Titre: Title:	An Improved Method to Increase Cluster Lifetime in Vehicular Ad Hoc Networks (VANETs)
Auteur: Author:	Seyedeh Elham Asghari Touchaei
Date:	2024
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Asghari Touchaei, S. E. (2024). An Improved Method to Increase Cluster Lifetime in Vehicular Ad Hoc Networks (VANETs) [Mémoire de maîtrise, Polytechnique Montréal]. PolyPublie. https://publications.polymtl.ca/62518/

URL de PolyPublie: PolyPublie URL:	https://publications.polymtl.ca/62518/
Directeurs de recherche: Advisors:	Yvon Savaria
Programme: Program:	Génie électrique

POLYTECHNIQUE MONTRÉAL

Affiliée à l'Université de Montréal

An Improved Method to Increase Cluster Lifetime in Vehicular Ad Hoc Networks (VANETs)

SEYEDEH ELHAM ASGHARI TOUCHAEI

Département de génie électrique

Mémoire présenté en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*

Génie électrique

Novembre 2024

POLYTECHNIQUE MONTRÉAL

Affiliée à l'Université de Montréal

Ce mémoire intitulé :

An Improved Method to Increase Cluster Lifetime in Vehicular Ad Hoc Networks (VANETs)

Présenté par **Seyedeh Elham ASGHARI TOUCHAEI**

en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*

a été dûment accepté par le jury d'examen constitué de:

Chahé NERGUIZIAN, président

Yvon SAVARIA, membre et directeur de recherche

Alejandro QUINTERO, membre

DEDICATION

To my parents and brothers, for their support and their encouragements during all these years.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to **Professor Yvon Savaria** for his invaluable academic and financial support, as well as his guidance throughout these past two years. His mentorship has been instrumental in the development and completion of this work.

I would also like to extend my thanks to **Professor Seyed Amir Asghari** and **Professor Mohammadreza Binesh Marvasti** for their academic support and insightful contributions, which have greatly enriched my research.

RÉSUMÉ

Les réseaux ad hoc véhiculaires (VANETs) représentent une technologie en plein essor visant à améliorer la sécurité routière, à optimiser la fluidité du trafic et à permettre une communication efficace entre les véhicules et l'infrastructure. Cependant, à mesure que le nombre de véhicules connectés augmente, la gestion de leur mobilité dans des environnements dynamiques et imprévisibles pose des défis significatifs. Une solution couramment employée est le regroupement de véhicules en clusters, où les véhicules sont organisés en groupes avec un chef de cluster désigné pour coordonner la communication. Bien que le regroupement offre des améliorations en termes d'efficacité de communication, la nature hautement dynamique des VANET introduit plusieurs difficultés, notamment des formations fréquentes de nouveaux regroupements, des têtes de regroupement de courte durée et des délais de communication accrus.

Les protocoles pour la formation de regroupement existants peinent souvent à maintenir leur dans de tels environnements en perpétuel changement, entraînant des changements fréquents de têtes de regroupement, une surcharge de communication accrue et une durée de vie du réseau réduite. Ces défis limitent le déploiement pratique des VANET dans des scénarios réels où la stabilité et une communication fiable sont essentielles. Par conséquent, il est nécessaire de disposer d'algorithmes pour la création et l'entretien de regroupements plus adaptatifs et stables, capables de gérer la haute mobilité des véhicules tout en assurant une communication inter-véhicule efficace et fiable.

Cette thèse aborde ces défis en proposant un mécanisme de regroupement amélioré et un protocole de sélection des têtes de regroupement adaptés à la nature dynamique des VANET. La méthode proposée utilise un algorithme de regroupement K-means, combiné avec la méthode du score de silhouette, pour déterminer dynamiquement le nombre optimal de clusters en fonction des paramètres clés des véhicules. Cela permet de former des regroupements plus stables pouvant s'adapter efficacement aux changements rapides de mobilité des véhicules. De plus, une formule pondérée est appliquée pour la sélection du chef de cