster generation part of VANETs is discussed in this section. An urban scenario is considered for the simulation, and location information is shared with every neighbouring vehicle in the test case. A network hypergraph is constructed using that location information, and this section discusses the formulation of the vehicle hypergraph.

Our proposed model aims to cluster vehicles so that minimum bandwidth occupancy at any instant is achieved [83]. In the urban scenario, road congestion is unavoidable, leading to slow-moving traffic. The location, speed and vehicles in an area affect the stability of the clustering [84]. Each vehicle in a cluster is categorised as either CH or CM. Only one CH is conventionally allowed in a cluster, except in special cases of warlike fields [85]. In this proposed, the maximum possible vehicle density $N$ is considered for the clustering algorithm. Using the location information, each vehicle finds its neighbour. A similarity matrix is generated, whose cell elements indicate the connection strength with another corresponding vehicle, as shown in Figure 3.4. The adjacency matrix showcases the relation amongst the vehicle nodes, and the incidence matrix showcases the relation between the vehicle nodes and the edges formed.




**Incidence Matrix**

| | e1 | e2 | e3 |
|---|---|---|---|
| V1 | 1 | 0 | 0 |
| V2 | 1 | 1 | 1 |
| V3 | 0 | 1 | 0 |
| V4 | 1 | 0 | 0 |
| V5 | 0 | 0 | 1 |

**Adjacency Matrix**

| | V1 | V2 | V3 | V4 | V5 |
|---|---|---|---|---|---|
| V1 | 0 | 1 | 0 | 1 | 0 |
| V2 | 1 | 0 | 1 | 1 | 1 |
| V3 | 0 | 1 | 0 | 0 | 1 |
| V4 | 1 | 1 | 0 | 0 | 0 |
| V5 | 0 | 1 | 1 | 0 | 0 |

Figure 3.4 Hypergraph Based VANET Analysis

In the process of spectral clustering, information loss is evident. To prevent it, the TTM clustering has been proposed to use for spectral clustering of vehicles. Spectral clustering using TTM is introduced in this subsection. Setting up k disjoint sets $\boldsymbol{V_1}, \dots \dots \boldsymbol{V_k}$ from the weighted hypergraph $\boldsymbol{V}$ is problematic, such that the partitions are balanced and the overall weight of the edges inside each cluster is high (vehicle connectivity that is dense) [86]. The degree of every node is the quantity that indicates how many vehicular nodes are connected to it, $\boldsymbol{v} \in \boldsymbol{V}$, determining the total weight of incident edges $\boldsymbol{v}$, i.e. $\boldsymbol{deg}(\boldsymbol{v}) = \sum_{\boldsymbol{e} \in \mathcal{E}: \boldsymbol{v} \in \boldsymbol{e}} \boldsymbol{w_e}$.

The volume is then described as $\boldsymbol{CM(V_1)} = \sum_{v \epsilon V_1} \boldsymbol{deg(v)}$, which is the quantity of nodes that have occurred on node $\boldsymbol{V_1}$, such that $\boldsymbol{V_1} \subseteq \boldsymbol{V}$. The following is the definition of the association between the edges in $\boldsymbol{V_1}$: $\boldsymbol{assoc(V_1)} = \sum \boldsymbol{w_e}$. These distinct partitions' normalised associativity is provided as

$$\boldsymbol{Par}\ (\boldsymbol{V_1}, \dots \dots \boldsymbol{V_k}) = \sum_{i=1}^{k} \frac{\boldsymbol{assoc(V_i)}}{\boldsymbol{CM(V_i)}} \tag{3.2}$$

The tensor (order $m$) is the adjacency matrix that is defined here,

$$A_{i_1,i_2,\dots\dots i_m} = \begin{cases} w_{\{i_1,i_2\dots\dots i_m\}} & if\ i_1,i_2,\dots\dots,i_m\ are\ distinct \\ 0 & otherwise \end{cases} \tag{3.3}$$

It is possible to rewrite the normalised associativity in terms of $A$ and $Y$. (inverse of the number of vehicles connected to a node) as

$$Par_{i\in\{1,..k\}} = \frac{1}{m!} Trace(A \times_1 Y^{(1)^T} \times_2 Y^{(2)^T} \times_3 Y^{(3)^T} \dots \dots . \times_m Y^{(m)^T}) \tag{3.4}$$

where $\times_l$ is the model-l product.

$Y^{(1)^T}, Y^{(2)^T}, Y^{(3)^T} \dots \dots ., Y^{(m)^T} \in \mathbb{R}^{k\times m}$, This indicates how many CMs are connected to each vertex's node $v_i$. $\mathrm{Y}^{\mathrm{i}\in\{1,2..\mathrm{m}\}}$, as shown as follows:

$$Y^i = \frac{1}{\sum_k CM(V_i)} \tag{3.5}$$

Spectral clustering takes into account the defined adjacency matrix [87], and the diagonal matrix (degree matrix) $Dig$ is obtained by transversely orienting this hypergraph that is the total of all vehicle node runs that are one-hop next to node $v_i$.

$$Dig_{ii} = \sum_{j=1}^{N} A_{ij} \tag{3.6}$$

For the Laplacian graph computation, this study utilises the unnormalised Laplacian matrix based on the Fiedler vector defined as

$$L = Dig^{-1/2} A Dig^{-1/2} \tag{3.7}$$

The top $k$ eigenvector $(U = eig(L))$ is taken for k-means clustering that provides $k$–partitions of the VANET hypergraph structure. These divisions are similar to how the vehicular network's cluster formation appears. They are further pruned to get the ideal collection of clusters for maintaining the VANET. The pseudocode for the spectral clustering is listed in Algorithm 3.2.

***Definition 2:*** The weighted hypergraph $\mathcal{H} = (\mathrm{V}, \mathcal{E}, \mathrm{W})$, and its cluster is a tuple of $(C_{num}, C_{optimal})$, where

$C_{num} = \{c_i \mid i = 1, 2, \ldots\ldots k\}$ is a cluster set, where $k$ is the total number of clusters. $\forall c_i \in C_{num}, c_i = \{v_j | \tau(v_j) = 0 \vee (\tau(v_j) = 1 \wedge \exists v_k \in N(v_j) \wedge \tau(v_k) = 0)\}$.

$C_{optimal}$ represents the optimal set of clusters, $C_{optimal} = [C_{num}: \forall\ max(s\ )]$.

The clustering efficiency can be evaluated using the Calinski–Harabasz index $(s)$ [88]. This index checks the closeness of vehicles in a cluster and the dispersion of all clusters by using Equation (3.8). The maximum value of s is the desired efficiency in the clustering.

$$s = \frac{tr(B_k)}{tr(Z_k)} \times \frac{Vehi_{num} - k}{k-1} \tag{3.8}$$

Here, $k$ represents the clusters' number, and each has the size of $Vehi_{num}$. $tr(B_k)$ is the dispersion amongst clusters, and $tr(z_k)$ is the dispersion amongst vehicles in a cluster. These two terms are calculated in Equations (3.9) and (3.10).

$$Z_k = \sum_{q=1}^{k} \sum_{x \in C_q} (x - c_q)(x - c_q)^T \tag{3.9}$$

$$B_k = \sum_{q=1}^{k} n_q (c_q - c_E)(c_q - c_E)^T \tag{3.10}$$

Here, $c_q$ is a set of points in cluster $q$, and $x$ is specifically the centre of the cluster. $c_E$ is the centre of clusters with $n_q$ points in them.

The maximum value of $s$ is used to select an ideal group of clusters using this index from the pool of generated clusters.

| **Algorithm 3.2: Cluster Formation using Hypergraph Theory** |
|---|
| ***Input***: Maximum number of vehicles: $N$, Each vehicle's location: $\mathrm{Vehi}_{\mathrm{Loc}}$<br>1. Choose the time t when there are the most vehicles $N$<br>2. Form a hypergraph $H = (V, \mathcal{E}, W)$<br>3. Determine the similarity matrix $A$ based on how close the vehicles are to one another in terms of distance<br>A square matrix of size $N \times N$ of similarity (adjacency matrix) as $A = \sum_{i_3,\ldots\ldots,i_m=1}^{N} A_{ij}$<br>4. A diagonal matrix $Dig \in \mathbb{R}^{N\times N}$ is with $Dig_{ii} = \sum_{j=1}^{N} A_{ij}$, and $L = Dig^{-1/2} A Dig^{-1/2}$<br>5. Then, $k$ dominant eigenvector of $L$ is computed as $U \in \mathbb{R}^{N\times k}$<br>6. Normalise each row of $\bar{U} = U$<br>7. Run k-means on the rows of $\bar{U}$<br>8. Obtain $C_{num}$ through k-means partition $Par = \{V_1, \ldots \ldots V_k\}$<br>9. For each cluster $C_{num}$, calculate the Calinski–Harabasz ($s$) criterion<br>10. Find the optimal cluster $C_{optimal} = [C_{num}: \forall \max(s)]$ |
| ***Output***: The optimal set of clusters: $C_{optimal}$ |

### 3.2.3 RSUs deployment

The RSU is an integral part of the VANET. The VANET is a hierarchical architecture consisting of the main server, RSUs, and vehicles. RSU collects the data from the moving vehicles. The clustering of vehicles has been discussed in the context of RSU by many researchers as if the RSU has request congestion, then packet drop will increase. Therefore, the optimal number of RSUs has to be calculated so that maximum probable vehicles can be served without congestion. The optimal number of clusters has been calculated in the previous section. The RSU is placed at the centroid of the initial clusters using a hypergraph assuming the maximum vehicle density. This way, a minimum number of RSUs can cover the maximum number of vehicles in the area. The installation cost would also be lesser (this is not evaluated in the simulation). Algorithm 3.3 tabulates the steps in locating the centroid for the RSUs and their deployment. The vehicles in any cluster cannot be controlled due to the nature of clustering. Consequently, few vehicles, such as three, can also lie in that precalculated cluster area. In such a case, the RSU is placed using a Gaussian probability distribution[89] as:

$$Vehi \sim N(\mu, \sigma^2) \tag{3.11}$$

The distribution is defined as the mean ($\mu = 0$) and variance $\sigma = 1$.

$$f(Vehi) = \frac{1}{\sigma\sqrt{2\pi}} \; e^{-(Vehi-\mu)^2} / 2\sigma^2 \tag{3.12}$$

A network graph $G = (V, \mathcal{E})$ amongst V vehicles' connections with $\varepsilon$ set of edges. The centrality matrix for a graph is the measure of its compactness [90]. The centrality determines the most visited vertex in a graph. For a vehicle $v$, it can be calculated as

$$C_B(v) = \sum_{s \neq v \neq u \in V} \frac{\sigma_{st}(v)}{\sigma_{st}} \tag{3.13}$$

Here, $\sigma_{st}$ is the total number of the shortest paths from node $s$ to node $u$, and $\sigma_{st}(v)$ is the number of paths that pass through $v$. The vehicle with maximum centrality value is considered the cluster's centre as in (line 10). This is the location where RSU is to be installed.

**Algorithm 3.3: RSU Deployment**

**Input**: Set of optimal clusters $C_{optimal}$ and number of vehicles in each cluster $Vehi_{num}$.

***1.*****For** i=1: $C_{optimal}$
**2.** **If** $Vehi_{num} \leq 3$
3. Select a vehicle based $Vehi \sim N(0,1)$ shown in Equation (3.12)
4. $RSU_{Loc} = \mathrm{Vehi_{Loc}}$
5. **Else**
6. Construct an urban road map with a graph $G = (V, \mathcal{E})$
7. Edge is $e_{ij} \epsilon\ \mathcal{E} = d_{ij}$ based on the distance amongst vehicles
8. Obtain a connection matrix

$$c_{ij} = \begin{cases} 1 & if\ d_{ij} \leq R_{vehi} \\ 0 & otherwise \end{cases}$$

9. Evaluate the betweenness centrality $C_B$ by using Equation (3.13)
10. $Vehi = \max(C_B)$
11. $RSU_{Loc} = \mathrm{Vehi_{Loc}}$
12. **End**
13. **End**

**Output:** RSU location $RSU_{Loc}$.

## 3.3 CH Selection and Cluster Maintenance for HGCM

### 3.3.1 CH Selection

The vehicle node $v_i$ in the network at time $t$ has features $f_i(t) = \{\vec{s}, \vec{p}, a, \theta, Vehi_{ID}, \eta\}$, where $\vec{s}$ is the vehicle speed, $\vec{p}$ is the location of each vehicle in both coordinates $(x, y)$, $a$ is the acceleration, and $\theta$ is the vehicle direction. Each vehicle is assigned a unique identity $Vehi_{ID}$, and $\eta$ refers to the one-hop neighbours of vehicle node $v_i$. Out of these nodes, a vehicle is selected as the CH. In this proposed, the CH selection metric $m_i(t)$ is a collection of metrics $\{\psi_{vehi}, \eta, \mathfrak{E}, \mathcal{t}\}$. The CH selection process is dependent upon the current CH selection metrics of each CM in the cluster.

$\psi_{vehi}$ is the relative speed, $\eta$ is the set of neighbours of vehicle $v_i$, $\mathfrak{E}$ is the eccentricity, and $\mathcal{t}$ is the trust calculated via spectrum sensing. The selected CH should have a maximum of $\sum_{i=1,2..n} m_i(t)$ at any instant $t$. Given that the hypergraph network is the cooperative network, each vehicle's feature is relative to every hyperedge linked to that hypernode [91].

The novel contribution in CH selection parameters is the strength of the cooperative nature of the hypergraph. The use of deep learning in the calculation of the trust score of each vehicle is another novel contribution to the CH selection.

All of the network's information is collected by the CH, sends it to the RSU and maintains the communication between the cluster vehicles and RSU. The stability of the CH will be higher if it will be in a communication link with the neighbour vehicles for a longer time.

The definitions of each metric with the essential background are presented below.

#### 1) Relative Speed Score ( $\psi_{vehi}$)

***Definition 3:*** A vehicular node's $v_j$ relative speed score $\psi_{vehi}$ is a score that either penalises or gives reward to a vehicle if it crosses a cluster's average speed or aligns with the cluster. A high score of $\psi_{vehi}$ indicates a high probability of election.

This parameter determines how close a vehicle's speed is to its neighbour's. The relative speed of each vehicle is calculated by differentiating its speed from the cluster's average speed at any instant of time. The moving direction of the vehicles also comes into play this way. The more vehicles are moving in the same direction, the higher $\psi_{vehi}$ will be. The relative speed score is evaluated as shown in Equation (3.14) [35]. The relative speed is compared with a threshold speed $S_{thr}$. If a vehicle is moving at higher speed than $S_{thr}$, its $\psi_{vehi}$ gets penalised with δ; else, a reward of δ is added to its score.

$$\begin{cases} \psi_{vehi}(t+1) = \psi_{vehi}(t) + \delta \;\; ; \left|V_{vehi} - V_{avg}\right| \leq S_{thr} \\ \psi_{vehi}(t+1) = \psi_{vehi}(t) - \delta \;\; ; \left|V_{vehi} - V_{avg}\right| > S_{thr} \end{cases} \tag{3.14}$$

$\delta$ and $S_{thr}$ are 0.01 and 2.77 for this work, respectively.

### 2) Neighbourhood Degree ($\eta$)

***Definition 4:*** The connection status between the two vehicular nodes $v_i$ and $v_j$ at time $t$ in the cluster formed $C_{optimal}$ with vehicle density $Vehi_{num}$ is defined as

$$\eta = \sum_{j=1}^{Vehi_{num}} c_{ij} \, ; \forall \, 1 \tag{3.15}$$

High $\eta$ ensures that the CH will not be dynamic for a long time. The degree of neighbourhood determines the number of vehicles that are generally present in the area. The vehicles under the transmission range of OBU are considered neighbours. $c_{ij}$ is 1 if the distance between two vehicles at the time stamp $t$ is less than or equal $R_{vehi}$ [92]. The transmission range and distance have an inverse relationship. That is, if two vehicles are close to each other, then a more reliable connection is bound.

### 3) Eccentricity ($\mathfrak{E}$)

***Definition 5:*** Let $A$ be a fundamental matrix. Then, an eigenvector $U > 0$ exists, such that $AU = \lambda_1 U, \lambda_1 > 0$ is an eigenvalue of an immense magnitude of $A$, the eigenspace associated with $\lambda_1$ is one-dimensional, and $U$ is the only non-negative eigenvector of $A$. $\mathfrak{E}$ is the average of top $k$ eigenvalues of $A$ designed for $\mathcal{H} = (\mathrm{V}, \mathcal{E}, \mathrm{W})$.

Due to the rapid movement of vehicles, communication links frequently fail in real-time. A requirement is placed for a progressive cluster model in order to maintain a link. Once the CH resigns or no longer meets the criteria to remain a CH, reclustering will typically become inevitable. The idea of eccentricity ($\mathfrak{E}$) is introduced in order to provide stability. In this instance, spectral clustering is used to create an updated graph-based model [58]. A vehicular graph topology is intended to be hypergraph $\mathcal{H} = (V, \mathcal{E}, W)$ with the usual procedure as defined in Section 3.2. The adjacency matrix $A$ is generated on the basis of the distance proximity amongst the vehicles present at each time instant $t$ for each cluster. The eigenvalues for a vehicle $i$ in each group are $\lambda_i$, where $i = \{1,2, \ldots \ldots Vehi_{num}\}$. Lastly, $\mathfrak{E}$ is the mean/average eigenscore of each vehicle calculated as [58]

$$\mathfrak{E} = \frac{1}{|Vehi_{num}|} \sum_{\lambda_i \in Vehi_{num}} \lambda_i \tag{3.16}$$

The maximum value of $\mathfrak{E}$ ensures a stable CH selection designed in accordance with hypergraph theory.

### 4) Trust Score $(t)$

***Definition 6:*** Through channel $h$, the signal is received from the user, and the probability of detection is 1. Then, the user is primary (PU); else, it is a secondary user (SU).

There might be some VIP and emergency cars on the vehicular network, which are regarded as the PUs of the communication spectrum in the network. Others are assigned as SUs. Every vehicle takes part in spectrum sensing. Given that the communication spectrum is limited, the cognitive spectrum sensing approach is used in the communication model [93]. Once the PU is detected in the network, the SU will have to vacate the spectrum for it. The SU following this protocol gains the trust, and the trust score $t$ is increased. The model of cognitive spectrum sensing is introduced in this work to elect the most trustworthy vehicle to serve as the CH.

By comparing the signal strength of nearby cars with the probabilistic threshold value, the SU detects the existence of the PU. The test statistic for the detected energy signal might be given in a complicated form as

$$T(Y) = \frac{1}{2N} \sum_{i=1}^{N} \left| Y_i^{re} + Y_i^{im} \right|^2 \tag{3.17}$$

where $T(Y)$ represents the test results for any vehicle. Using the chi-square probability distribution function, this random variable $T(Y)$ can be approximated as

$$P_d(\varepsilon, t) = \mathbb{Q}\left( \left( \frac{\varepsilon}{\sigma_u^2} - \gamma - 1 \right) \sqrt{\frac{tf_s}{2\gamma+1}} \right) \tag{3.18}$$

The car is recognized as the PU if the probability of detection $P_d$ is higher than the threshold $\varepsilon$ [94]. $\varepsilon$ is calculated by the inverse of this chi-pdf:

$$\mathbb{Q}^{-1}(P_d) = \left(\frac{\varepsilon}{\sigma_u^2} - \gamma - 1\right)\sqrt{\frac{tf_s}{2\gamma+1}} \tag{3.19}$$

Here, $\mathbb{Q}(.)$ is the complementary distribution function and is Gaussian in nature, i.e.

$$\mathbb{Q}(x) = \frac{1}{\sqrt{2\pi}}\int_x^\infty exp(-\frac{t^2}{2})dt \tag{3.20}$$

The threshold value decides the accuracy of detection of the PU. In this work, we follow the concept of deep learning to detect the presence of PU. It has proposed the stack of deep learning layers with LSTM in the focus to factor down the signal features. Threshold value estimate is comprised of two distinct stages: data generation and deep learning model training.

***Data Generation***

To collect data, the spectrum sensing network is simulated with an ideal energy-sensing condition to generate the training data with various modulation schemes and random input data streams. Simulated modulations are BPSK, QPSK, 8-PSK and 16-PSK. With every simulation, the generated signals' energy is mapped with the results of the PU detection with a threshold value calculated using Equation (3.19). As a result, forty thousand samples are used to create a labeled dataset. The PU and non-PU labels are assigned to detected signals.

***LSTM Network Training***

The LSTM network is trained on the data to teach the decision based on sensed signal energy. The network is trained with the randomly sampled 90% data for training and 10% for the testing. Two biLSTM layers with forward and backward data sequencing are used which are connected with the fully connected layer. On training, the network is able to correctly classify the absence of any PU upto 89% whereas any PU is correctly detected upto 83.5%.

The trained network is used to obtain the threshold value for $P_d(\varepsilon, t)$ on the unknown signals. For every successful detection, the trust score $(\mathcal{t})$ is incremented. The higher $\mathcal{t}$ is, the higher the probability of a vehicle to be elected as CH will be. This trust score calculation scheme is portrayed in Figure 3.5.

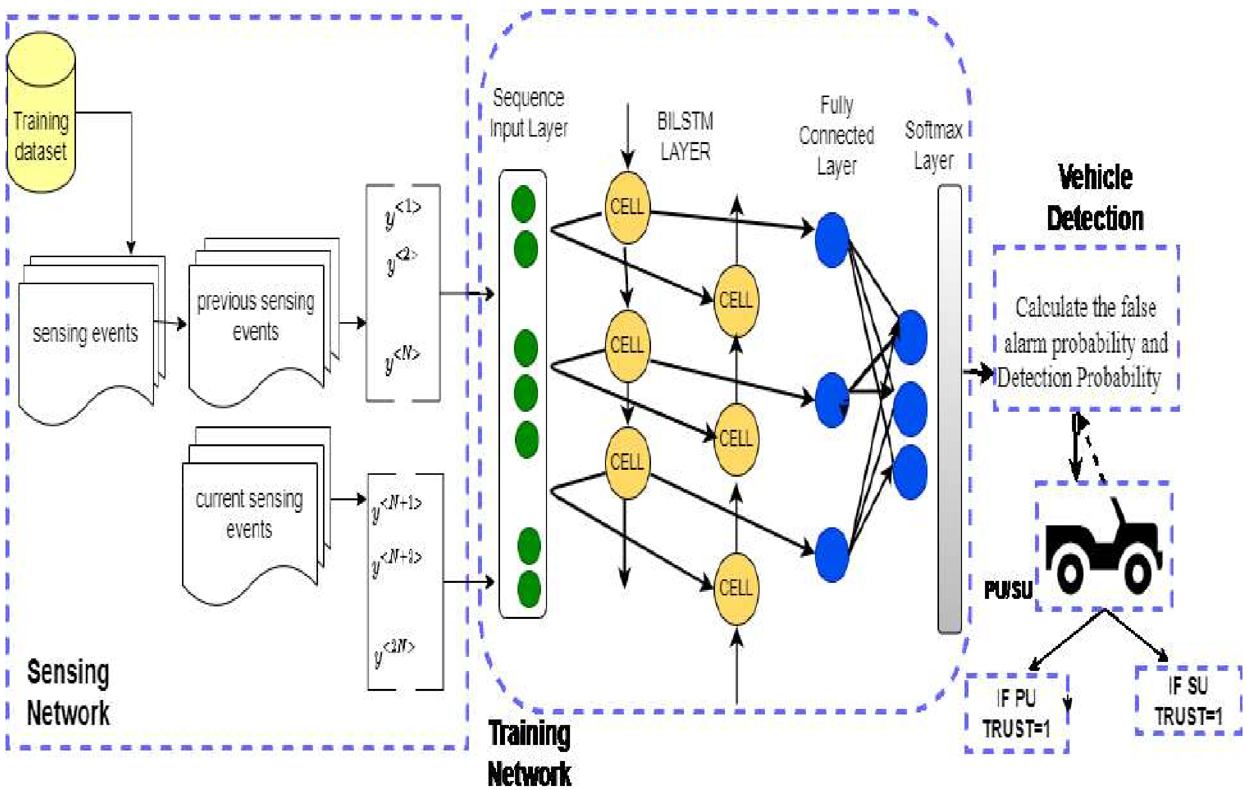



Figure 3.5 Adaptive Spectrum Sensing Model Using LSTM.

The model is divided into three subparts: sensing block, training block and PU detection block. The energy signal database is collected by simulating the network in the ideal and Rayleigh noisy channel environment. The data are fed into the LSTM training block. After training, the detected energy signal is tested with the trained model, and the vehicle is assigned to the PU or SU. The true detection increases the trust score by 1.

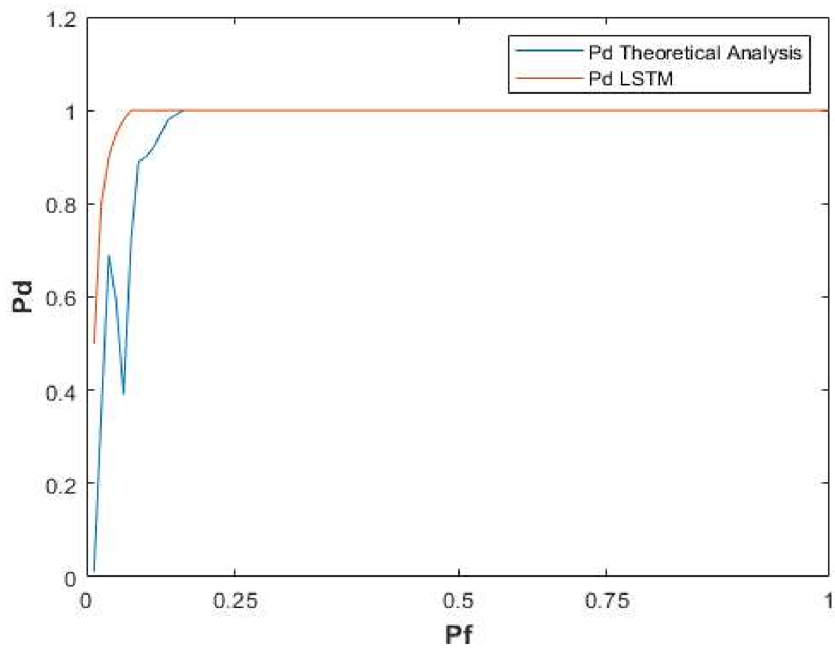



Figure 3.6 The $\mathbf{P_d}$ and $\mathbf{P_f}$Using LSTM Method and Theoretical Analysis.

A comparison of the ROC curve between the theoretical analysis of threshold calculation by Equation (3.19) as in [95] and the proposed LSTM-trained network is presented in Figure 3.6. The detection probability $\boldsymbol{P_d}$ is high for a small value of $\boldsymbol{P_f}$. The trained LSTM network performs better in a noisy environment, which means that it is efficient and can predict with appropriate accuracy.

In Algorithm 3.4, the complete algorithm created for CH selection is displayed. All four parameters are summed and integrated to select a stable CH for a long period of time. The vehicles are firstly clustered using Algorithm 3.2. For all members at each cluster, the four parameters are found. Then, $\boldsymbol{CH_{score}}$ is calculated to select the stable CH.

**Algorithm 3.4: CH selection**

***Input:*** *Relative speed* $V_{vehi}$ *and location* $[x, y]$;
*Number of lanes in a map* ($no\_lane$); *time_span; Cluster member (CM);(N) no. of clusters.*

1. **For** t=1:$time\_span$
2. **For** $i = 1: N$
3. **For** $j = 1: CM$
4. Calculate $\psi_{vehi}$ from Equation (3.14)
5. Find the neighbouring vehicles and calculate $\eta$ by using a connection matrix Equation (3.15)
6. Calculate the maximum eigenvalues $\lambda$;
7. Then obtain $\mathfrak{E}$ *by* using Equation (3.16)
8. Signals' energy is mapped with the results of the PU detection with a threshold value calculated using Equation (3.17) and Equation (3.19).
9. Calculate $\mathfrak{t}$ score using the LSTM trained network based on sensed signal energy.
10. Find $CH_{score}$ for each CM
11. $CH_{score} = \psi_{vehi} + \eta + \mathfrak{E} + \mathfrak{t}$
12. **End**
13. $CH_j = \text{Max} \sum_{j=1}^{k} CH_{score} \ \forall j$
14. **End**
15. **End**

***Output:*** *CH Vehicle*

## 3.3.2 HGCM Maintenance Phase

The reduction of communication overhead after the selection of CH is also an important part of the designed algorithm. The cluster maintenance process ensures strong connectivity and stable link lifetime through CH. In this work, the joining of a new vehicle in a cluster and leaving of any CM are considered vital for the cluster maintenance phase. HGCM is designed with the parameters that ingest the restructuration of the topology and vehicular speed. The CH score provides a significant contribution to capturing the information of vehicle movement. The maintenance does not deal with the networking. It is designed to maintain a smooth transition of vehicles in and out over time.

### 1) Cluster Enrolment

A small number of vehicles in a cluster with a large transmission range will lead to inconsiderably reliable networking. The selected CH starts its task by sending the polling signals and if it receives any signal in return within a stipulated time period time_span with the condition that $dist\_(vehi, CH) < R_{CH}$. A new vehicle is assigned to that cluster (each formed under RSU) and becomes CM of that particular cluster. CH will update the local database and the list of vehicles in RSU. The arrival of a new vehicle in a cluster can also trigger CH reselection in the worst-case scenario. The algorithm designed is thus coined to formulate CH score based on four factors $\{\psi_{vehi}, \eta, t, \mathfrak{E}\}$; with this, the stability of CH is ensured.

### 2) Cluster Leaving

CMs can leave any cluster at any moment of time. The reasons for this could be lane change or exit from a road, the ever-changing dynamics of vehicles and the topology that affects the number of CMs. Therefore, a frequent polling of signals is done between the established members and CH. If a CM does not reply within the speculated period time_span, then the CM is considered to be disconnected and leave the cluster. The CH removes the recorded vehicle, and an updated list is appended at the RSU. The complete algorithm designed for cluster maintenance is shown in Algorithm 3.5.

**Algorithm 3.5: Cluster Maintenance**

***Input:*** *CH; Cluster C; No. of vehicles N; CH transmission range* $R_{CH}$

**For** t=1:$time\_span$
  **For** $j = 1{:}C$
    **For** $i = 1{:}N$
        **IF** $dist\_(vehi, CH) < R_{CH}$
          CM=CM+1;
CH $< Adds\ the\ vehicle\ (vehi)\ in\ the\ RSU\ List\ \&\ Local\ List\ >$
        **Else**
          CM=CM-1
CH $< Removes\ the\ vehicle\ (vehi)\ from\ the\ RSU\ List\ \&\ Local\ List\ >$
            **End**
          **End**
        **End**
      **End**

***Output:*** *Cluster Update*

### 3.3.3 Time Complexity of the HGCM Scheme

Cluster formation, RSU deployment and CH selection are the key components of the proposed scheme. Accordingly, the total time complexity for our HGCM is expressed as

$$O_{TOT} = O_{CF} + O_{RSU} + O_{CH} \tag{3.21}$$

where $O_{CF}$ is the cluster formation's time complexity, $O_{RSU}$ is for RSU deployment, and $O_{CH}$ is for CH selection. The cluster is generated by hypergraph partitioning. The major steps involved are as follows: (1) hypergraph construction$= (V, E, W)$; (2) Laplacian construction; (3) eigenproblem solving; (4) applying k-means to $\bar{U}$.

When constructing the nearest neighbour graph in a hypergraph with pairwise similarity, the cost is $O(N^2)d$ , given that it requires $d$-dimensional similarity computation for each vertex pair, where $N$ represents the largest number of cars in the study' worst-case scenario with $m$ hyperedges. The sparsity of the adjacency matrix $A$ has a direct relationship with the Laplacian construction step through the non-zero elements $NNZ$ (or the number of vehicles in our case).

$L = O(NNZ(A^2))$, the eigen complexity is $E_C = O(N^3)$, and the last is the k-means complexity which is dependent on $O_{C_{optimal}} = O(\tau NC_{num})$, where $\tau$ is the number of iterations.

$$O_{CF} = O(N^2)d + O(NNZ(A^2)) + O(N^3) + O(\tau NC_{num}) \tag{3.22}$$

This can be reduced after removing the terms of less computational power as

$$O_{CF} = O(N^2)d + O(N^3) \tag{3.23}$$

The RSU deployment is done using a graph, so the computational complexity is

$$O_{RSU} = O(N^2)d \tag{3.24}$$

In this proposed, the CH selection metric $m_i(t)$ is a collection of metrics $\{\psi_{\text{vehi}}, \eta, \mathfrak{E}, t\}$. $\psi_{\text{vehi}}$ is the relative speed, $\eta$ is the set of neighbours of vehicle $v_i$, $\mathfrak{E}$ is the eccentricity, and $t$ is the trust calculated via spectrum sensing

$$O_{CH} = O_{\psi_{\text{vehi}}} + O_{\eta} + O_{\mathfrak{E}} + O_{t} \tag{3.25}$$

The relative speed is a simple threshold function done on the basis of the vehicle speed; thus,

$$O_{\psi_{\text{vehi}}} = O(N) \tag{3.26}$$

The next is the neighbourhood, which is a function of the $c_{ij}$ affinity matrix for nearby vehicles.

$$O_{\eta} = O\ (\log N^2) \tag{3.27}$$

Utilizing spectral clustering techniques which include the affinity matrix and eigenvalue decomposition, the eccentricity is computed. The entire spectral clustering complexity is

$$O_{\mathfrak{E}} = O(N^2)d + O\ (N^3) \tag{3.28}$$

The last factor is trust LSTM, which plays the primary role in this for spectrum sensing; The LSTM's theoretical time complexity is provided as

$$O_{LSTM} = O(4IH + 4H^2 + 3H + HK) \tag{3.29}$$

where $I$ represents the number of inputs, $K$ represents the number of outputs, and $H$ represents the number of hidden layers. In this study, because the model is trained only once for a given vehicle signal, the LSTM detects whether the vehicle is a primary or secondary user through spectrum sensing. Thus, the time complexity condenses to

$$O_{LSTM} = O(4H^2) \tag{3.30}$$

Then, the complete time complexity is reduced to moving all the terms with less complexity than cubic and quadratic terms, as shown below:

$$O_{CH} = O\ (\log N^2) + O(N^3) \tag{3.31}$$

The overall complexity is primarily dependent on the hypergraph, i.e.

$$O_{TOT} = 2\ O(N^2)d + O\ (\log N^2) + 2O\ (N^3) \tag{3.32}$$

## 3.4 Simulation and Performance Evaluation

This section describes the detailed background of the simulation tools and the various evaluation parameters utilised. The results' discussion is carried out in three phases: the effect of different traffic densities on the stability of the designed HGCM in a comparison with individual metric, State-of-the-Art Comparison, and the effect of different traffic densities on the routing performance.

### 3.4.1 Simulation Tools Used

The simulation is implemented using MATLAB (R2018b), with the processor Intel ® Core TM i7, 1.98 GHz, (SUMO 1.7.0) [96] and (TraCI) [97]. SUMO is an open-source microscopic road traffic simulator licensed under the General Public License. It was created through cooperation between the Institute of Transportation Systems at the German Aerospace Center (DLR), the Centre for Applied Informatics, and Cologne (ZAIK). TraCI is an API developed to interface SUMO with other coding platforms, such as MATLAB, Python and Java. The complete network design and the routing performance for metrics, such as throughput and PD are evaluated in MATLAB. The area considered for the study is a crowded market area of Baghdad having the latitude =33.3573° S, 33.3730° N and longitude= 44.3960° W, 44.4190° E; it is extracted from OSM. Baghdad is the capital of Iraq and one of the largest cities in the Arab world, with massive population and a geographical area of 204 km$^2$. Table 3.1 presents a summary of the traffic environment parameters. These parameters are considered on the basis of extensive literature survey. Moreover, the values are minutely crafted to portray a real urban scenario with congestion, many crossroads and a large number of vehicles during peak hours [53].

Table 3.1 Simulation Parameters.

| **Parameter** | **Value** |
|---|---|
| Scenario | Urban |
| Vehicle density | 100–1000 |
| Lanes | 458 lanes |
| Vehicle average speed | 10.71 m/s |
| $R_{RSU}$ | 350 m |
| Maximum Simulation time ($t$) | 1500 s |
| $R_{vehi}$ | 200 m |
| Length of vehicle | 5 and 10 m |
| Maximum vehicle speed | 25 m/s |

The geographical region considered for the simulation is shown in Figure 3.7; it is a vast area with urban infrastructure. Algorithm 3.2 suggests the optimal number of clusters in that region, and RSUs are deployed using Algorithm 3.3. Further vehicle features $f_i(t) = \{\vec{s}, \vec{p}, a, \theta, Vehi_{ID}, \eta\}$ are recorded for 1000 vehicles. The number of vehicles in the simulation area varies as in real-world scenarios. Twelve clusters are optimally selected using Algorithm 3.2. Different colours portray each cluster and vehicles in each cluster. The triangles (in black) are the different RSUs placed, which will serve as the cluster centre (providing auxiliary facilities), Figure 3.8.

Figure 3.7 Simulated Part of Baghdad Real Map in SUMO.

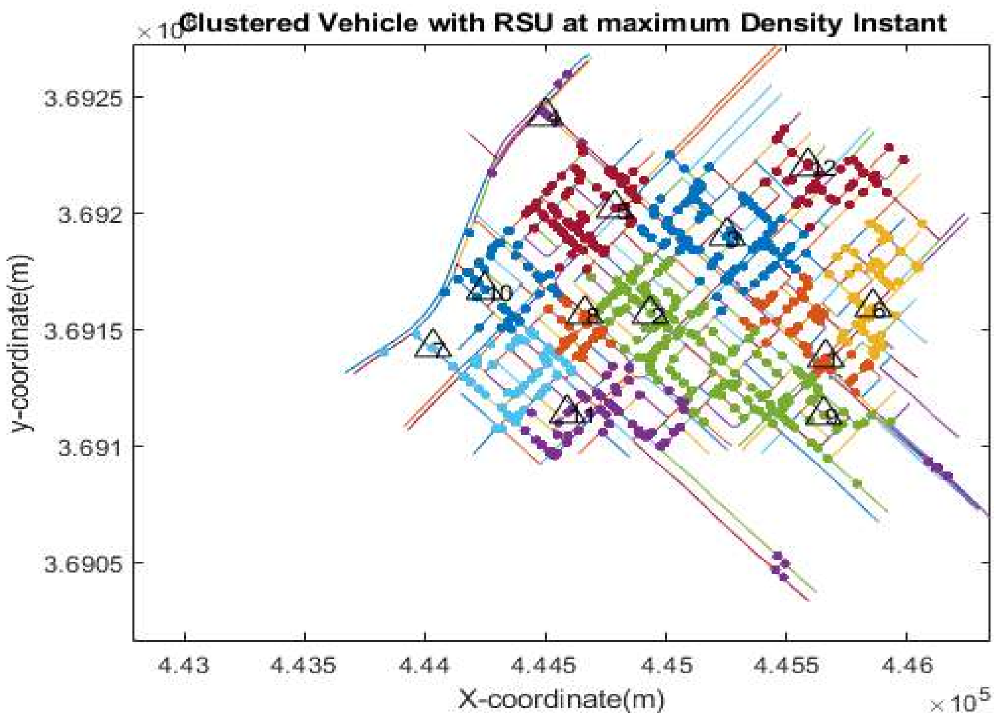



Figure 3.8 Cluster Formation Along with RSU Deployment at an Instant with Maximum Vehicular Density.

### 3.4.2 Evaluation Metrics

The designed HGCM is also tested in terms of routing performance. The communication amongst vehicles is modelled through the Rayleigh fading channel with BPSK modulation. Owing to the vehicles' movement, the network is dynamic and fast, which introduces a Doppler effect. The effect is incorporated as the signal fades over time. The communication network parameters are listed in Table 3.2 [3].

Table 3.2 Integrated Network Parameters for IEEE 802.11p.

| Parameters | Value |
|---|---|
| Fading channel | Rayleigh Channel |
| Fading Parameter | Filtered Gaussian noise |
| Doppler shift | $V_{s,r}/\lambda$ |
| Sample rate | 10 MHz |
| FFT points number | 32 |
| Subcarriers number | 52+ D |
| Number of data subcarriers | 52 |
| Number of pilot subcarriers | 4 |
| Cyclic prefix | 16 |
| Modulation scheme | BPSK |
| Coding scheme | ½ |
| Available data rate | 14Mbps |

Various metrics are computed to assess the performance and stability of our HGCM. These metrics are Cluster number, Cluster size, CH Lifetime, CM Lifetime, CH Change Rate, PD, and Throughput [5], [98].

### 3.4.3 Results and Discussion

The designed HGCM is analysed on a real map in an urban scenario where different densities of vehicles at various mobilities are infused into the network. The number of clusters produced throughout time affects algorithm efficiency as well. In comparison with roads, mobility is less of an issue in cities. CH is also more stable in an urban setting, where the vehicle density is higher, but the mobility is lower. The results of the effect of different traffic densities on HGCM stability and on the routing performance are presented in this section.

#### 3.4.3.1 Effect of Different Traffic Densities on HGCM Stability

The work by Maoli et al. that was presented in [82] opened up a way to present VANET as a hypergraph, Despite the fact that the authors only touched on that in the context of fog computing, they did not touch on the network performance metrics.

The effectiveness of the designed algorithm was also gazed by the number of clusters formed over time. These numbers allow us to evaluate the quality of the formed clusters. Few clusters with vehicles having low mobility achieve efficient connection and stable clustering. On the contrary, more clusters eventually lead to high overhead and mergers. The average number of vehicles in a cluster represents the cluster size. The larger the cluster size is, the higher the clustering efficiency will be. Figure 3.9, shows the average number of vehicles in a cluster and the number of clusters generated at different vehicular densities for our HGCM.

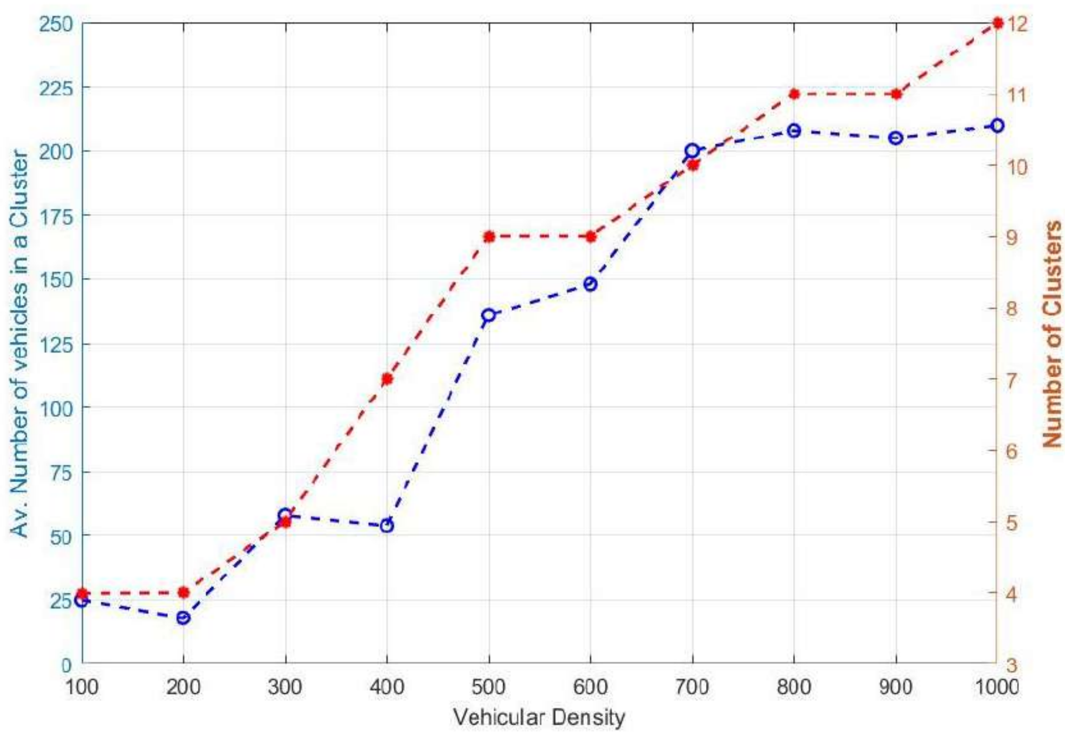



Figure 3.9 Av. Number of Vehicles in a Cluster and Cluster number of HGCM at Different Densities.

In HGCM, four and 12 clusters are generated with an average of 25 and 210 vehicles in a cluster at low and high traffic, respectively.

For spectral clustering, this study employs the eigenvalues derived using VANET's hypergraph presentation. The idea is motivated by the connectivity graph eigenvalues in [58] and [70]. Both works in [58]and [70]were designed for the highway scenario, whereas our work is designed for the urban environment. The eccentricity parameter in our work is inspired by the connectivity-based CH selection in [58]. A high connectivity with vehicles represents that dense traffic and maximum CH can be connected with the maximum number of vehicles. Eccentricity is a positional parameter that can be correlated with the connectivity issue. In a graph network, the central point has the highest connectivity, as does in the hypergraph. The neighbourhood degree is another connectivity parameter. In CH selection, a relative vehicle speed denotes uniform cluster generation. The CH stability using these three parameters $\{\boldsymbol{\psi}_{\boldsymbol{vehi}}, \boldsymbol{\eta}, \boldsymbol{\mathfrak{E}}\}$ is evaluated on different vehicle densities in the same network and represented in Figure 3.10. Given that the vehicle deployment and movement are random and near to a real environment in SUMO, $\{\boldsymbol{\psi}_{\boldsymbol{vehi}}, \boldsymbol{\eta}, \boldsymbol{\mathfrak{E}}\}$ parameters are not able to conclude any concrete pattern. We hereby use a nontrivial CH selection parameter, i.e. trust score $\boldsymbol{t}$. The trust score $\boldsymbol{t}$, along with the remaining three CH selection parameters, improves CH stability. The novel set of CH selection parameters significantly improves the CH stability by 20% at all vehicle densities as shown in Figure 3.10.

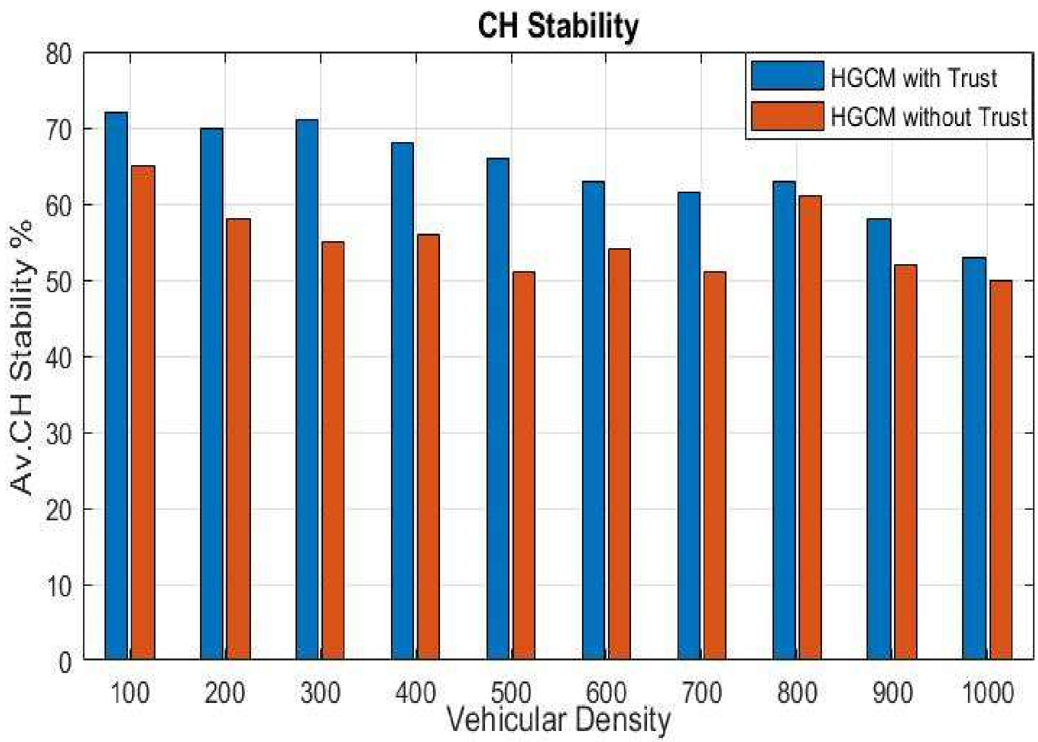



Figure 3.10 Average Stability of CH for HGCM with and without Trust Factor.

Regardless of the non-uniform pattern in improving stability by the proposed set of parameters, the novel contribution shows a constant improvement compared with each parameter, as shown in Figure 3.11. The method designed using eccentricity only provides satisfactory stability compared with the others. This is because the network has a dynamic structure that is perfectly emulated utilising the hypergraph concept. By contrast, the rest of the parameters, such as the relative speed and neighbours, could not trace the stability with increasing vehicle densities. The contribution of each at an individual level is low, but the stability provided is best when they are combined $\{\boldsymbol{\psi_{vehi}}, \boldsymbol{\eta}, \boldsymbol{\mathfrak{E}}, \boldsymbol{t}\}$.

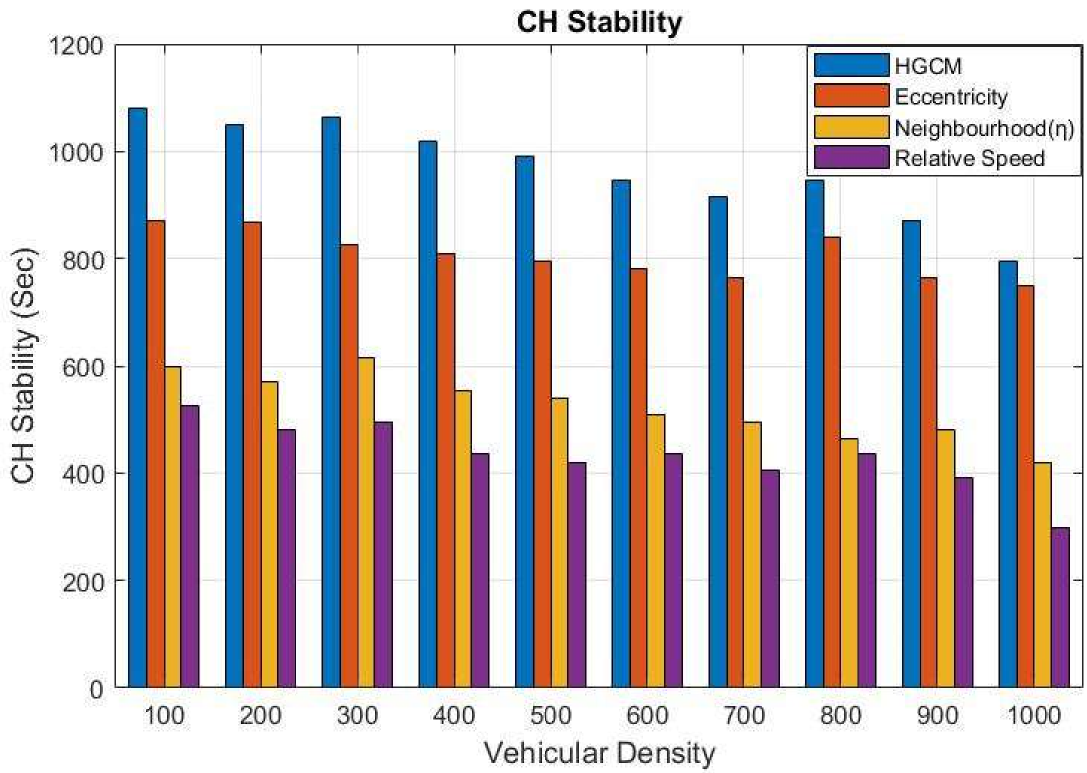



Figure 3.11 Stability of CH at Different Vehicle Densities.

The efficient cluster generation in the proposed scheme leads to enhanced CH stability. The CH stability has already been validated in Figure 3.10 and 3.11. The proposed HGCM with the four CH selection parameters achieves 72% and 53% of stability at low and high traffic density, respectively.

The CH achieves enhanced stability, as evaluated in Figure 3.11. However, other vehicles in the cluster are marked as CMs. The increased lifetime of CMs indicates efficient clustering by using a hypergraph. In the case of non-uniform clustering, a CM leaves clusters frequently and joins others. Figure 3.12 presents a comparison of CM lifetime of our proposed HGCM. The HGCM scheme gains the highest lifetime compared with its counterparts, although the traffic congestion with the increase in vehicle' density imposes performance degradation. Nevertheless, it can be ignored because for a 10-fold increase in traffic from 100 vehicles to 1000 vehicles in the network, the CM lifetime decreases to 4.2% only.

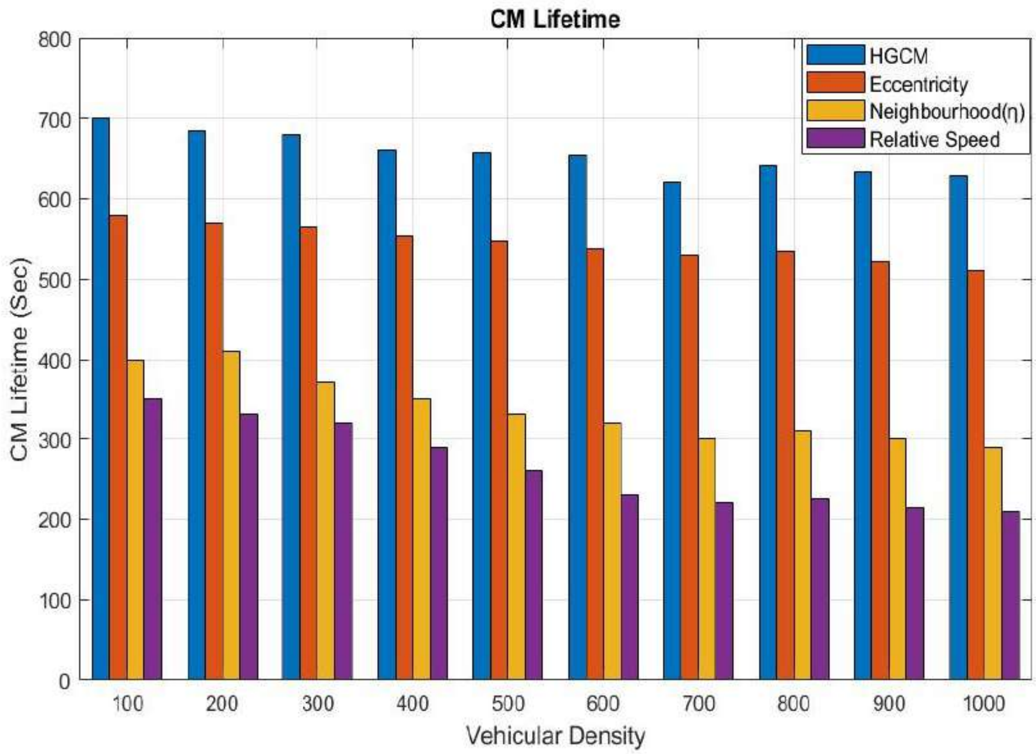



Figure 3.12 CM lifetime at Different Vehicle Densities.

Also, the lower change rate of the CH, the more stable the cluster structure. From Figure 3.13 we can see, that the CH change rate is the lowest due to the hypergraph spectral clustered network with the CH selected using the cumulative multimetric $\{ \psi_{vehi}, \eta, \mathfrak{E}, t\}$. Our proposed achieve 0.2 and 0.34 /s of the change rate of the CH at low and high traffic, respectively.

The overhead caused by the CH's frequent switching from one vehicle to another is decreased by the cumulative multimeric. Thus, it improves CH stability and CM lifetime and reduces CH change rate in a comparison with individual metrics.

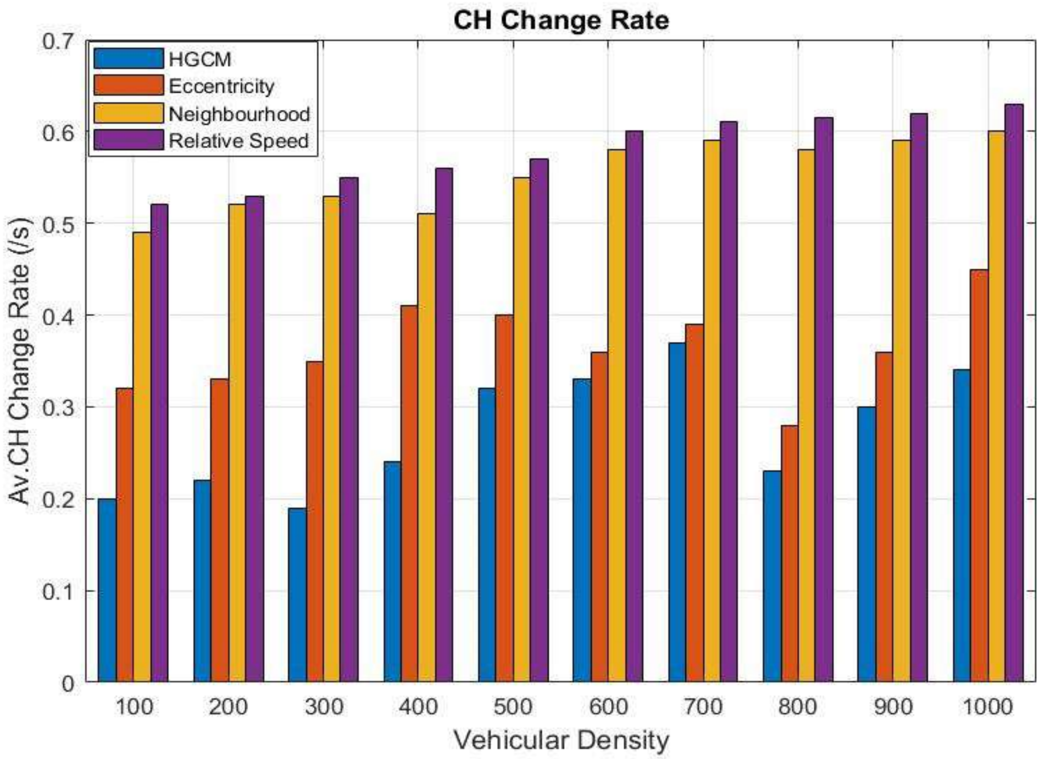



Figure 3.13 CH Change Rate at Different Vehicle Densities.

#### 3.4.3.2 State-of-the-Art Comparison

Figure 3.14 shows a graph that plots the CH stability for different vehicle densities. The CH stability decreases with the increase of traffic density. On the same network conditions listed in Tables 3.1 and 3.2, the CH stability is also evaluated using the algorithms in articles [58], and [35], with the same vehicle properties recorded from SUMO as for the proposed work. In Figure 3.14, CVoEG [58] seems to select a lesser durable CH than the proposed HGCM, followed by the method proposed by Arkian et al. [35]. The reason is that the CH selection in Arkian et al.'s method [35] is based on vehicle speed. As we have mentioned previously, the speed metric is lost in an urban scenario when there is immense congestion. Hence, this proposed method achieves the lowest stability. HGCM achieves good CH stability in comparison with other algorithms due to the effectiveness of hypergraph theory and the novel set of CH selection parameters. Our HGCM succeeds in achieving more than 53% of CH stability from the total time at all vehicle densities. Thus, the presentation of VANET as a hypergraph with its eigenvalues improves the CH stability.

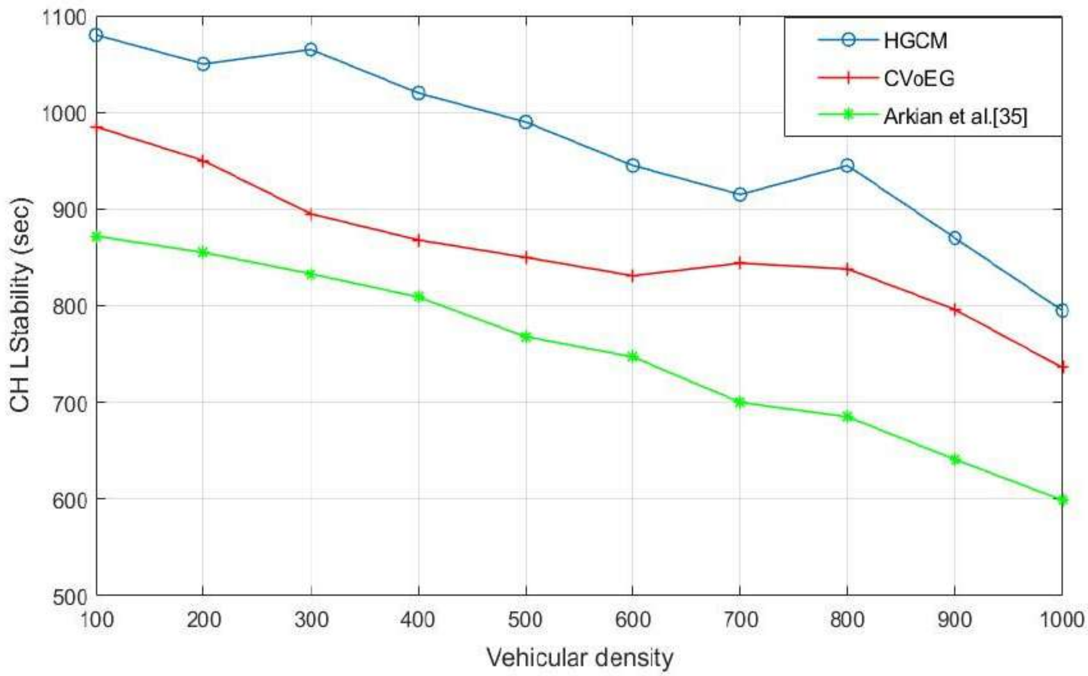



Figure 3.14 Stability of CH at Different Vehicle Densities.

### 3.4.3.3 Effect of Different Traffic Densities on Routing Performance

The stable CH improves the routing parameters, such as PD and throughput. These parameters are distance dependent. The minimum distance travelled by the packet leads to low PD and high throughput. All CMs should be one hop away from the CH. In an efficient cluster, the hop distance would be minimal. In the work presented in this chapter, HGCM divides the network into 12 efficient clusters, which results in an average hop distance of 150 m for 1000 vehicles. By contrast, it is 260 and 330 m for CVoEG [58], and the method proposed by Arkian et al. [35], respectively. Figure 3.15 shows the hop distance versus vehicle density curves on the right-hand y-axis and PD versus vehicle density on the left-hand y-axis. The maximum delay is witnessed in the method proposed by Arkian et al. [35] because it has a maximum hop distance. The location of the CH in the method proposed by Arkian et al. [35] is random, and it does not guarantee the centrality of CH while the CH location in CVoEG. [58] is chosen based on the graph centrality. PD is low for a small average hop distance. With the increase in vehicle densities, the average hop distance increases and so is the PD. This finding validates that HGCM clustering shows better performance for a sparse network, which aligns with the general convention that a crowded area increases PD. In sum, our HGCM reduces the PD by approximately 41% and 48% compared with the methods of CVoEG [58] and Arkian et al. [35], respectively, at high traffic.

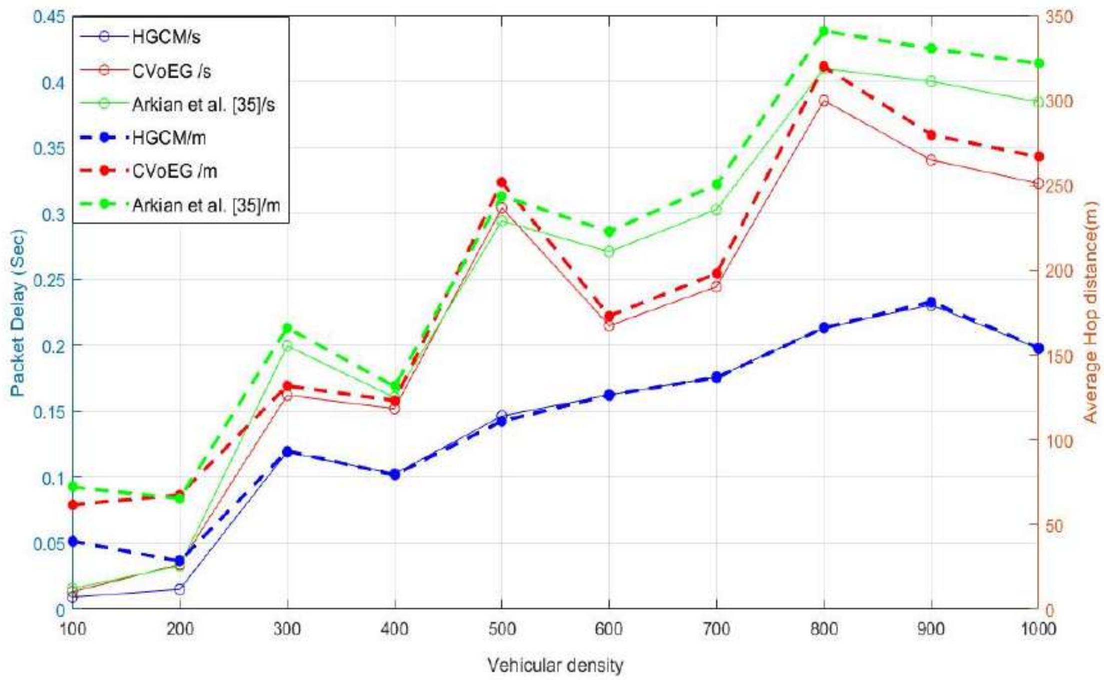



Figure 3.15 Analysis of PD and Hop Distance at Different Vehicle Densities.

Throughput depends on the number of packets received in a small span. The minimum PD increases the throughput for the proposed HGCM scheme irrespective of the number of vehicles. Figure 3.16 shows the throughput curves. The hypergraph network presentation and novel set of CH selection help achieve 460 kb/s throughput compared with 330, and 310 kb/s in other works at a density of 1000 vehicles. The proposed scheme helps achieve consistently improved throughput performance by approximately 39% and 48% compared with the methods of CVoEG and Arkian et al. [35], respectively, for high traffic.

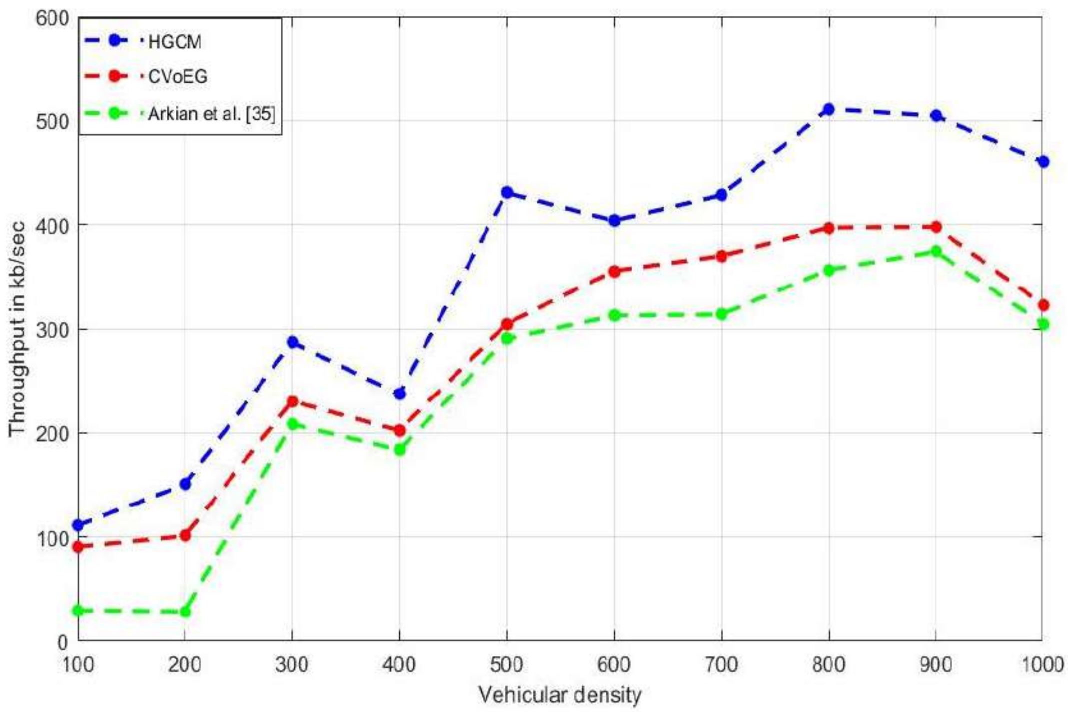



Figure 3.16 Throughput at Different Vehicle Densities.

## 3.5 Conclusion

We have developed a novel cluster generation and maintenance strategy in this chapter. The CH is chosen based on a combination of four indicators that help maintain the stability of the dynamic network. A changing structure and the frequent connection and disconnection of communication links amongst vehicles are modelled in an evolving hypergraph formulation of VANET. Spectral clustering creates the ideal number of groups on the basis of the density of vehicles. Each cluster has a single RSU at its centre. Relative velocity score, eccentricity, neighbourhood degree and trust score are all recommended in this study for finding the most stable CH in each cluster. Trust metric with deep learning spectrum sensing has been introduced for CH selection. Deep learning-trained spectrum sensing is used as a model for trust calculation. LSTM layers are used to identify the primary vehicle in both noiseless and noisy environments. A high trust score is awarded to the vehicle which vacates the spectrum in the sensing of the primary vehicle. The proposed HGCM is tested for various vehicle densities in a real area in Iraq's capital, Baghdad. Compared with individual measures and other techniques, our cumulative approach significantly improves CH stability. The addition of the trust element results in 20% gain in average CH stability over the combined performance of three existing measures (i.e. relative speed, eccentricity and neighbourhood). A one-hop network configuration is used to evaluate the approach for different integrated network metrics, including packet latency and throughput. The average packet distance travelled by the proposed method is 150 m with a delay of 0.2 s, whereas the other comparative algorithms under the same network conditions report a PD of 0.38 s for approximately 330 m according to the PD analysis for the worst-case scenario (i.e. 1000 vehicles). Therefore, HGCM has the lowest PD whilst still allowing for the shortest possible hop distance. In addition, PD directly influences throughput; hence, HGCM has the maximum throughput compared with other methods.

In the next chapter, we introduce a new proposed scheme based on improving the hypergraph spectral clustering algorithm by using the Eigen-trick method in the cluster formation phase. The Eigen-trick is used to calculate the modified Laplacian value in the TTM to improve the clustering. It improves the clustering efficiency by using higher-order information in eigenvalues. Four different parameters neighbourhood, eccentricity, relative speed and the key attribute to estimate the time to leave are extracted from each vehicle. The relational analysis of these four CH selection attributes is attempted with Grey Relational Analysis (GRA). The GRA is used to select a most stable CH, through which strong connectivity and stable link lifetime are maintained. There is an evident change in stability with the incorporation of junction information and reactive speed variation. Our proposed is named as an Eigen trick-based Hypergraph Stable Clustering (EtHgSC).

***Published papers***

1. M. K. Jabbar and H. Trabelsi, “A novelty of Hypergraph Clustering Model (HGCM) for Urban Scenario in VANET,” *IEEE Access*, 2022.
2. M. K. Jabbar and H. Trabelsi, “A Betweenness Centrality Based Clustering in VANETs”, *15th International Conference on Security of Information and Networks (SIN), 2022.*

# Chapter 4

# EtHgSC: Eigen trick-based Hypergraph Stable Clustering in VANET

## 4.1 Introduction

For unifying many interactions between items, a hypergraph is frequently employed. Spectral clustering is one of the best methods for dividing those objects (vertices) into distinct communities [99]. In the first proposed in Chapter 3, the hypergraph spectral clustering has been used to cluster the vehicles in the cluster formation phase. The hypergraph partitioning through the TTM method has been proposed.

This chapter presents the Eigen trick-based Hypergraph Stable Clustering model (EtHgSC), which has a two-fold scheme for stable clustering. In the first part of the proposed scheme, the cluster generation is handled using an improved hypergraph-based spectral clustering algorithm using the Eigen-trick method. The Eigen-trick is used to calculate the modified Laplacian value in the TTM to improve the clustering. Although the TTM was developed to partition the hypergraph into the lower non-decomposable graphs, the information loss can be minimised further with the help of the Eigen-trick in TTM hypergraph partitioning.

‘‘Eigen-trick” method is used to partition both vertices and hyperedges, this offers a method for lessening the clustering's computational complexity. The Eigen-trick improves the clustering efficiency by using higher-order information in eigenvalues. It considers the transformable connection between the vertex Laplacian and the hyperedge Laplacian, which can speed up the solution of eigenproblems. During this time, the partition for vertices and hyperedges are both obtained simultaneously [100].

The CH is chosen in the second part taking into account the requirements for keeping a stable connection with most neighbours. In addition to relative speed, neighbouring degree, and eccentricity that are used to select the CH, the vehicle time to leave estimation is introduced to increase the clustering stability. The grey relational analysis model is used to find each vehicle’s score. The vehicle with the highest score is selected as the CH for each cluster at each instant of time. The contributions of this chapter are listed as follows:

- The hypergraph-based spectral clustering algorithm is introduced for cluster formation, and the Laplacian value of the TTM is modified using the Eigen-trick method. It considers the transformable connection between the vertex Laplacian $Lv$ and the hyperedge Laplacian $Le$, which can speed up the solution of eigenproblems without losing information.
- The fast-moving vehicle may leave the cluster at any time and go out of the CH's transmission range. In that case, the packet drop would be higher. It is better to know the leaving time of the vehicle in the cluster. Time to leave ($T_{leave}$) is an important parameter used to select the stable CH.
- A vehicle may change its direction, so estimating the next position of the vehicle beforehand is necessary to know. Thus, the predicted vehicle direction at the next instant is calculated with the help of the predicted vehicle position by the Predictive directional Greedy Routing Protocol (PGRP). It helps to select the lane and its length to get the necessary estimated time to leave $T'_{leave}$. This parameter reduces the frequent cluster breakage at junctions.
- A set of CH selection parameters to deal with the vehicle's abrupt lane change is proposed, which includes relative speed, neighbouring degree, eccentricity, and $T'_{leave}$.
- A relational analysis is developed using Grey Relational Analysis (GRA) between four CH selection parameters instead of knowledge-based weightage. By using this scheme, strong connectivity and a stable link lifetime are obtained.

## 4.2 EtHgSC Proposed Model

In this chapter, a new clustering approach for a VANET structure in an urban scenario is presented. The proposed approach (EtHgSC) is divided into two parts: cluster generation using the improved hypergraph spectral clustering method and CH selection.CH stability is governed by using four factors, inclusive of relative speed, nieghbouring degree, eccentricity, and vehicle's time to leave with the estimated next vehicle's position. These two parts are explained in this section.

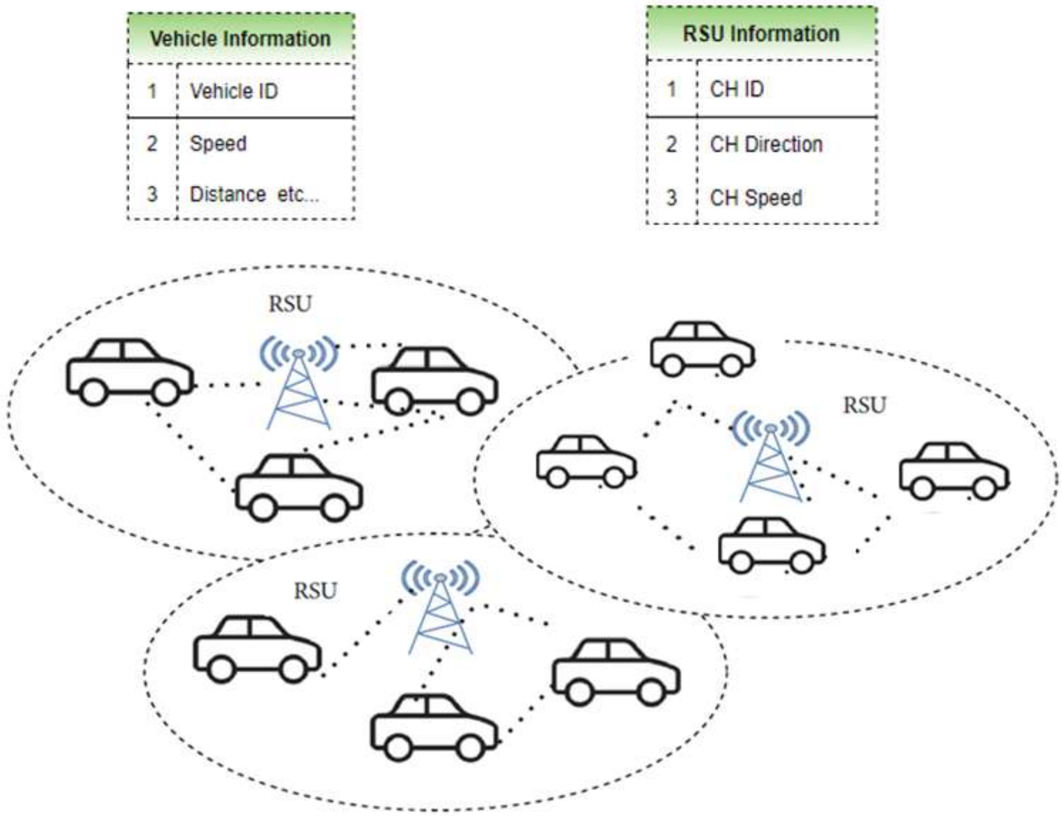


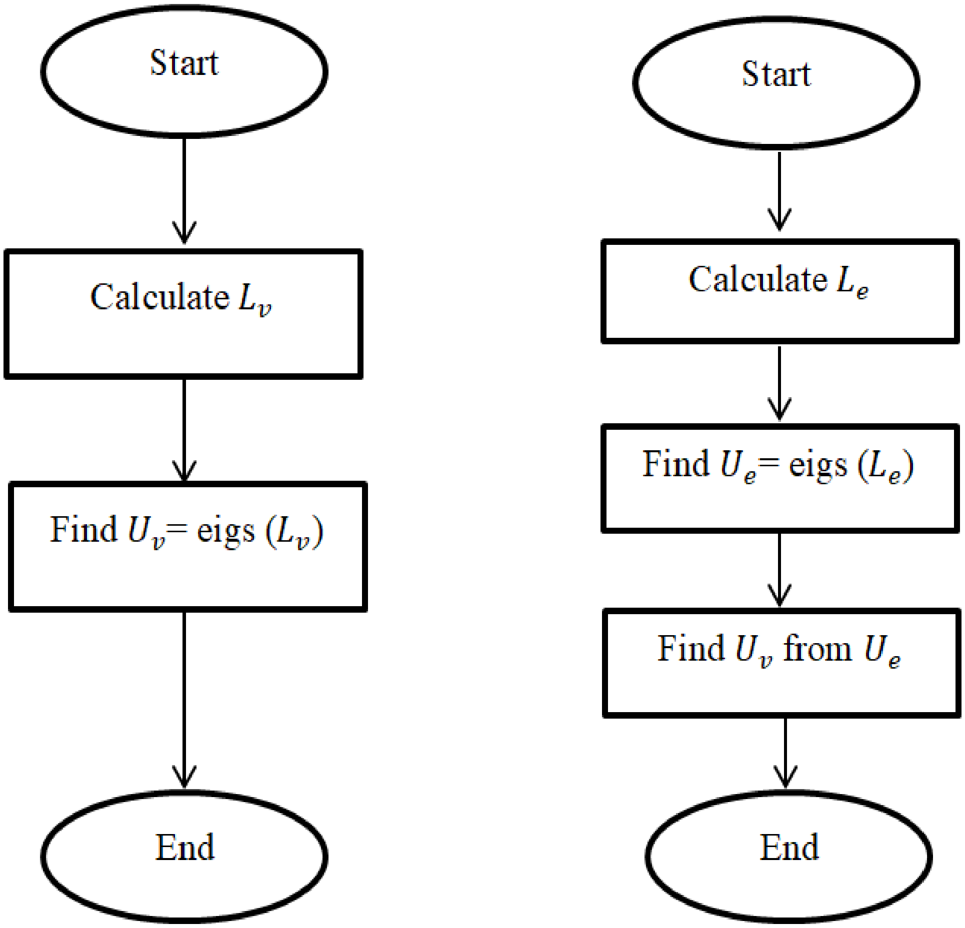

Figure 4.1 Proposed Model.

## 4.2.1 Cluster Generation

In spectral clustering, the most common criterion method is the radio cut method. For a set of clusters $V_1, V_2, \ldots, V_k$ satisfying $V_1 \cap \ldots \cap V_k = \emptyset$ and $V_1 \cup \ldots \cup V_k = V$. And as we mentioned before, $L$ is a spectral clustering major tool.

Weighted hypergraph TTM clustering has been proposed to use for spectral clustering of vehicles. It is called TTM because the hypergraph partition is equivalent to tensor decomposition into a lower rank [86]. Although the TTM is developed further to partition the hypergraph into the lower non-decomposable graphs, the information loss can be minimised further with the help of the Eigen-trick method in TTM hypergraph partitioning. Weighted hypergraph TTM to partition with Eigen-trick for the large-scale VANET for clustering is proposed in this chapter. $L$ in TTM is modified by the use of the Eigen-trick method. In this method, the eigenvectors of vertex Laplacian are computed from the corresponding eigenvectors of hyperedge Laplacian. Figure 4.2

depicts the proposed improvement in the hypergraph using the Eigen-trick method. In the weighted hypergraph $\mathcal{H} = (V, \mathcal{E}, W)$, the problem is partitioning $V$ into k disjoint sets $(V_1, \dots \dots V_k)$.




a. Traditional b. Eigen-trick

Figure 4.2 Obtaining $U_v$ using a. Traditional Hypergraph and b. Hypergraph with the Eigen-trick method.

The following are the steps for partitioning the hypergraph with the modified TTM using the Eigen-trick method:

1. For spectral clustering, the defined adjacency matrix is taken into account. It is defined here for the tensor (order m), and it is calculated using Equation (3.3) in Chapter 3.

$$A_{i_1,i_2,\dots\dots i_m} = \begin{cases} w_{\{i_1,i_2\dots\dots i_m\}} & if\ i_1,i_2,\dots\dots,i_m\ are\ distinct \\ 0 & otherwise \end{cases}$$

Also, the incidence matrix ($I$) is defined, $I \in \mathbb{R}^{N\times m}$. It represents the connection between vertices and edges. $i(v,e)=1$, if there is a connection between a vertex $v$ and an edge $e$, otherwise $i(v,e)=0$ [100].

2. The degree matrix (diagonal matrix) for vertices $Dig_v \in \mathbb{R}^{N\times N}$, and for edges $Dig_e \in \mathbb{R}^{m\times m}$ are obtained, where $\deg(v)=\sum_{e\in E} w_e$ and $\deg(e)=\sum_{v\in V} i(v,e)$.

In our work, the weighted incidence matrix is also used, it is calculated as:

$$i(v,e)=\begin{cases} w_e & if\ connected \\ 0 & otherwise \end{cases} \tag{4.1}$$

3. In actuality, there is a link between the symmetry of $L_v$ and $L_e$. In ‘‘Eigen-trick”, an interesting attribute of two Laplacians is that the k trailing eigenvectors of vertex Laplacian $U_v \in \mathbb{R}^{N\times k}$ can be computed from the corresponding eigenvectors of hyperedge Laplacian $U_e \in \mathbb{R}^{m\times k}$ and vice versa. $U_e = eig(\,L_e\,)$. The hyperedge Laplacian $L_e$ is calculated as, $L_e \in \mathbb{R}^{m\times m}$ [100].

$$L_e = {Dig_e}^{-1/2} W^{1/2}\, I^T\, Dig_v^{-1} I\, W^{1/2} {Dig_e}^{-1/2} \tag{4.2}$$

4. Obtained the $U_v \in \mathbb{R}^{N\times k}$ from the hyperadge eigenvectors $U_e$

$$U_v = {Dig_v}^{-\frac{1}{2}}\, I\, {Dig_e}^{-\frac{1}{2}}\, W^{1/2} U_e \tag{4.3}$$

5. View each row of $V$ as a vector for a node, and cluster the $N$ nodes into $k$ clusters $(V_1, V_2, \ldots, V_k)$ through k-means clustering. K-means clustering is applied on the normalized eigenvectors of vertex Laplacian $\overline{\mathrm{U_v}}$.

6. The greatest value of the $s$ index is used to select an ideal group of clusters from the pool of generated clusters. $C_{optimal} = [C_{num}\colon \forall\ \max(s\,)]$. The $s$ index is calculated using Equation (3.8) as in Chapter 3

$$s = \frac{tr(B_k)}{tr(Z_k)} \times \frac{Vehi_{num} - k}{k-1}$$

The complete cluster formation algorithm is as in Algorithm 4.1.

**Algorithm 4.1: Cluster Formation using Eigen-trick method**

**Input:** Maximum number of vehicles: N , Each vehicle's location : $\mathrm{Vehi_{Loc}}$

1. Choose the time $t$ when there are the most vehicles N
2. Form a hypergraph $\mathbb{H} = (\mathrm{V}, \mathcal{E}, \mathrm{W})$
3. Determine the similarity matrix A based on how close the vehicles are to one another in terms of distance

   A square matrix of size $\mathrm{A} \in \mathbb{R}^{\mathrm{N}\times\mathrm{N}}$ of similarity (adjacency matrix) as $A = \sum_{\mathrm{i_3,\ldots\ldots,i_m=1}}^{\mathrm{N}} \mathrm{A_{ij}}$

   Incidence matrix I is generated, its size of $I \in \mathbb{R}^{\mathrm{N}\times\mathrm{m}}$.
4. A diagonal matrix for vehicles $\mathrm{Dig_v} \in \mathbb{R}^{\mathrm{N}\times\mathrm{N}}$ , and for edges $\mathrm{Dig_e} \in \mathbb{R}^{\mathrm{m}\times\mathrm{m}}$
5. Find the hyperage Laplacian $\mathrm{L_e}$=$Dig_e^{-1/2} W^{1/2}\, I^T\, Dig_v^{-1} I\, W^{1/2} Dig_e{}^{-1/2}$
6. k dominant eigenvector of $\mathrm{L_e}$ are computed as $\mathrm{U_e} \in \mathbb{R}^{\mathrm{m}\times\mathrm{k}}$
7. Computed the $\mathrm{U_v} \in \mathbb{R}^{\mathrm{N}\times\mathrm{k}}$ From the hyperadge eigenvectors $\mathrm{U_e}$
8. Normalize each row of $\overline{\mathrm{U_v}} = \mathrm{U_v}$
9. Run k-means on the rows of $\overline{\mathrm{U_v}}$
10. Obtain $\mathrm{C_{num}}$ through k-means partition $\mathrm{Par} = \{\mathrm{V_1}, \ldots \ldots \mathrm{V_k}\}$
11. For each cluster $\mathrm{C_{num}}$, calculate the Calinski–Harabasz (s) criterion
12. Find the optimal cluster $\mathrm{C_{optimal}} = [\mathrm{C_{num}}: \forall\ \max(\mathrm{s})]$

**Output:** The optimal set of clusters : $\mathrm{C_{optimal}}$, partitioning vehicles in clusters

VANET's weaknesses as well as network performance are enhanced by installing RSUs. The RSUs can successfully meet increased reliable communication, vehicular density, and decreased overhead delay, particularly in urban environments. When connecting to one or more vehicle clusters, the RSU functions as a gateway inside a router. It can also connect with RSUs using wireless or optical networks through an inclusive deployment [101]. Few studies in the literature addressed the RSU deployment, but the RSU is a part of VANET, and it is most necessary, which we have developed in our work. The actual VANET network needs RSU communication, too [102].

So, after Algorithm 4.1 has produced the ideal set of clusters, the fetching location for RSU starts. Here, an evolving graph structure of the traffic is conceived using betweenness centrality as discussed in Algorithm 3.3 in Chapter 3. Now, the cluster generation phase for our scheme is completed.

### 4.2.2 CH Selection Measures

The next step is the selection of the CH. Continuous communication between the cluster and RSU, security, and upkeep of the routing path all fall under the purview of a CH. The stability of the CH in VANET is the main motive for designing any clustering algorithm. In the real scenario of driving, the driving decisions are highly dependent upon the number of neighbouring vehicles, their locations and speeds. A vehicle moving at a uniform relative speed with the neighbours and in a central location in relation to other vehicles has a higher probability of getting selected as the CH. For example, in 3 lanes road scenario, the vehicle moving in the central lane is highly probable to be CH, as it has an equal communication probability with both sides' vehicles. It is required that a vehicle should stay in the cluster for a longer period of time. The road junctions pose a challenge to this belief. So, in this proposed, we calculate the vehicle's time to leave the cluster ($T_{leave}$). $T_{leave}$ fails to estimate the road junctions. The unpredictable change in the direction of the vehicle at the junction can abruptly change the prediction. The new lane length may differ, so it is required to predict the vehicle's next position beforehand. The proposed methodology for the CH selection is based on four parameters: relative speed, neighbouring degree, eccentricity, and vehicle's time to leave with the estimated next vehicle's position. These parameters are discussed as follows:

### 1. Relative Speed Score ( $\psi_{vehi}$)

It is calculated using Equation (3.14) in Chapter 3.

$$\psi_{vehi}(t+1) = \psi_{vehi}(t) + \delta \quad ; \left|V_{vehi} - V_{avg}\right| \leq S_{thr}$$
$$\psi_{vehi}(t+1) = \psi_{vehi}(t) - \delta \quad ; \left|V_{vehi} - V_{avg}\right| > S_{thr}$$

### 2. Neighbouring Degree ($\eta$)

Neighbour degree is calculated using Equation (3.15) in Chapter 3.

$$\eta = \sum_{j=1}^{Vehi_{num}} c_{ij} \,; \forall\, 1$$

### 3. Eccentricity ($\mathfrak{E}$ )

The idea of eccentricity ($\mathfrak{E}$) is introduced to guarantee stability. It is calculated using Equation (3.16) in Chapter 3.

$$\mathfrak{E} = \frac{1}{|\,Vehi_{num}|} \sum\nolimits_{\lambda_i \,\epsilon\, Vehi_{num}} \lambda_i$$

### 4. Vehicle's Time to Leave Estimation ($T'_{leave}$)

All the above parameters are based on neighbourhood properties. The inclusion of a travel path is important to understand the complexity and speed of the travelling vehicle. The challenge of a vehicle's unpredictable direction at the road junction is dealt with in this parameter. The vehicles transit fast and leave the cluster, which eventually leaves the node to go out of the CH's transmission range. This decreases the throughput of the model designed. To address this issue, the leaving time of the vehicle in the cluster will fetch the maximum time a node spends in the cluster.

The time to leave is the amount of time needed for a vehicle to reach the last section of the lane. This parameter guarantees choosing a CH with enough time remaining to finish the lane, that causes it to head for a longer period. It is determined using the length of the lane $L_{th}$, the distance covered by a vehicle on the road segment $d_{vehi}$, and vehicle speed at that time $v_{vehi}$. The time to leave ($T_{leave}$) is calculated as:

$$T_{leave} = \frac{L_{th} - d_{vehi}}{v_{vehi}} \tag{4.4}$$

The possibility of the vehicle changing direction changes the lane length. Figure 4.3 can give a picture of the problem. $T_{leave}$ fails to estimate the road junctions because the lane lengths may differ. So, the vehicle's estimated next position must be known ahead of time. It helps to select the lane and its length to get the necessary $T'_{leave}$.

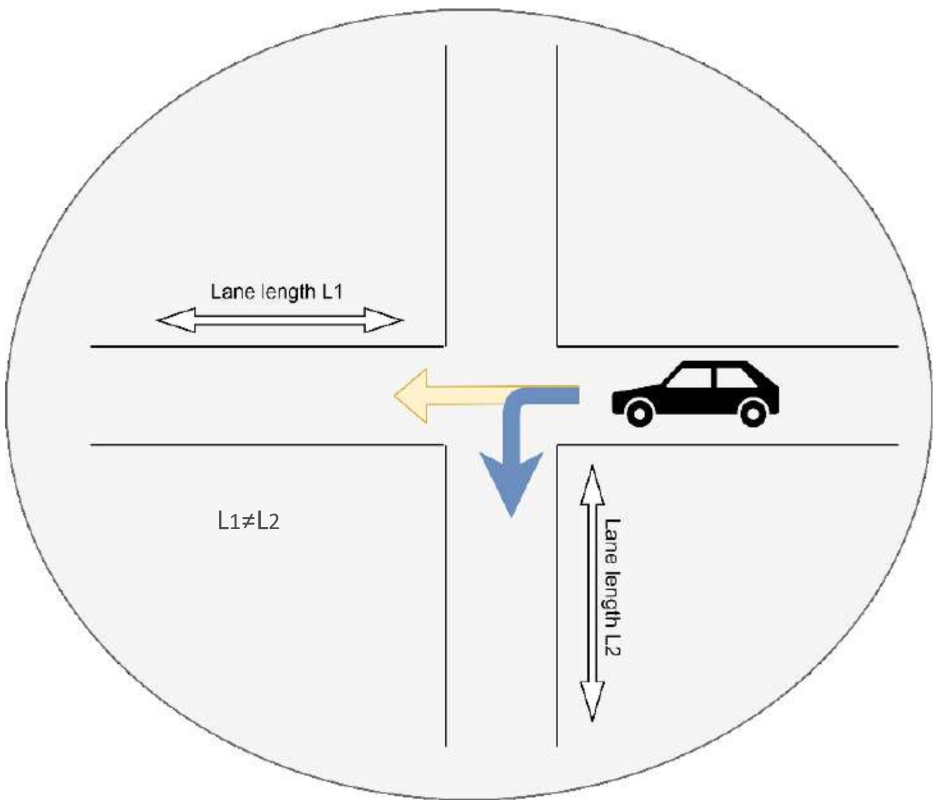



Figure 4.3 Time to leave estimation problem illustration in absence of next position information.

### *Estimated next vehicle position*

In the CH selection phase, this proposed has raised concerns about the CH stability at the junction, as the vehicle's direction is unpredictable at the road junction. The changes in the direction of travel define the time span of the vehicle under the signal CH. Thus, the prediction of vehicle direction is an important parameter of analysis. The $\theta'_{vehi}$ is the predicted direction of the vehicle at the next instant. It is calculated with the help of the predicted vehicle position by the Predictive directional greedy routing protocol (PGRP). $\theta'_{vehi}$ is calculated as:

$$\theta'_{vehi} = \tan^{-1}(Y_{vehi} - Y'_{vehi})/(X_{vehi} - X'_{vehi}) \tag{4.5}$$

Where $Y'_{vehi}$ and $X'_{vehi}$are the predicted new position while $Y_{vehi}$ and $X_{vehi}$ original coordinates of the vehicle. The routing algorithm is designed based on the parameter's position, direction, and speed of each neighbour to choose the next forwarding node [103].

$$\begin{cases} X'_{vehi} = X_{vehi} + d_{vehi} \cos \theta_{vehi} \\ Y'_{vehi} = Y_{vehi} + d_{vehi} \sin \theta_{vehi} \end{cases} \tag{4.6}$$

Here, $d_{vehi} = v_{vehi}\ t_{vehi}$. Equation (4.6) shows cases the change in the position of the vehicle from $(X_{vehi}, Y_{vehi})$ to $(X'_{vehi}, Y'_{vehi})$ when the vehicle is moving with a velocity of $v_{vehi}$, and the change in angle of $\theta_{vehi}$ at time $t_{vehi}$.

From the vehicle's predicted direction $\theta'_{vehi}$, the next lane length $L_{th}$and $d_{vehi}$ are known now, then we can calculate the estimated $T'_{leave}$ with predicted next vehicle's position using Equation (4.4).

The pseudocode for complete algorithm steps for $T'_{leave}$ is discussed in Algorithm 4.2. After calculating the four measures for each vehicle at each instant of time, the GRA model is used to find the vehicles' score. The GRA is explained in the next subsection.

| **Algorithm 4.2:** $T'_{leave}$ **Pseudocode for each vehicle in the cluster** |
|---|
| **Inputs:** $L_{th}$, $d_{vehi}$, $v_{vehi}$, $Vehi_{num}$: number of vehicles in the cluster<br>1. For $i = 1$: $Vehi_{num}$<br>Calculate the predicted vehicle position $(X'_{vehi}, Y'_{vehi})$ and direction $\theta'_{vehi_{t+1}}$ at the next time instant using PGRP, Equations (4,6) and (4.5), respectively.<br>a. if $\theta'_{vehi_{t+1}} \neq \theta_{vehi_t}$<br>Calculate the estimated time to leave $T'_{leave}$ using Equation (4.4) using the new Lane length and distance.<br>b. end if<br>2. end for |
| **Output:** $T'_{leave}$ |

### 4.2.3 Grey Relational Analysis Model

This chapter proposes a novel metric for selecting the CH that meets the requirements of maximum relative speed ( $\psi_{vehi}$), neighbourhood degree ($\eta$), eccentricity ($\mathfrak{E}$), and maximum time to leave estimation ($T'_{leave}$) based on GRA. The relation analysis is calculated between those parameters for each vehicle to get a single weighted relational value. The vehicle with the highest relational analysis value is assigned as the head of that cluster. GRA is part of grey system theory, which is suitable for solving problems with complicated interrelationships between multiple factors and variables. The fundamental advantage of GRA is that it can handle with data that is imprecise, ambiguous, or vague. This is due to the fact that GRA constructs the grey relational grade (GRG) using a mathematical technique. GRA is thus more robust than other systems that rely on heuristics or subjective judgments. It also has the advantage of being able to work with both quantitative and qualitative data. GRA is thus more adaptable than other approaches that demand data in a certain format. The procedures of GRA are shown in Figure 4.4.

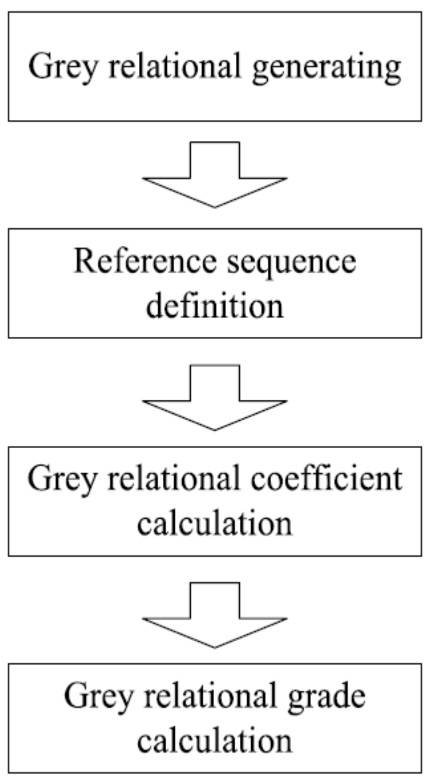



Figure 4.4 Grey Relational Analysis Procedure [104].

In GRA, there are three steps for decision making:

1. Find the grey relational generation.

2. Calculate the grey relational coefficient.

3. Use the grey relational coefficient to make a decision.

These steps are mathematically formulated as:

**Step 1:** The CH selection parameters are used to generate a matrix of $Vehi_{num} \times 4$, where $Vehi_{num}$ is the number of vehicles in a cluster. This matrix is used to generate a relational matrix as [104]

$$\mathrm{Y_{ik}} = \frac{\mathrm{x_{ik}} - \min(\mathrm{x_{ik}})}{\max(\mathrm{x_{ik}}) - \min(\mathrm{x_{ik}})} \tag{4.7}$$

The basic step of GRA is to convert the performance of each option $\mathrm{x_{ik}}$ into a comparability sequence which is denoted by the letters $\mathrm{Y_{ik}}$. A reference sequence is defined in accordance with these sequences. Equation (4.7) creates a normalized matrix of CH selection attributes to avoid biasing caused due to larger sample values in any attribute.

**Step 2:** The grey relational coefficient is the closeness value between $\mathrm{Y_{ik}}$ and $\mathrm{Y_{0k}}$. Higher coefficient value indicates the closer are two samples. It can be calculated as:

$$\gamma(\mathrm{Y_{0k}}, \mathrm{Y_{ik}}) = \frac{\Delta_{\min} + \mu\, \Delta_{\max}}{\Delta_{\mathrm{ik}} + \mu\, \Delta_{\max}} \tag{4.8}$$

$\gamma$ is the grey relational coefficient between $\mathrm{Y_{0k}}$ and $\mathrm{Y_{ik}}$.

Here $\Delta_{\mathrm{ik}} = |\mathrm{Y_{0k}} - \mathrm{Y_{ik}}|$, $\Delta_{\min} = \min \Delta_{\mathrm{ik}}$ , and $\Delta_{\max} = \max \Delta_{\mathrm{ik}}$ . $\mu$ is the distinguishing coefficient and randomly lies in between 0 and 1. It regulates the expansion and compression of relational coefficient. Using Equation (4.8) coefficient, grey relational coefficient is calculated which is used to select the higher relational samples.

**Step 3:** In our case, the output from Equation (4.8) is used to generate vehicle grade $\mathrm{g(v)}$. The vehicle with the highest $\mathrm{g(v)}$ among other vehicles in the cluster is considered as the head of that cluster. The $\mathrm{g(v)}$ is obtained using [105]:

$$\mathrm{g(v)} = \sum_{k=1}^{n} w_k \gamma(Y_{0k}, Y_{ik}) \qquad \text{for } i = 1,2, \ldots \ldots, Vehi_{num} \tag{4.9}$$

**Algorithm 4.3: Pseudo Code for CH selection Using GRA Model**

**Input :** Number of clusters , Number of vehicles in each clusters at each instant of time $Vehi_{num}$, vehicles coordinators $(X, Y)$, current Lane length (L), current direction (θ)

For t= 1 : instant of time
- For i=1: clusters
  - For j=1: vehicles in cluster
    - Calculate the CH parameters: $\eta$ , $\psi_{vehi}$, and $\mathfrak{E}$
    - Predicted the next vehicle's coordinators (X', Y')
    - Find the new vehicle's direction from predicted coordinators $\theta'$
    - Calculate $T'_{leave}$ using predicted vehicle's position
  - End for
  - Input the four matrices (attributes) to GRA model with their weights, where $\sum_{k=1}^{n} w_k = 1$
  - Generate a matrix $X$ of $Vehi_{num} \times 4$
  - Normalize the matrix X
    - For $\mathrm{i} = 1{:}\,Vehi_{num}$
      - a. For $\mathrm{k} = 1{:}\,\mathrm{n}$
        - Calculate the grey relational coefficient $\gamma(\mathrm{Y_{0k}}, \mathrm{Y_{ik}})$
      - b. end for
    - end for
  - Generate a graph object from grey relational coefficient $\gamma(\mathrm{Y_{0k}}, \mathrm{Y_{ik}})$
  - Calculate the Vehicle's score $g(v)$ using Equation (4.9).
    - Highest-score vehicle is selected as a CH
      $$CH = \max g(v)$$
- End for

End for

**Output:** CH vehicle

Where $n = 4$ (the number of CH selection parameters in our work), $i$ is a number of alternatives. $\gamma(Y_{0k}, Y_{ik})$ is the grey relational coefficient, how near $Y_{ik}$ is to $Y_{0k}$ is determined using it. $w_k$ is the weight of the four parameters, where $\sum_{k=1}^{n} w_k = 1$.

The grade for each vehicle in a cluster is calculated using the GRA. At any given time $t$, the vehicle with the maximum relational analysis ($g(v)$) is assigned as the head for each cluster. The pseudo-code for the CH selection phase is introduced in Algorithm 4.3.

### 4.2.4 Time Complexity of EtHgSC Scheme

The total time complexity of the EtHgSC scheme is expressed as in Equation (3.21) in Chapter 3:

$$O_{TOT} = O_{CF} + O_{RSU} + O_{CH}$$

The major steps involved of the cluster formation are the construction of a hypergraph, Laplacian followed by eigenproblem solving using Eigen-trick method, and then performing the K-means to get the optimal set of clusters.

In hypergraph, the quotient of similarity is taken between each of the vehicles this is calculated by constructing a $d$ nearest neighbour graph with complexity $O(N^2)d$. By using Eigen-trick method, the hyperedge Laplacian $L_e$ is processed instead of vertex Laplacian $L_v$, the time complexity for $L_e$ is $O(NNZ(I^2)/N)$.

“Eigen-trick” provides us a way to calculate $U_v$ with lower time and space complexity, which is demonstrated and compared with Algorithm 3.2 in Chapter 3. The time complexity for solving the eigenproblem of the vertex is $O(N^3)$, while the corresponding cost to hyperedge is $O(m^3)$. Instead of solving the eigenproblem of $L_v$ directly, the Eigen-trick method is proposed to solve it using the eigenproblem of $L_e$. It will reduce the computational complexity from $O(N^3)$, to $O(m^3)$ significantly ($m < N$). Another benefit is that the conversion between $U_e$ and $U_v$ is accurate without any information loss [100]. The last is the K-Means complexity which is dependent on the $O_{C_{optimal}} = O(\tau N C_{num})$. So, $O_{CF}$ will be:

$$O_{CF} = O(N^2)d + O(NNZ(I^2)/N) + O(m^3) + O(\tau N C_{num}) \tag{4.10}$$

The complexity is reduced by removing the terms of less computational power

$$O_{CF} = O(N^2)d + O(m^3) \tag{4.11}$$

The RSU deployment is done using a graph, so the computational complexity is as in Equation (3.24) in Chapter 3.

$$O_{RSU} = O(N^2)d$$

The CH selection parameters are Relative Speed score ( $\psi_{\mathrm{vehi}}$), Neighbourhood Degree ($\eta$), Eccentricity ($\mathfrak{E}$). Time to leave Estimation ($\mathrm{T'}_{\mathrm{leave}}$).

$$O_{CH} = O_{\psi_{\mathrm{vehi}}} + O_{\eta} + O_{\mathfrak{E}} + O_{\mathrm{T'}_{\mathrm{leave}}} \tag{4.12}$$

The time complexity for $\psi_{vehi}$, $\eta$, and $\mathfrak{E}$ are calculated as in Equations (3.26), (3.27), and (3.28) respectively.

$$O_{\psi_{\mathrm{vehi}}} = O(N)$$

$$O_{\eta} = O\ (\log N^2)$$

$$O_{\mathfrak{E}} = O(N^2)d + O\ (N^3)$$

The complexity of time to leave is calculated for all the vehicles in the network

$$O_{\mathrm{T'}_{\mathrm{leave}}} = \mathrm{O}(N) \tag{4.13}$$

Thus, the complete time complexity is reduced to moving all the terms with less complexity than cubic and quadratic terms

$$O_{CH} = O(\log(N^2)) + O(N^3) \tag{4.14}$$

The overall complexity is primarily dependent on the hypergraph

$$O_{TOT} = 2O(N^2)d + O(m^3) + O(\log(N^2)) + O(N^3) \tag{4.15}$$

## 4.3 Simulation Results

The simulation is done using MATLAB as a network simulator, SUMO as a traffic simulator, and with the help of TraCI. The area considered for the study is the real map of Iraq's capital, Baghdad. This study's goal is to evaluate cluster stability under various vehicle densities (100 and 1000) and the dynamic scenario. Also, our proposed is tested at different vehicle speeds (10, 15, 20, and 25 m/s).

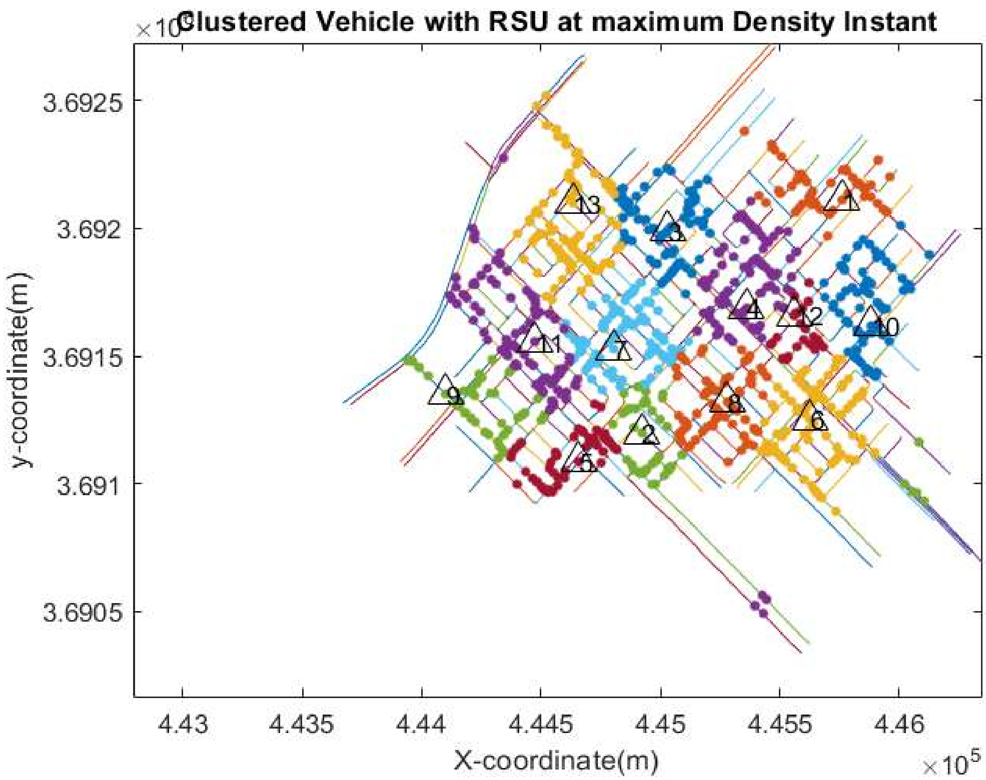



Figure 4.5 Cluster Formation Along with RSU Deployment at an Instant for Our EtHgSC.

Thirteen clusters are generated using Algorithm 4.1. Each cluster is depicted in a different colour. RSUs serve as a cluster centre, they are portrayed in a triangle shape in Figure 4.5. The vehicles are more uniformly clustered in the improved hypergraph as shown in Figure 4.5, whereas the conventional hypergraph has an uneven distribution of the vehicles as shown in Figure 3.8 in Chapter 3.

The results' discussion is carried out in three phases; The effect of the predicted next vehicle's position on the cluster stability at different vehicle speeds; the effect of Eigen-trick method on the clustering stability (EtHgSC comparison with our HGCM presented in Chapter 3 and improved HGCM by applying the Eigen-trick method in terms of stability at low and high traffic); and a state-of-the-art comparison in terms of stability using different cluster performance parameters.

### 4.3.1 Effect of Predicted next Vehicle's Position on the Clustering Stability

The cluster stability is evaluated with some of the cluster performance parameters. Further in this section, CH lifetime, CM lifetime, and CH change rate are evaluated in two cases. These cases are as:

**Case1:** Proposed Scheme with $T'_{leave}$: Evaluation parameters are calculated with the novel improved hypergraph using the Eigen-trick method and a set of CH selection parameters with an approach to predict the vehicle's time to leave $T'_{leave}$ ($\eta$ , $\psi_{vehi}$, $\mathfrak{E}$, and $T'_{leave}$). In this case, the PGRP prediction is used to estimate the future vehicle's direction.

**Case2:** Proposed Scheme with $T_{leave}$: This is the case that uses the improved hypergraph with the same set of CH selection parameters ($\eta$ , $\psi_{vehi}$, $\mathfrak{E}$, and $T_{leave}$). However, the PGRP prediction is not used to estimate the future direction of the vehicle. The time to leave $T_{leave}$ is calculated using the current vehicle's position and direction.

These two tested cases are plotted at different vehicle speeds (10, 15, 20, and 25 m/s). The lifetime of CH and CM should be maximum and the CH change rate should be minimum for the best case. Figure 4.6, Figure 4.7, and Figure 4.8 show the effect of the proposed scheme with $T'_{leave}$ (Case 1) and the effect of the proposed scheme with $T_{leave}$ (Case 2) on the CH lifetime, CM lifetime, and the change rate of CH, respectively.

The CH lifetime, CM lifetime, and CH change rate are tested for the worst-case scenario of 1000 vehicles. We see that our proposed using $T'_{leave}$ with the predicted next position achieves the highest CH lifetime, CM lifetime, and lowest CH change rate particularly at high speed in a comparison with our proposed using $T_{leave}$ without predicated. The use of $T'_{leave}$ with predicated next vehicle's position increases the CH lifetime by 22%, the CM lifetime by 12%, and decreases the CH change rate by 31% at all vehicle speeds in a comparison with our proposed using $T_{leave}$. The change in the vehicle's direction at the road junction hammers the stability. So, the overall stability is enhanced by finding the estimated next vehicle position with the help of PGRP prediction, which helps to select the CH that has a high ability to stay in the cluster as long as possible.

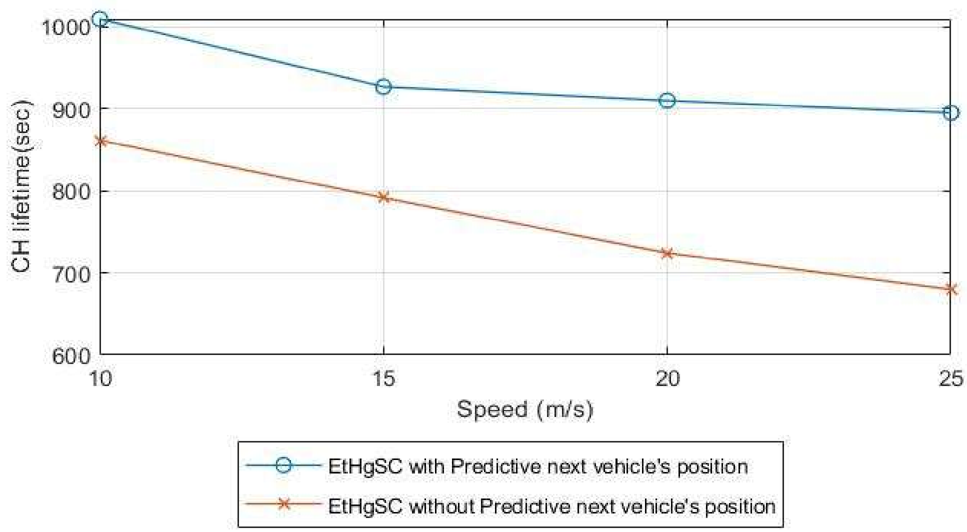



Figure 4.6 CH Lifetimes for EtHgSC with and without Predicted next Vehicle's Position.

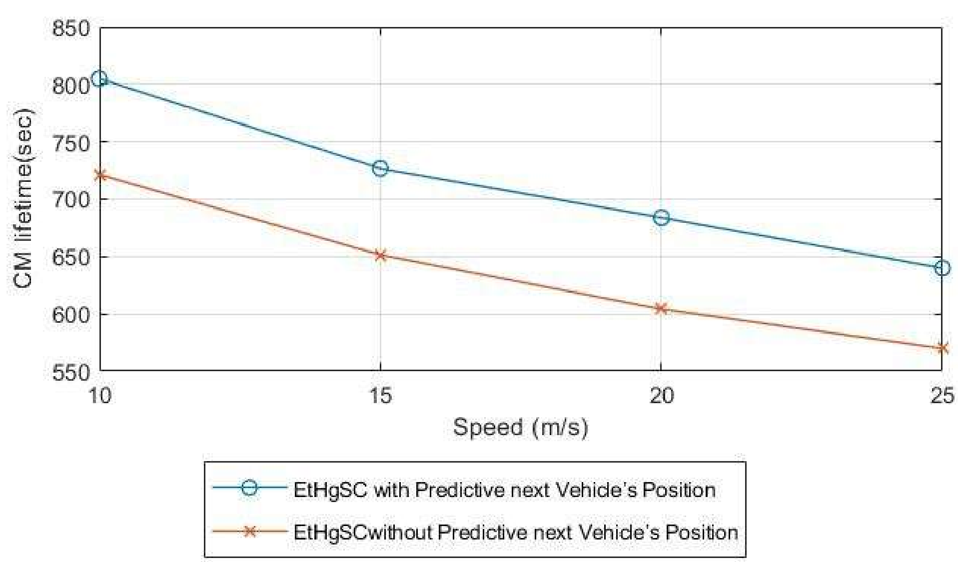



Figure 4.7 CM Lifetimes for EtHgSC with and without Predicted next Vehicle's Position.

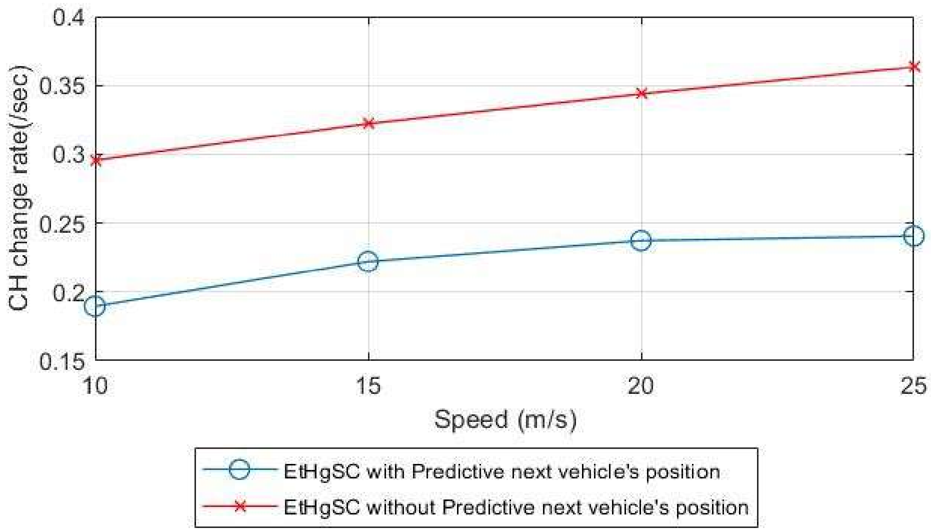



Figure 4.8 CH Change Rate for EtHgSC with and without Predicted next Vehicle's Position.

### 4.3.2 Effect of Eigen-trick Method on the Clustering Stability.

The effect of the Eigen-trick method on the stability is shown in two cases; case 1: Our proposed EtHgSC, and case 2: when Algorithm 4.1 is applied in the cluster formation phase of HGCM instead of Algorithm 3.2 in Chapter 3, we named it as Improved HGCM.

HGCM, Improved HGCM, and EtHgSC are compared at a maximum speed of vehicles at 25 m/s and at low and high vehicular density (100, 1000). When Algorithm 4.1 is used instead of Algorithm 3.2 in Chapter 3, we find that there is a noticeable improvement in the cluster stability in terms of CH lifetime, CM lifetime, and the change rate of CH in a comparison with the original HGCM, see Figure 4.9, Figure 4.10, and Figure 4.11.

The cluster stability of Improved HGCM is enhanced, especially at high densities, with approximately 6% of CH lifetime, and the change rate of CH is reduced by approximately 6% in a comparison with original HGCM. The CM lifetime is also enhanced, but not too much. This improvement is due to the effectiveness of the Eigen-trick method on the hypergraph algorithm, which increases the clustering efficiency.

Moreover, our EtHgSC achieves more improvement than the Improved HGCM with approximately 11% and 6% of CH lifetime and reduces the change rate of CH by 28% and 25% at low and high traffics, respectively. Also, there is a considerable improvement in the CM lifetime with approximately 2.5% at all traffics. This improvement in our EtHgSC is due to the effectiveness of cluster formation algorithm and the CH selection scheme using efficient measures with the help of GRA model.

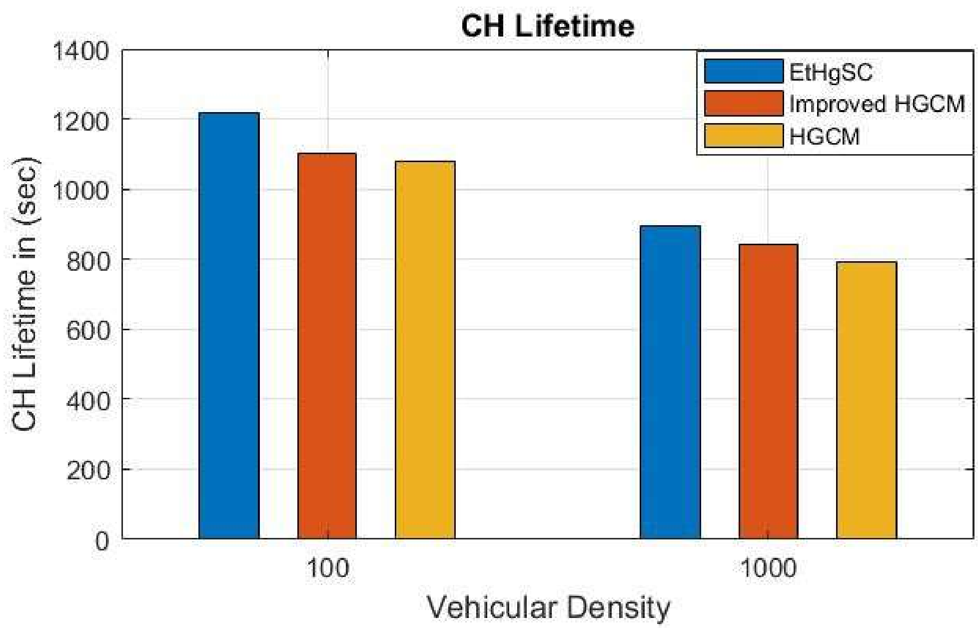



Figure 4.9 CH lifetime for EtHgSC in a Comparison with HGCM and Improved HGCM.

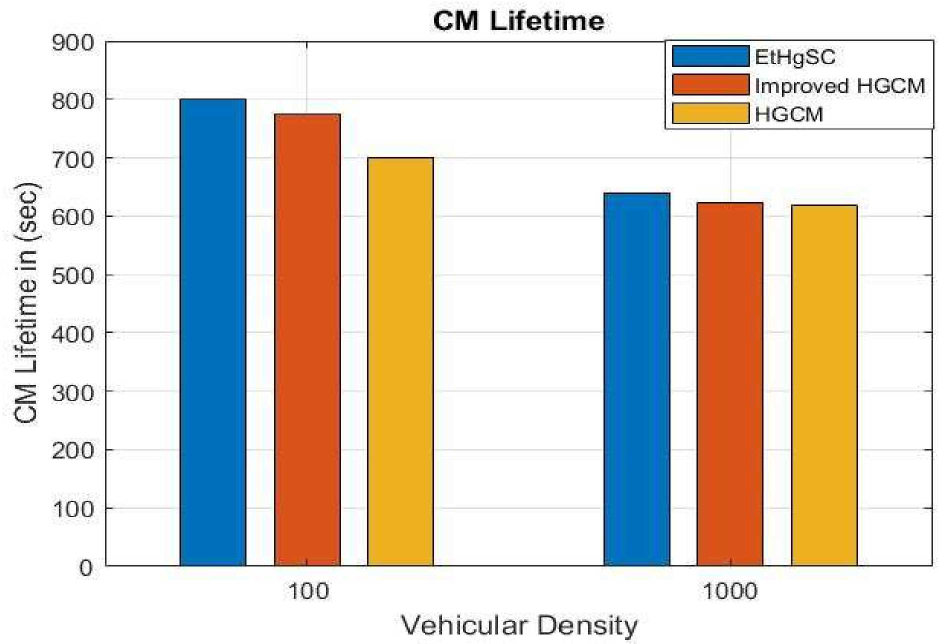



Figure 4.10 CM lifetime for EtHgSC in a Comparison with HGCM and Improved HGCM.

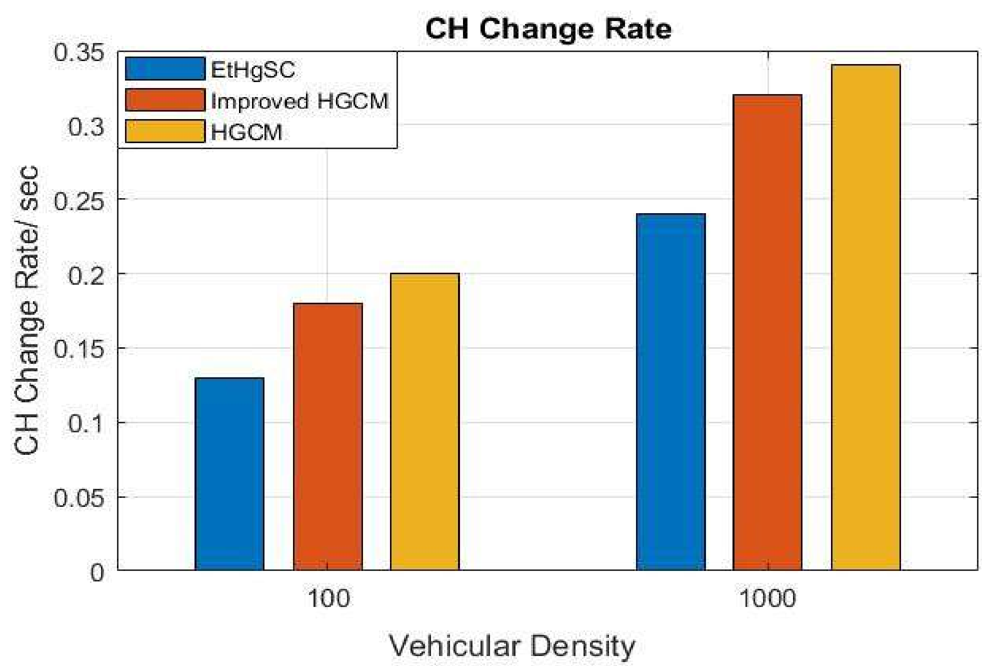



Figure 4.11 CH change rate for EtHgSC in a Comparison with HGCM and Improved HGCM

### 4.3.3 State-of-the-Art Comparison

A comparison of EtHgSC with some algorithms presented in the literature in terms of cluster number and Av. CH lifetime is tabulated in Table 4.1. In addition to the CVoEG and Arkian et al. [35] algorithms which we have compared the HGCM with them in Chapter 3, we also compared with VMaSC [39] and PMC [47] algorithms which have the highest citations in the literature of 365 and 221, respectively. We also compared our EtHgSC with the JCV in [68], which highlighted the same cluster stability problem at junctions. This comparison occurs at low traffic density (100 vehicles), and the maximum vehicles speed is set to 25 m/s, as well as $R_{vehi}$ is set to 200 m for all algorithms.

As mentioned before, the effectiveness of the designed algorithm was also gazed by the number of clusters formed over time. These numbers allow us to evaluate the quality of the formed clusters. Few clusters with low-mobility vehicles achieve efficient connections and stable clustering. On the contrary, more clusters eventually lead to high overhead and mergers. At low and high traffic levels, the EtHgSC constructs 5 and 13 clusters, respectively.

Table 4.1 Comparison with Different Algorithms at Low Traffic Density.

| Algorithm | Av.CH Lifetime | Cluster Number |
|---|---|---|
| **EtHgSC** | 81% | 5 |
| **JCV** | 76% | 16 |
| **CVoEG** | 65.5% | 20 |
| **Arkian et al.** [35] | 58% | 55 |
| **VMaSC** | 66% | 17 |
| **PMC** | 41% | - |

CVoEG was introduced by Khan et al. in [58]. They used a graph spectral clustering algorithm and tested it on a highway network. According to the I-5 highway analysis of the California environment, CVoEG [58] produces 20 clusters with low traffic density along a road length of 12 km. It is expected to achieve 65.5% stability. In this study, the speed of vehicles is used to emulate graph edges. Thus, at low variance, as the speed of vehicles is nearly identical, the eigenvalues are almost the same, which eventually leads to low cluster formation.

Our proposed model has raised the concern of the CH stability at the junction, as the vehicle's direction is unpredictable at the road junction. A similar problem is highlighted by Mohammad et al. [68] in their work. In the JCV [68], the vehicle direction and transmission range are considered for the clustering, and the relative position, direction at the junction, time spent on the road, and node's degree are considered as the CH selection parameters. With 16 clusters constructed, the JCV achieves 76% stability

In the method proposed by Arkian et al. [35], for a highway length of 3000 m, a large number of dynamic clusters are projected with a low variance of only 90 vehicles using two-lane analysis. This method uses neighbourhood analysis, thus when there are just 90 vehicles, there must be a lot of clusters to cover all the vehicles in a sparse region. The CH stability is 58% with low traffic flow and large number of clusters.

The average CH lifetime of the VMaSC algorithm is 66%, with 17 clusters generated for 100 vehicles. This algorithm considers the relevant mobility metric as the CH selection factor. Because of the rapid movement of the vehicle nodes in VANET, the duration of each CH is extremely short. So this may cause decreasing the stability.

PMC algorithm achieves 41% of stability. This ratio is the lowest compared with all other algorithms although the CH selection is based on the neighbouring follow strategy. The authors didn't mention the number of constructed clusters, such that Table 4.1 lacks that.

We can see that there is a significant and noticeable effect of the number of constructed clusters on stability; a few clusters' numbers improve the clustering stability. The highest stability is achieved using our EtHgSC followed by JCV. So, we can conclude that the formation of VANET as a hypergraph improves the clustering efficiency compared with other algorithms. Also, the improving hypergraph algorithm using the Eigen-trick method and the CH selection scheme in the EtHgSC enhance the CH stability by approximately 9% compared with the HGCM in Chapter 3 at low traffic.

The results in Figures 4.12 to 4.14 evaluate the clustering stability using three of the most important cluster performance parameters, including average CH lifetime, average CM lifetime, and CH change rate. As well as the average delay is calculated in Figure 4.15. Our EtHgSC is evaluated at different speeds and under low traffic density (100 vehicles) and compared with JCV, VMaSC, and PMC algorithms to show its supremacy.

**Average CH Lifetime:** We can see that the average CH lifetime indicates a decreasing trend as vehicle speed increases. The network's architecture significantly changes as a result of the speed increases, breaking the connection. From Figure 4.12, the average CH lifetime for JCV is higher than that of VMaSC and PMC because the JCV algorithm takes into consideration the change of vehicle route at the junctions when selecting the CH, so it achieves a high CH duration. Our EtHgSC achieves the highest duration time of CH in a comparison with JCV, VMaSC, and PMC. This achievement is due to our intelligent cluster formation algorithm and the CH selection scheme, which helps to select the vehicle that stays in the cluster as long as possible based on different measures. Also, our proposed solved the problem of unpredicted changes for CH at the junctions, which led to an increase in the stability of the CH.

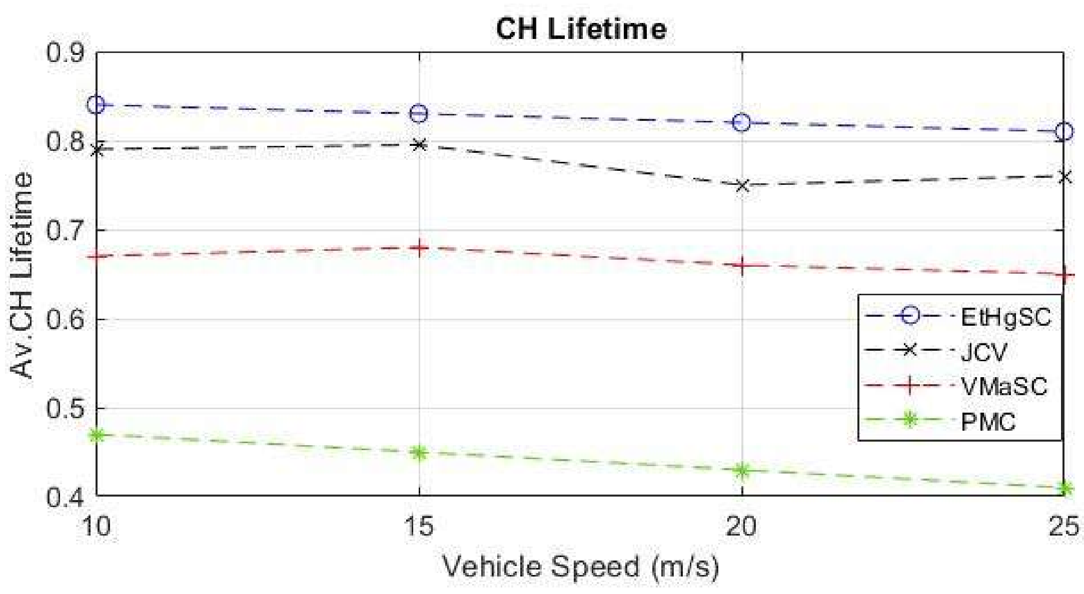



Figure 4.12 Average CH lifetime at different speeds.

**Average CM Lifetime:** Figure 4.13 shows the lifetime of CM for different speeds. The PMC algorithm has a higher CM lifetime than VMaSC, this is due to the use of the following vehicle with the highest priority. However, the average CH movement speed is an issue for VMaSC, so when a cluster's speed increases, joining to other clusters is simple, and therefore the lifetime of CM decreases.

Although the duration of CMs decreases when the speed is increased, our proposed EtHgSC still maintains consistency. The EtHgSC achieves the highest lifetime for CMs of about 10%, 35%, and 86% compared with the JCV, PMC, and VMaSC, respectively, at all vehicle speeds.

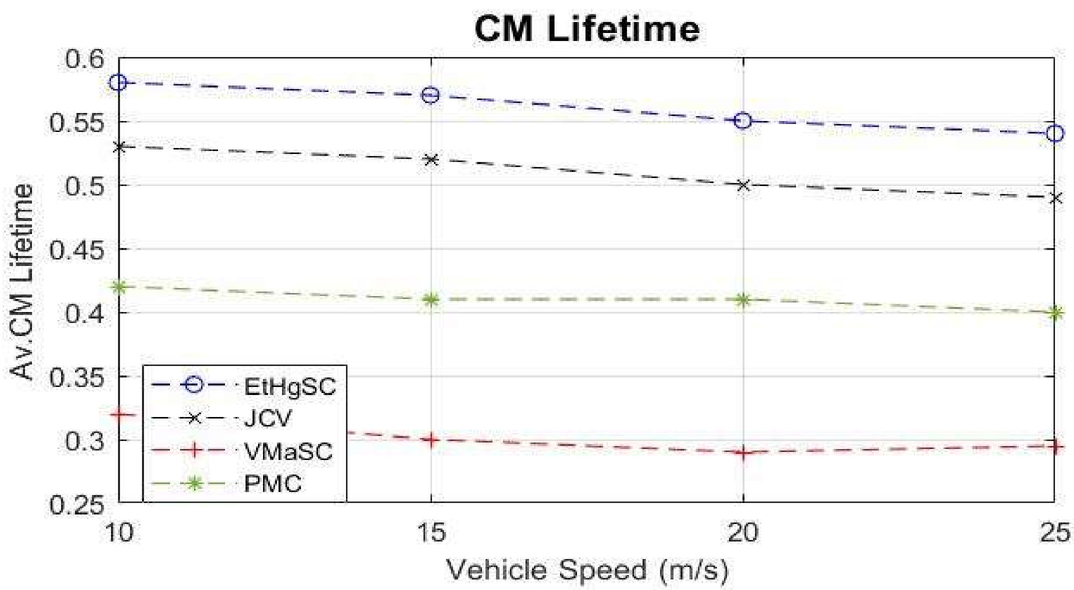



Figure 4.13 Average CM lifetime at different speeds.

**CH Change Rate:** Figure 4.14 shows the relationship between the change rate of CH and the vehicle speed. The results show that the CH change rate increases with an increase in the maximum permitted speed. The CH change rate for our EtHgSC is lower than JCV, VMaSC, and PMC under all conditions. The change rate of CH in our EtHgSC is reduced by 6%, 14%, and 23% compared with JCV, VMaSC, and PMC, respectively, at all vehicle speeds.

**Average Delay:** The average delay is also calculated for our proposed as shown in Figure 4.15; it is the time taken for a packet to transfer from source to destination. The delay is a distance-dependent parameter, and the optimal CH's location helps to reduce the delay, so the EtHgSC achieves less delay, approximately 24%, 66%, and 80% in a comparison with the JCV, PMC, and VMaSC algorithms, respectively.

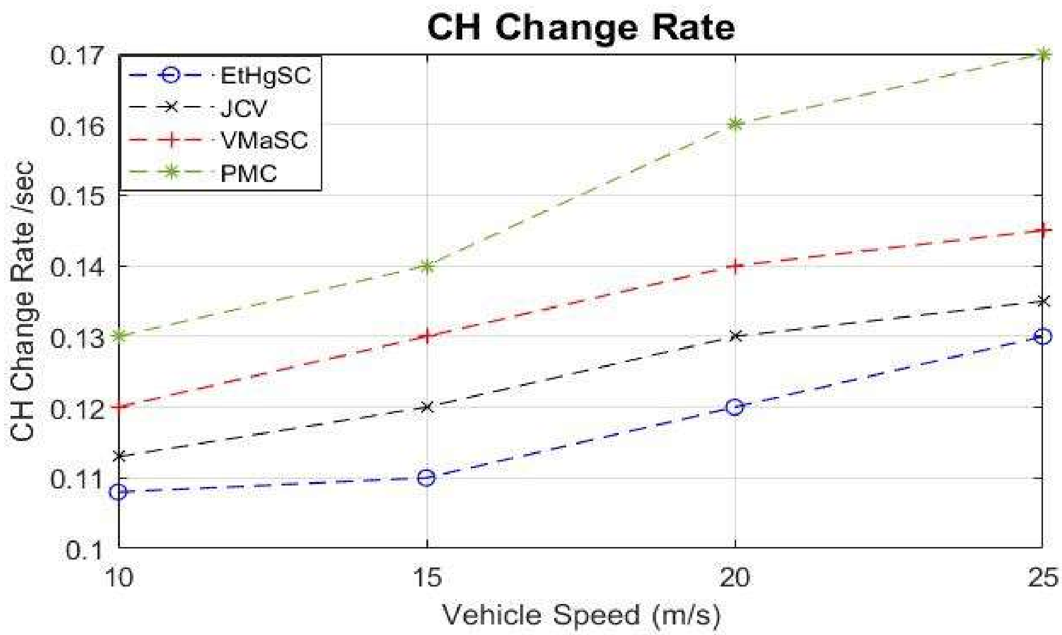



Figure 4.14 CH Change rate at different speeds.

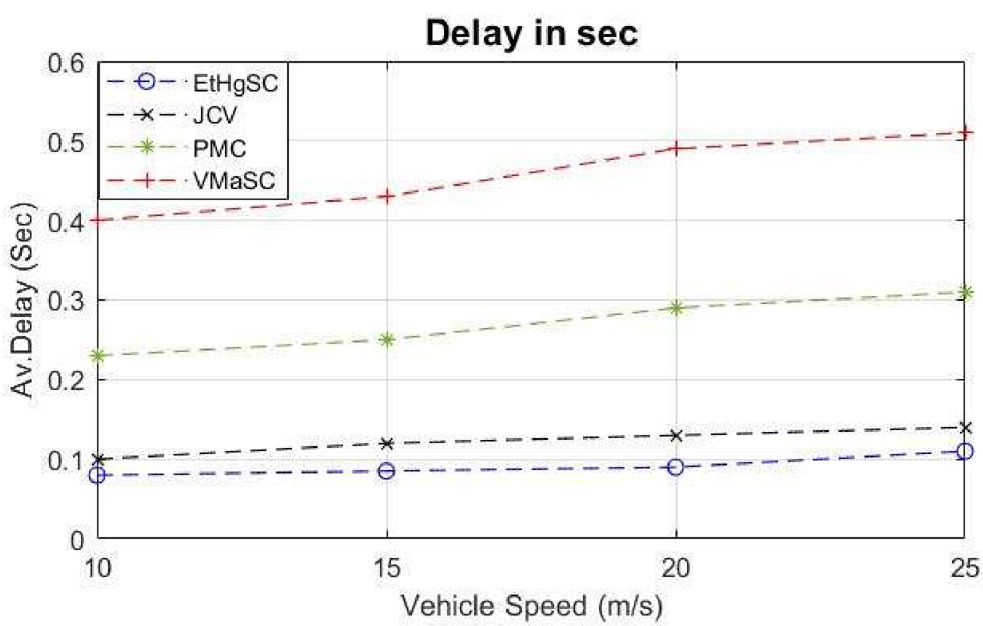



Figure 4.15 Average Delay at different Speeds.

It is also validated from Figures 4.12- 4.15 that as the speed increases, the delay and the change rate of the CH increase, and CH and CM stability decrease. This also proves the convention that the fast-moving vehicle is less reliable to be selected as CH.

Finally, our EtHgSC achieves the highest stability in comparison with other clustering algorithms; this is because of the Eigen-trick method, which is used to improve the hypergraph algorithm and make clustering more efficient. The scheme of CH selection using the four measures and the GRA model is also a reason why stability is getting better.

## 4.4 Conclusion

In this chapter, an Eigen trick-based stable clustering approach in VANET is introduced, with the goal of increasing the clustering stability through efficient cluster formation and CH selection methods. The whole concept works on two levels: cluster generation and CH selection strategy.

The cluster formation has been handled using a hypergraph-based spectral clustering algorithm, which has been introduced in the previous chapter. The hypergraph algorithm has been improved in this chapter using the Eigen-trick method; this method is used to partition both vertices and hyperedges, which provides an approach for reducing the computational complexity of the clustering. The hypergraph is further optimised to explore the hidden sparsity through the removal of the lower non-decomposable graphs. Further, the information loss can be minimised further with the help of the Eigen-trick in the tensor trace maximisation hypergraph partitioning.

Four different parameters have been recommended in this proposed to find the most stable CH in each cluster at each instant of time. In addition to relative speed, neighbour degree, and eccentricity, the vehicle’s time to leave estimation parameter has been presented to select the CH. A relational analysis is developed using grey relational analysis between the four CH selection parameters instead of knowledge-based weightage. The stability is highly hampered at the change in the vehicle’s direction at the road junction; this also has been improved using our efficient CH selection scheme, which enhances the overall stability by avoiding the frequent cluster breakage at the junctions.

The proposed method has been tested in a real area in Iraq's capital, Baghdad. Better stability has been achieved using our proposed scheme, tested at low and high traffic levels and under different vehicle speeds. The results show the supremacy of our proposed in terms of cluster stability, also our proposed scheme achieves a considerable reduction in terms of delay in a comparison with our previous proposed and other techniques.

***Published Paper***

1. M. K. Jabbar and H. Trabelsi, "EtHgSC: Eigen Trick-Based Hypergraph Stable Clustering Algorithm in VANET," *J. Electr. Comput. Eng.*, vol. 2023, p. 6327247, 2023, doi: 10.1155/2023/6327247.

.

# Chapter 5

# Conclusion and Future Works

## 5.1 Conclusion

A new future for dynamic information exchange between cultures is represented by vehicular ad hoc network (VANET). VANET has a wide range of applications in a variety of aspects, including intelligent transportation systems (ITS).

An overview of VANET, its characteristics, and its challenges have been presented in this thesis. Also, VANET communication modes, technologies, and the most important VANET applications have been discussed.

Due to VANET characteristics and challenges, effective clustering techniques are required in order to tackle the network scalability issue and support additional VANETs applications. Also, clustering methods are necessary to ensure the stability of clusters because VANETs' network architecture is dynamic.

In Chapter 2, we have looked at the current clustering techniques used in VANETs. An intensive survey of 55 clustering algorithms has been observed and studied from 2008 to 2022. These algorithms have been summarised and classified in terms of clustering procedures: cluster formation, cluster head (CH) selection, and cluster maintenance. These algorithms have been compared using different parameters. Also, some of the most common metrics used for evaluating the performance of clustering algorithms and simulator tools have been presented.

From the literature survey in Chapter 2, many issues and gaps have been noted, so based on these gaps our proposed solution has been presented in Chapter 3.  Our proposed is named a HyperGraph Clustering Model (HGCM), and its goal is to increase the clustering stability. In the HGCM, clusters have been formed through the designed vehicular-hypergraph-based spectral clustering algorithm using the tensor trace maximisation method (TTM). Different metrics have been introduced for selecting the CH and using the cumulative multimeric scheme to increase the clustering stability. For CH selection, trust metric with deep learning spectrum sensing has been introduced. Deep learning-trained spectrum sensing has been used as a model for trust calculation. Long short term memory (LSTM) layers are used to recognise the primary vehicle in noisy and noiseless environments. A high trust score is awarded to the vehicle which vacates the spectrum in the sensing of the primary vehicle. The addition of the trust element results in 20%

gain in average CH stability over the combined performance of three existing measures (i.e. relative speed, eccentricity, and neighbourhood). The proposed HGCM has been tested for various vehicle densities in a real area of Iraq's capital, Baghdad. Compared with individual measures and other techniques, our cumulative approach significantly improved the clustering stability as well as improved the network performance in terms of packet delay and throughput.

To increase the clustering efficiency, the hypergraph spectral clustering algorithm in Chapter 3 has been improved using the Eigen-trick method. The Eigen-trick method improves the clustering efficiency by using higher-order information in eigenvalues. The Eigen-trick has been used to calculate the modified Laplacian value in the TTM to improve the clustering. A new proposed scheme has been introduced by using this improved algorithm in the cluster formation phase in Chapter 4. The proposed scheme is named Eigen trick-base Hypergraph Stable Clustering (EtHgSC). In the CH selection phase, In addition to relative speed, neighbour degree, and eccentricity, the vehicle's time to leave estimation parameter has been presented to increase the CH stability. A relational analysis has been developed using grey relational analysis among the four CH selection parameters instead of knowledge-based weightage. The selected CH using this scheme maintains strong connectivity and a stable link lifetime.

Our EtHgSC proposed has been applied to an urban scenario at low and high traffic levels and under different vehicle speeds. The stability is highly hampered at the change in the vehicle's direction at the road junction, this also has been improved which enhances the overall stability by finding the estimated next vehicle position with the help of the predictive directional greedy routing protocol (PGRP). In the CH selection phase, the vehicle's time to leave has been calculated in two cases; by using PGRP prediction to estimate the future vehicle's direction to get the time to leave, and without estimating the next vehicle's position (using current vehicle's position). The results showed how the estimated time to leave affected increasing the clustering stability by about 22% of CH lifetime and 12% of CM lifetime, and decreasing the CH change rate by 31% in a comparison with the second case.

Also, the effect of the improved hypergraph algorithm has been tested on the cluster formation phase of the first proposed in Chapter 3. The results showed the efficiency of this improved algorithm with different CH selection schemes and how it helps to increase the stability.

Also, the results showed that our EtHgSC method has better performance than state-of-the-art methods in terms of cluster stability and delay.

## 5.2 Future Works

In future work, we will attempt to generate a more efficient clustering algorithm using the other hypergraph methods like the multilevel recursive graph partition method.

In order to select the more stable CH, especially at high vehicle densities, we intend to explore more metrics for analysis, through which the proposed methodology can be understood.

With the development of LTE and fifth-generation (5G) and other cellular network technologies, many hybrid network architectures for vehicle networks have been proposed. Clustering algorithms are becoming more and more important in these hybrid network architectures, and they are highly recommended to support vehicle services and to increase the efficiency of information delivery with minimum delay.

Also, due to the high level of collaboration required by clustering algorithms, vehicles must share their private information with their neighbours. In this instance, there is still a concern about how to achieve a balance between privacy and collaboration. Moreover, not all of the sharing information is convinced enough, therefore, during the cooperation process, it is important to know how to recognise malicious vehicles.

The performance of the clustering algorithms may be impacted when taking into account the privacy and cooperative security of the vehicles. For example, there may be a delay in the transmission of information when a malicious vehicle is detected. Therefore, how to balance the clustering performance and the clustering security is still an open challenge. So, we intend to generate a more stable and secure clustering technique.

Also, we will try to develop routing protocols for our models (HGCM and EtHgSC) and compare the performance with the existing clustering protocols.

# References


[1] F. Li and Y. Wang, "Routing in vehicular ad hoc networks: A survey," *IEEE Veh. Technol. Mag.*, vol. 2, no. 2, pp. 12–22, 2007.

[2] A. Singh, M. Kumar, R. Rishi, and D. K. Madan, "A relative study of MANET and VANET: Its applications, broadcasting approaches and challenging issues," in *International Conference on Computer Science and Information Technology*, 2011, pp. 627–632.

[3] F. Arena, G. Pau, and A. Severino, "A review on IEEE 802.11p for intelligent transportation systems," *J. Sens. Actuator Networks*, vol. 9, no. 2, 2020, doi: 10.3390/jsan9020022.

[4] M. K. Jabbar and H. Trabelsi, "Clustering Review in Vehicular Ad hoc Networks: Algorithms, Comparisons, Challenges and Solutions.," *Int. J. Interact. Mob. Technol.*, vol. 16, no. 10, 2022.

[5] M. K. J. Alsabah, H. Trabelsi, and W. Jerbi, "Survey on Clustering in VANET Networks," in *2021 18th International Multi-Conference on Systems, Signals & Devices (SSD)*, 2021, pp. 493–502.

[6] S. Baras, I. Saeed, H. A. Tabaza, and M. Elhadef, "VANETs-Based Intelligent Transportation Systems: A Survey."

[7] M. S. Talib, A. Hassan, B. Hussin, and A. A. Hassan, "Vehicular ad-hoc networks: Current challenges and future direction of research," *Jour Adv Res. Dyn. Control Syst*, vol. 10, no. 2, pp. 2065–2074, 2018.

[8] H. Moustafa and Y. Zhang, *Vehicular networks: techniques, standards, and applications*. Auerbach publications, 2009.

[9] J. B. Kenney, "Dedicated short-range communications (DSRC) standards in the United States," *Proc. IEEE*, vol. 99, no. 7, pp. 1162–1182, 2011.

[10] Y. Shi, “An efficient cluster-based service model for vehicular ad-hoc networks on motorways.” Aston University, 2017.

[11] J. Liu, J. Wan, Q. Wang, P. Deng, K. Zhou, and Y. Qiao, “A survey on position-based routing for vehicular ad hoc networks,” *Telecommun. Syst.*, vol. 62, no. 1, pp. 15–30, 2016.

[12] M. M. A. Al-shareeda, M. Anbar, M. A. Alazzawi, S. Manickam, and I. H. Hasbullah, “Security schemes based conditional privacy-preserving in vehicular ad hoc networks,” *Indones. J. Electr. Eng. Comput. Sci.*, vol. 21, no. 1, 2020.

[13] M. Ren, J. Zhang, L. Khoukhi, H. Labiod, and V. Vèque, “A review of clustering algorithms in VANETs,” *Ann. Telecommun.*, vol. 76, no. 9, pp. 581–603, 2021.

[14] W. Li, “Network clustering in vehicular communication networks.” 2011.

[15] P. Basu, N. Khan, and T. D. C. Little, “A mobility based metric for clustering in mobile ad hoc networks,” in *Proceedings 21st international conference on distributed computing systems workshops*, 2001, pp. 413–418.

[16] M. Chatterjee, S. K. Das, and D. Turgut, “WCA: A weighted clustering algorithm for mobile ad hoc networks,” *Cluster Comput.*, vol. 5, no. 2, pp. 193–204, 2002.

[17] R. Ghosh and S. Basagni, “Mitigating the impact of node mobility on ad hoc clustering,” *Wirel. Commun. Mob. Comput.*, vol. 8, no. 3, pp. 295–308, 2008.

[18] Z. Wang, L. Liu, M. Zhou, and N. Ansari, “A position-based clustering technique for ad hoc intervehicle communication,” *IEEE Trans. Syst. Man, Cybern. Part C (Applications Rev.*, vol. 38, no. 2, pp. 201–208, 2008.

[19] R. T. Goonewardene, F. H. Ali, and E. Stipidis, “Robust mobility adaptive clustering scheme with support for geographic routing for vehicular ad hoc networks,” *IET Intell. Transp. Syst.*, vol. 3, no. 2, pp. 148–158, 2009.

[20] C. Shea, B. Hassanabadi, and S. Valaee, “Mobility-based clustering in VANETs using affinity propagation,” in *GLOBECOM 2009-2009 IEEE Global Telecommunications*

*Conference*, 2009, pp. 1–6.

[21] S. Kuklinski and G. Wolny, “Density based clustering algorithm for VANETs,” in *2009 5th international conference on testbeds and research infrastructures for the development of networks & communities and workshops*, 2009, pp. 1–6.

[22] E. Souza, I. Nikolaidis, and P. Gburzynski, “A new aggregate local mobility (ALM) clustering algorithm for VANETs,” in *2010 IEEE International Conference on Communications*, 2010, pp. 1–5.

[23] T. Song, W. Xia, T. Song, and L. Shen, “A cluster-based directional routing protocol in VANET,” in *2010 ieee 12th international conference on communication technology*, 2010, pp. 1172–1175.

[24] A. Daeinabi, A. G. P. Rahbar, and A. Khademzadeh, “VWCA: An efficient clustering algorithm in vehicular ad hoc networks,” *J. Netw. Comput. Appl.*, vol. 34, no. 1, pp. 207–222, 2011.

[25] A. Koulakezian, “Aspire: Adaptive service provider infrastructure for vanets.” 2011.

[26] Z. Zhang, A. Boukerche, and R. Pazzi, “A novel multi-hop clustering scheme for vehicular ad-hoc networks,” in *Proceedings of the 9th ACM international symposium on Mobility management and wireless access*, 2011, pp. 19–26.

[27] L. A. Maglaras and D. Katsaros, “Distributed clustering in vehicular networks,” in *2012 IEEE 8th international conference on wireless and mobile computing, networking and communications (WiMob)*, 2012, pp. 593–599.

[28] A. Ahizoune and A. Hafid, “A new stability based clustering algorithm (SBCA) for VANETs,” in *37th Annual IEEE conference on local computer networks-workshops*, 2012, pp. 843–847.

[29] Z. Y. Rawashdeh and S. M. Mahmud, “A novel algorithm to form stable clusters in vehicular ad hoc networks on highways,” *Eurasip J. Wirel. Commun. Netw.*, vol. 2012, no. 1, pp. 1–13, 2012.

[30] K. A. Hafeez, L. Zhao, Z. Liao, and B. N.-W. Ma, “A fuzzy-logic-based cluster head selection algorithm in VANETs,” in *2012 IEEE international conference on communications (ICC)*, 2012, pp. 203–207.

[31] M. M. C. Morales, C. S. Hong, and Y.-C. Bang, “An adaptable mobility-aware clustering algorithm in vehicular networks,” in *2011 13th asia-pacific network operations and management symposium*, 2011, pp. 1–6.

[32] R. R. Sahoo, R. Panda, D. K. Behera, and M. K. Naskar, “A trust based clustering with Ant Colony Routing in VANET,” in *2012 Third International Conference on Computing, Communication and Networking Technologies (ICCCNT’12)*, 2012, pp. 1–8.

[33] N. Kumar, N. Chilamkurti, and J. H. Park, “ALCA: agent learning–based clustering algorithm in vehicular ad hoc networks,” *Pers. ubiquitous Comput.*, vol. 17, no. 8, pp. 1683–1692, 2013.

[34] S. Ucar, S. C. Ergen, and O. Ozkasap, “VMaSC: Vehicular multi-hop algorithm for stable clustering in vehicular ad hoc networks,” in *2013 IEEE wireless communications and networking conference (WCNC)*, 2013, pp. 2381–2386.

[35] H. R. Arkian, R. E. Atani, A. Pourkhalili, and S. Kamali, “Cluster-based traffic information generalization in vehicular ad-hoc networks,” *Veh. Commun.*, vol. 1, no. 4, pp. 197–207, 2014.

[36] Y. Chen, M. Fang, S. Shi, W. Guo, and X. Zheng, “Distributed multi-hop clustering algorithm for VANETs based on neighborhood follow,” *Eurasip J. Wirel. Commun. Netw.*, vol. 2015, no. 1, pp. 1–12, 2015.

[37] M. Hadded, R. Zagrouba, A. Laouiti, P. Muhlethaler, and L. A. Saidane, “A multi-objective genetic algorithm-based adaptive weighted clustering protocol in vanet,” in *2015 IEEE congress on evolutionary computation (CEC)*, 2015, pp. 994–1002.

[38] J. P. Singh and R. S. Bali, “A hybrid backbone based clustering algorithm for vehicular ad-hoc networks,” *Procedia Comput. Sci.*, vol. 46, pp. 1005–1013, 2015.

[39] S. Ucar, S. C. Ergen, and O. Ozkasap, “Multihop-cluster-based IEEE 802.11 p and LTE

hybrid architecture for VANET safety message dissemination," *IEEE Trans. Veh. Technol.*, vol. 65, no. 4, pp. 2621–2636, 2015.

[40] A. Z. Ahwazi and M. NooriMehr, "MOSIC: mobility-aware single-hop clustering scheme for vehicular ad hoc networks on highways," *Int. J. Adv. Comput. Sci. Appl.*, vol. 7, no. 9, 2016.

[41] S. Harrabi, I. Ben Jaafar, and K. Ghedira, "A Novel Clustering Algorithm Based on Agent Technology for VANET.," *Netw. Protoc. Algorithms*, vol. 8, no. 2, pp. 1–19, 2016.

[42] M. M. Mohammed Nasr, A. M. S. Abdelgader, Z.-G. Wang, and L.-F. Shen, "VANET clustering based routing protocol suitable for deserts," *Sensors*, vol. 16, no. 4, p. 478, 2016.

[43] M. Ren, L. Khoukhi, H. Labiod, J. Zhang, and V. Veque, "A mobility-based scheme for dynamic clustering in vehicular ad-hoc networks (VANETs)," *Veh. Commun.*, vol. 9, pp. 233–241, 2017.

[44] I. Hussain and C. Bingcai, "Cluster formation and cluster head selection approach for vehicle ad-hoc network (VANETs) using K-means and floyd-Warshall technique," *Int. J. Adv. Comput. Sci. Appl.*, vol. 8, no. 12, p. 1, 2017.

[45] A. Abuashour and M. Kadoch, "Performance improvement of cluster-based routing protocol in VANET," *Ieee access*, vol. 5, pp. 15354–15371, 2017.

[46] G. V Rossi, Z. Fan, W. H. Chin, and K. K. Leung, "Stable clustering for ad-hoc vehicle networking," in *2017 ieee wireless communications and networking conference (wcnc)*, 2017, pp. 1–6.

[47] D. Zhang, H. Ge, T. Zhang, Y.-Y. Cui, X. Liu, and G. Mao, "New multi-hop clustering algorithm for vehicular ad hoc networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 4, pp. 1517–1530, 2018.

[48] X. Ji, H. Yu, G. Fan, H. Sun, and L. Chen, "Efficient and reliable cluster-based data transmission for vehicular ad hoc networks," *Mob. Inf. Syst.*, vol. 2018, 2018.

[49] M. Ren, J. Zhang, L. Khoukhi, H. Labiod, and V. Vèque, “A unified framework of clustering approach in vehicular ad hoc networks,” *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 5, pp. 1401–1414, 2017.

[50] A. Abuashour and M. Kadoch, “An enhanced control overhead messages reduction algorithm in VANET,” in *2018 IEEE 6th International Conference on Future Internet of Things and Cloud (FiCloud)*, 2018, pp. 275–281.

[51] T. Koshimizu, H. Wang, Z. Pan, J. Liu, and S. Shimamoto, “Normalized multi-dimensional parameter based affinity propagation clustering for cellular V2X,” in *2018 IEEE Wireless Communications and Networking Conference (WCNC)*, 2018, pp. 1–6.

[52] M. A. Saleem *et al.*, “Expansion of cluster head stability using fuzzy in cognitive radio CR-VANET,” *IEEE Access*, vol. 7, pp. 173185–173195, 2019.

[53] G. H. Alsuhli, A. Khattab, and Y. A. Fahmy, “Double-head clustering for resilient VANETs,” *Wirel. Commun. Mob. Comput.*, vol. 2019, 2019.

[54] A. Bello Tambawal, R. Md Noor, R. Salleh, C. Chembe, and M. Oche, “Enhanced weight-based clustering algorithm to provide reliable delivery for VANET safety applications,” *PLoS One*, vol. 14, no. 4, p. e0214664, 2019.

[55] O. Senouci, S. Harous, and Z. Aliouat, “A new heuristic clustering algorithm based on RSU for internet of vehicles,” *Arab. J. Sci. Eng.*, vol. 44, no. 11, pp. 9735–9753, 2019.

[56] X. Cheng and B. Huang, “A center-based secure and stable clustering algorithm for VANETs on highways,” *Wirel. Commun. Mob. Comput.*, vol. 2019, 2019.

[57] M. Syfullah, J. M.-Y. Lim, and F. L. Siaw, “Mobility-Based Clustering Algorithm for Multimedia Broadcasting over IEEE 802.11 p-LTE-enabled VANET,” *KSII Trans. Internet Inf. Syst.*, vol. 13, no. 3, pp. 1213–1237, 2019.

[58] Z. Khan, P. Fan, S. Fang, and F. Abbas, “An unsupervised cluster-based VANET-oriented evolving graph (CVoEG) model and associated reliable routing scheme,” *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 10, pp. 3844–3859, 2019.

[59] K. Bylykbashi, D. Elmazi, K. Matsuo, M. Ikeda, and L. Barolli, "Effect of security and trustworthiness for a fuzzy cluster management system in VANETs," *Cogn. Syst. Res.*, vol. 55, pp. 153–163, 2019.

[60] S. Haider, G. Abbas, Z. H. Abbas, S. Boudjit, and Z. Halim, "P-DACCA: A probabilistic direction-aware cooperative collision avoidance scheme for VANETs," *Futur. Gener. Comput. Syst.*, vol. 103, pp. 1–17, 2020.

[61] L. Rivoirard, M. Wahl, and P. Sondi, "Multipoint relaying versus chain-branch-leaf clustering performance in optimized link state routing-based vehicular ad hoc networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 3, pp. 1034–1043, 2019.

[62] F. Abbas, G. Liu, Z. Khan, K. Zheng, and P. Fan, "Clustering based resource management scheme for latency and sum rate optimization in V2X networks," in *2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring)*, 2019, pp. 1–6.

[63] G. L. Moore and P. Liu, "A hybrid (active-passive) clustering technique for VANETs," in *2019 IEEE ComSoc International Communications Quality and Reliability Workshop (CQR)*, 2019, pp. 1–6.

[64] W. Ahsan *et al.*, "Optimized node clustering in VANETs by using meta-heuristic algorithms," *Electronics*, vol. 9, no. 3, p. 394, 2020.

[65] M. Bersali, A. Rachedi, and H. Bouarfa, "A new collaborative clustering approach for the Internet of vehicles (CCA-IoV)," in *2020 Second International Conference on Embedded & Distributed Systems (EDiS)*, 2020, pp. 58–63.

[66] A. Jalooli, K. Zhang, M. Song, and W. Wang, "A Novel Clustering Scheme for Heterogeneous Vehicular Networks," in *ICC 2020-2020 IEEE International Conference on Communications (ICC)*, 2020, pp. 1–6.

[67] R. Singh, D. Saluja, and S. Kumar, "Reliability improvement in clustering-based vehicular ad-hoc network," *IEEE Commun. Lett.*, vol. 24, no. 6, pp. 1351–1355, 2020.

[68] M. Mukhtaruzzaman and M. Atiquzzaman, "Junction-based stable clustering algorithm for vehicular ad hoc network," *Ann. Telecommun.*, pp. 1–10, 2021.

[69] U. Maan and Y. Chaba, “Accurate Cluster Head Selection Technique for Software Defined Network in 5G VANET,” *Wirel. Pers. Commun.*, vol. 118, no. 2, pp. 1271–1293, 2021.

[70] Z. Khan, A. Koubaa, S. Fang, M. Y. Lee, and K. Muhammad, “A Connectivity-Based Clustering Scheme for Intelligent Vehicles,” *Appl. Sci.*, vol. 11, no. 5, p. 2413, 2021.

[71] K. Kandali, L. Bennis, and H. Bennis, “A new hybrid routing protocol using a modified K-means clustering algorithm and continuous hopfield network for VANET,” *IEEE Access*, vol. 9, pp. 47169–47183, 2021.

[72] B. Liu *et al.*, “A region-based collaborative management scheme for dynamic clustering in green vanet,” *IEEE Trans. Green Commun. Netw.*, 2022.

[73] S. Vodopivec, J. Bešter, and A. Kos, “A survey on clustering algorithms for vehicular ad-hoc networks,” in *2012 35th international conference on telecommunications and signal processing (TSP)*, 2012, pp. 52–56.

[74] I. A. Aljabry and G. A. Al-Suhail, “A survey on network simulators for vehicular ad-hoc networks (VANETS),” *Int. J. Comput. Appl*, vol. 174, no. 11, pp. 1–9, 2021.

[75] R. Adrian, S. Sulistyo, and I. W. Mustika, “A study on communication system in VANET,” in *2018 4th International Conference on Science and Technology (ICST)*, 2018, pp. 1–6.

[76] K. G. Lim, C. H. Lee, R. K. Y. Chin, K. B. Yeo, and K. T. K. Teo, “Simulators for vehicular ad hoc network (VANET) development,” in *2016 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia)*, 2016, pp. 1–4.

[77] S. Elaraby and S. M. Abuelenin, “Connectivity analysis of directed highway vehicular ad hoc networks using graph theory,” *Int. J. Commun. Syst.*, vol. 34, no. 5, p. e4745, 2021.

[78] S. Fortunato, “Community detection in graphs,” *Phys. Rep.*, vol. 486, no. 3–5, pp. 75–174, 2010.

[79] J. Z. Huang, M. K. Ng, H. Rong, and Z. Li, “Automated variable weighting in k-means

type clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 5, pp. 657–668, 2005.

[80] L. Kaufman and P. J. Rousseeuw, *Finding groups in data: an introduction to cluster analysis*. John Wiley & Sons, 2009.

[81] U. Von Luxburg, "A tutorial on spectral clustering," *Stat. Comput.*, vol. 17, no. 4, pp. 395–416, 2007.

[82] M. Ran and X. Bai, "Vehicle cooperative network model based on hypergraph in vehicular fog computing," *Sensors*, vol. 20, no. 8, p. 2269, 2020.

[83] G. Khayat, C. X. Mavromoustakis, G. Mastorakis, J. M. Batalla, H. Maalouf, and E. Pallis, "VANET clustering based on weighted trusted cluster head selection," in *2020 International Wireless Communications and Mobile Computing (IWCMC)*, 2020, pp. 623–628.

[84] F. Sanhaji, H. Satori, and K. Satori, "Cluster head selection based on neural networks in wireless sensor networks," in *2019 International Conference on Wireless Technologies, Embedded and Intelligent Systems (WITS)*, 2019, pp. 1–5.

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

[86] D. Ghoshdastidar and A. Dukkipati, "Uniform hypergraph partitioning: Provable tensor methods and sampling techniques," *J. Mach. Learn. Res.*, vol. 18, no. 1, pp. 1638–1678, 2017.

[87] D. Ghoshdastidar and A. Dukkipati, "A provable generalized tensor spectral method for uniform hypergraph partitioning," in *International Conference on Machine Learning*, 2015, pp. 400–409.

[88] X. Wang and Y. Xu, "An improved index for clustering validation based on Silhouette index and Calinski-Harabasz index," in *IOP Conference Series: Materials Science and Engineering*, 2019, vol. 569, no. 5, p. 52024.

[89] P. Ahrendt, “The multivariate gaussian probability distribution,” *Tech. Univ. Denmark, Tech. Rep*, 2005.

[90] D. R. White and S. P. Borgatti, “Betweenness centrality measures for directed graphs,” *Soc. Networks*, vol. 16, no. 4, pp. 335–346, 1994, doi: 10.1016/0378-8733(94)90015-9.

[91] J. Cheng, G. Yuan, M. Zhou, S. Gao, Z. Huang, and C. Liu, “A connectivity-prediction-based dynamic clustering model for VANET in an urban scene,” *IEEE Internet Things J.*, vol. 7, no. 9, pp. 8410–8418, 2020.

[92] M. S. Kakkasageri and S. S. Manvi, “Multiagent driven dynamic clustering of vehicles in VANETs,” *J. Netw. Comput. Appl.*, vol. 35, no. 6, pp. 1771–1780, 2012, doi: 10.1016/j.jnca.2012.07.002.

[93] B. Soni, D. K. Patel, and M. Lopez-Benitez, “Long Short-Term Memory Based Spectrum Sensing Scheme for Cognitive Radio Using Primary Activity Statistics,” *IEEE Access*, vol. 8, no. May, pp. 97437–97451, 2020, doi: 10.1109/ACCESS.2020.2995633.

[94] X. Qian and L. Hao, “On the performance of spectrum sensing in cognitive vehicular networks,” in *2015 IEEE 26th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, 2015, pp. 1002–1006.

[95] H. A. Shah and I. Koo, “Reliable machine learning based spectrum sensing in cognitive radio networks,” *Wirel. Commun. Mob. Comput.*, vol. 2018, 2018, doi: 10.1155/2018/5906097.

[96] M. Behrisch, L. Bieker, J. Erdmann, and D. Krajzewicz, “SUMO--simulation of urban mobility: an overview,” *Proc. SIMUL 2011, Third Int. Conf. Adv. Syst. Simul.*, no. May 2014, 2011.

[97] A. Wegener, M. Piórkowski, M. Raya, H. Hellbrück, S. Fischer, and J. P. Hubaux, “TraCI: An interface for coupling road traffic and network simulators,” *Proc. 11th Commun. Netw. Simul. Symp. CNS’08*, no. May 2014, pp. 155–163, 2008, doi: 10.1145/1400713.1400740.

[98] M. Wazid *et al.*, “Design of lightweight authentication and key agreement protocol for

vehicular ad hoc networks," *IEEE Access*, vol. 5, pp. 14966–14980, 2017.

[99] P. S. Chodrow, N. Veldt, and A. R. Benson, "Generative hypergraph clustering: From blockmodels to modularity," *Sci. Adv.*, vol. 7, no. 28, p. eabh1303, 2021.

[100] Y. Yang, S. Deng, J. Lu, Y. Li, Z. Gong, and Z. Hao, "GraphLSHC: towards large scale spectral hypergraph clustering," *Inf. Sci. (Ny).*, vol. 544, pp. 117–134, 2021.

[101] D. Kim, Y. Velasco, W. Wang, R. N. Uma, R. Hussain, and S. Lee, "A new comprehensive RSU installation strategy for cost-efficient VANET deployment," *IEEE Trans. Veh. Technol.*, vol. 66, no. 5, pp. 4200–4211, 2016.

[102] R. Adrian, S. Sulistyo, I. W. Mustika, and S. Alam, "A controllable rsu and vampire moth to support the cluster stability in vanet," *Int. J. Comput. Networks Commun*, vol. 13, no. 3, pp. 79–95, 2021.

[103] R. Karimi and S. Shokrollahi, "PGRP: Predictive geographic routing protocol for VANETs," *Comput. Networks*, vol. 141, pp. 67–81, 2018.

[104] A. Jayant, "Use of grey relational analysis in solving multiple attribute decision-making problem: A case study of warehouse location selection," *Adv. Ind. Eng. Manag.*, vol. 4, no. 2, pp. 157–164, 2015.

[105] Y. Kuo, T. Yang, and G.-W. Huang, "The use of grey relational analysis in solving multiple attribute decision-making problems," *Comput. Ind. Eng.*, vol. 55, no. 1, pp. 80–93, 2008.

# Appendices

## Appendix 1: The Map of Baghdad Loading and Directory Making

```
                    %%%%----------The map loading and directory making------%%
Specify the map to be used
vhcls=100;            %-----> the vehilce density
map=['map_of_bagdad/',num2str(vhcls)];
crntdir=pwd;  % save the current directory path
addpath([crntdir,'\utility'])
addpath([crntdir,'\cluster_generations'])
addpath([crntdir,'\cluster_maintain'])
addpath([crntdir,'\radio network'])

Create directory to store the results
d=num2str(clock);
d(isspace(d))=[];
if ~exist('Results','dir')
   mkdir('Results')
end
dirname=['Results/',map];
mkdir(dirname)
vehicle_transrange=200;   %% transmission range of the each vehicle
rsuTrRnage =350;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

## Appendix 2: Getting the TraCI and Vehicles' Information

```
import traci.constants
Get the filename of the example scenario
% scenarioPath = [scenarioPath '\inter_palmas'];
cd([map]);

SUMO interfacing and recording of vehicle parameters
disp('1. SUMO starting and collecting vehicle information...')
traci.start('sumo-gui -c ./Configuration.sumocfg --start');
traci.gui.setZoom('View #0',350)
traci.gui.setSchema('View #0','real world')
mapBoundary = traci.gui.getBoundary('View #0');
LaneIDlist = traci.lane.getIDList;  % get the lane ids on the map
% screenshot save
% traci.gui.screenshot('View #0', 'Network ScreenShot.bmp')

Recording of vehicle parameters at each simulation time step
disp('The recording of the different parameters of the vehicles');
initialVhclID=[];
initialPos=[];
simTime=1500;
for i = 1: simTime   % simulation time
    traci.simulation.step();
    vhclID=traci.vehicle.getIDList();  % get vehicle's ID on the road

    if ~isempty(vhclID)
        for vhcl=1:numel(vhclID) % loop for all vehicles at any instant on the road
            temp = traci.vehicle.getPosition(vhclID{vhcl}); % measure vehcile position
            locationX(i,vhcl)=temp(1);
            locationY(i,vhcl)=temp(2);
            speedY(i,vhcl) = traci.vehicle.getSpeed(vhclID{vhcl}); % speed
```

```
        laneid{i,vhcl} = traci.vehicle.getLaneID(vhclID{vhcl}); % getes the lane
        diststart(i,vhcl) = traci.vehicle.getDistance(vhclID{vhcl}); % total distance traveled from
begining
        SpeedDeviation(i,vhcl) = traci.vehicle.getSpeedDeviation(vhclID{vhcl}); % speed
change
        vehilceID(i,vhcl) = str2double(vhclID{vhcl})+1; % each vehicle ID
        angle(i,vhcl) = traci.vehicle.getAngle(vhclID{vhcl}); % angle of the vehicle
        trust(i,vhcl)=0;
```

Store the vehicle's ID appearing very first time only with their positions

```
      end
   end
end
% %%%---------------------Final map of the city--------------------------%%

cnt=1;laneShape=[];

for ii=1:numel(LaneIDlist)   % get the lanes ID and their length

   temp = strsplit(LaneIDlist{ii},'-');

   if size(temp)==1

      Singlelane(cnt) = LaneIDlist(ii);  % lane IDs

%        SinglaneLength(cnt) = traci.lane.getLength(LaneIDlist{ii}); % lane length

%        laneShape=[laneShape,(traci.lane.getShape(LaneIDlist{ii}))]; % lane co-ordinates

      cnt=cnt+1;

   end

end
```

## Appendix 3: Obtaining the Adjacency Matrix for Maximum number of Vehicles

```
                    %% find the time slot with the maximum number of vehicles%%
disp(' Step.1 : Obtain the time slot with maximum vehicles');
size_detected=[];
for i=1:size(locationX,1)  %% scan each time slot
    temp_var = numel(find(locationX(i,:))); % number of vehicles at any time.
    size_detected=[size_detected temp_var]; %% store the number vehicles in each time slot
end
[maxvhcl, indx]=max(size_detected);  %% index of the slot with maximum vehicle
disp('Finished.')
disp(' Step.2 : Obtain an adjacency matrix for the time stamp with maximum vehicle');
               %% Calculate the distance of each vehicle to other at that time instant%%
dist_matrix=zeros(maxvhcl);  %% for a adjacency matrix for each vehicle
adajency_matrix = zeros(maxvhcl);
for i=1:maxvhcl  %% number of vehicle
    for k=1:maxvhcl   %% number of vehicle
        dist_matrix(i,k)=sqrt(sum((locationX(indx,i)-locationX(indx,k)).^2+(locationY(indx,i)-
locationY(indx,k)).^2));
        if dist_matrix(i,k)<=vehicle_transrange
            adajency_matrix(i,k)=1;
        else
```

```
            adajency_matrix(i,k)=0;
        end
    end
end
                    %% The hypergraph method for finding optimal number of clusters%%
disp(' Step.3 : The hypergraph method TTM for finding the optimal clsuters');
tic
[evecs,~] = hypergraph(adajency_matrix);  %% hypergraph function
% evecs = diag(evecs);
%% The cluster formation with the ID
disp(' Step.4 : Find the optimal clusters based on the above evaluation');
eva = evalclusters(evecs,'linkage','CalinskiHarabasz','KList',[iniK:K]); % cluster evaluation using
the CalinskiHarabasz
no_clusters=eva.OptimalK;
evecs1 = evecs(1:no_clusters,:)'; % eigen vectors for optimal clusters
N=size(dist_matrix,2);
for i = 1:N      % normalize the eigen vectors before clustering
    if (norm(evecs1(i,:))>0)
        evecs1(i,:) = evecs1(i,:)./norm(evecs1(i,:));
    end
end
```

```
vhcls_inst = [locationX(indx,:)',locationY(indx,:)'];% vehicles coordiantes at clustering time instant

[idx]=kmeans(evecs1,no_clusters);

toc

for i=1:no_clusters   %% no of clusters

%        cluster_bin{i}=find(eva.OptimalY(:,1)==i);  %% no of vehicles in each cluster

        cluster_bin{i}=find(idx==i);  % vehicle's allotment in clusters to display in figure

        binCount(i)  = numel(find(idx==i));

end

if any(binCount<3)% if cluster has less than 3 vehicles, remove that cluster

    removbinIndx = find(binCount<3);

    cluster_bin(removbinIndx)=[];

    no_clusters=no_clusters-numel(removbinIndx);

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

## Appendix 4: Cluster Head Selection Parameters

```
%% Cluster Head selection

%%%--------------------Cluster maintenance Phase--------------------------%

disp(' Step.7. Calculation for CH parameters')

clusterNo=size(placed_RSU_loc,1);    %% Number of clusters

car_RSU_connect = zeros(size(locationX,1),size(locationX,2));
```

```
%% Start the clusterias intial data
car=[];
tic
for tim=20:size(locationX,1)
   %% for each time span
   car=locationX(tim,:);  %% take the vehicles at each time instant
   %% Distance between the RSU and the vehicles
   %disp('3.1 Distance between the car and RSU')

   dist_matrix=zeros(size(locationX,2),clusterNo);  % initialize the distance matrix of vehicle
and RSU
   rsu_car_connect = zeros(size(locationX,2),clusterNo);% initialize the connection matrix of
vehicle and RSU
   for inst=1:size(locationX,2)
      if car(inst)~=0
       for k=1:clusterNo   %% distance between the car and the RSU
           dist_matrix(inst,k)=sqrt(sum((locationX(tim,inst)-
placed_RSU_loc(k,1)).^2+(locationY(tim,inst)-placed_RSU_loc(k,2)).^2));
           if dist_matrix(inst,k)<=rsuTrRnage   % check the vehicle connected to which RSU
               rsu_car_connect(inst,k) = 1;
           else
               rsu_car_connect(inst,k)=0;
```

```
          end
        end
       end
        % if vehicle is connected to more than 1 RSU, select the nearest
        % one
        if numel(find(rsu_car_connect(inst,:)))>1
           [~,ind] = min(dist_matrix(inst,:));  % indx of minimum distant RSU
           car_RSU_connect(tim,inst)=ind;       % variable that saves the car connected to RSU a
each time instant
        end
    end

    %% Start the parameter analysis
    for k=1:clusterNo  % loop for each cluster
        no_car = numel(find(rsu_car_connect(:,k)));% cars in each cluster
        if ~(no_car==0)
           if no_car<3      % if there is only one car in the cluster
%                disp('Less than 3 cars in cluster; so exiting the loop')
           else
               %average speed calculation in each cluster
              cluster_average_speed = mean(speedY(tim,find(rsu_car_connect(:,k))));
```

```
%% ===== Calculation for the eccentricity ============

% calculate the distance on each vehicle to other in that

% cluster

vhclsind_Cluster = find(rsu_car_connect(:,k));  %get the index of vehilces

vhclsIDinCluster = vehilceID(tim,vhclsind_Cluster); % vehicles id in each cluster

 TotalCM_old(k,tim) = numel(vehilceID(tim,vhclsind_Cluster)); % Total cluster
members in each cluster at every simulation time

% CM ID

CMID_old{k,tim} = vhclsIDinCluster;

cluster_CHeccencity=[];

[cluster_CHeccencity,nbrsDegree]=eccenctricity_new(tim,vhclsind_Cluster,locationX,locationY
,vehicle_transrange); % call eccentricity function



%% ===== Calculation of Vehicle s Relative Speed ============

sthr=2.77;   % the speed threshold

delta=0.01;    %% delta update in speed



Rel_speedDiff=speedY(tim,find(rsu_car_connect(:,k)))-cluster_average_speed;
%relative speed of each vehicle

spedInd = Rel_speedDiff < sthr ;   % index of vehicle with relative speed lesser than
threshold

Rel_speedDiff = Rel_speedDiff+delta.*ones(1,numel(spedInd)).*spedInd;
```

```
spedInd_1 = Rel_speedDiff > sthr ;   % index of vehicle with relative speed higher than threshold
Rel_speedDiff = Rel_speedDiff-delta.*ones(1,numel(spedInd_1)).*spedInd_1;
cluster_CHavrege_speed = Rel_speedDiff;

%====== calculation of neighborhood degree
cluster_CHneighbourDegree = nbrsDegree;

%====== total score calculation
cluster_CHtotal=sum([cluster_CHneighbourDegree',...
              cluster_CHtrust',....
              cluster_CHavrege_speed',...
              cluster_CHeccencity']);

%--------------### Ch from the proposed scheme ---------------------------
[~,Chindex]=max(cluster_CHtotal);   % get the cluster head vehicle ID index for
CHid(tim,k)  = vhclsIDinCluster(Chindex); % cluster head vehicle ID
[~,index_ecc]=max(cluster_CHeccencity); % maximum eccentricity vehicle index in a cluster
CHid_eccentricity(tim,k)  = vhclsIDinCluster(index_ecc);
[~,index_trust]=max(cluster_CHtrust); % maximum reliability score vehicle index in a cluster
```

CHid_trust(tim,k)  = vhclsIDinCluster(index_trust);

[~,index_rspeed]=max(cluster_CHavrege_speed);% maximum relative speed vehicle index in a cluster

CHid_relativeSpeed(tim,k)  = vhclsIDinCluster(index_rspeed);

[~,index_neigh]=max(cluster_CHneighbourDegree); % maximum Neighbourhood vehicle index in a cluster

CHid_nbrdegree(tim,k)  = vhclsIDinCluster(index_neigh);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

## Appendix 5: Grey Relational Analysis and PGRP Function

```
function grg = gra(temp,w)
% normtemp = normalize(temp,1,'range');
normtemp = normalize(temp,1);
for pp=1:size(normtemp,2)
   maxval(1,pp) = max(normtemp(:,pp));
   delta(:,pp)= maxval(1,pp)-normtemp(:,pp);
end
grg=zeros(size(delta,1),1); % grey relational grade initialization
for pp=1:size(normtemp,2)
   gamma(:,pp) = (min(delta(:,pp))+0.5.*max(delta(:,pp)))/(delta(:,pp)+0.5.*max(delta(:,pp)));
   grg = grg+w(pp).*gamma(:,pp);
end
end
```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

PGRP Function

```
function predicted = predictPGRP(loc,v,theta)
```

```
% loc = current locations co-ordinates
%v= velocity
% theta= moving angle
%t % time insec
X = loc(1)+v*t*cos(theta);
Y = loc(2)+v*t*sin(theta);
predicted=[X,Y];
%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Republic of Tunisia
Ministry of Higher Education, Scientific Research

University of Sfax
National Engineering School of Sfax

***Doctoral school***
***Science and Technology***

**Doctoral thesis**

**Computer Systems Engineering**
**Order N. 21/2023**

# HyperGraph Based Stable Clustering in VANET

**Mays Kareem Jabbar Alsabah**

**Abstract:** This thesis brings original contributions, mainly to VANET clustering algorithms. In order to guarantee cluster stability, two clustering approaches based on the hypergraph theory are designed as a complete solution for VANET challenges. Each approach is bifurcated into two parts; cluster generation and cluster head selection (CH). This is our experiment with the hypergraph theory at different clustering schemes and CH selection parameters. In this thesis, a formulation of VANET through hypergraph spectral clustering is introduced. Hypergraph partitioning through the tensor trace maximisation (TTM) method is presented. Then, the Eigen-trick is used to calculate the modified Laplacian value in TTM to improve the clustering. Also, this thesis presents two CH selection schemes using different parameters to select the most stable vehicle as a head. Strong connectivity and a stable link lifetime are obtained using these schemes. As well as these schemes improve the network performance compared with other techniques in the literature.
All simulation is conducted in MATLAB, with the Traffic Control Interface (TraCI) and the Simulation of Urban Mobility (SUMO) being used to model urban transportation. The crowded market area of Baghdad, Iraq is considered for the simulation, it is extracted from the open street map (OSM).

**Résumé :** Cette thèse apporte des contributions originales, principalement aux algorithmes de clustering VANET. Afin de garantir la stabilité du cluster, deux approches de clustering basées sur la théorie des hypergraphes sont conçues comme une solution complète pour les défis VANET. Chaque approche est divisée en deux parties ; génération de cluster et sélection de tête de cluster (CH). Il s'agit de notre expérience avec la théorie de l'hypergraphe à différents schémas de regroupement et paramètres de sélection CH. Dans cette thèse, une formulation de VANET par regroupement spectral d'hypergraphes est introduite. Le partitionnement hypergraphique par la méthode de maximisation de la trace du tenseur (TTM) est présenté. Ensuite, le Eigen-trick est utilisé pour calculer la valeur laplacienne modifiée en TTM pour améliorer le clustering. En outre, cette thèse présente deux schémas de sélection CH utilisant différents paramètres pour sélectionner le véhicule le plus stable en tant que tête. Une forte connectivité et une durée de vie de liaison stable sont obtenues à l'aide de ces schémas. De plus, ces schémas améliorent les performances du réseau par rapport aux autres techniques de la littérature.
Toutes les simulations sont effectuées dans MATLAB, avec l'interface de contrôle du trafic (TraCI) et la simulation de la mobilité urbaine (SUMO) utilisées pour modéliser le transport urbain. La zone de marché bondée de Bagdad, en Irak, est prise en compte pour la simulation, elle est extraite du plan des rues ouvert (OSM).

**Key-words:** VANET, Clustering algorithm, hypergraph, Tensor trace maximisation, Laplacian value, Cluster head, Stability, Relative speed, Neighbouring degree, Trust, Time to leave estimation, Long short term memory, Eigen-trick, Grey relational analysis.
**Mots clés:** VANET, Algorithme de clustering, hypergraphe, Maximisation de la trace tensorielle, Valeur laplacienne, Tête de cluster, Stabilité, Vitesse relative, Degré de voisinage, Confiance, Estimation du temps de départ, Mémoire à long terme, Eigen-trick, Analyse relationnelle grise .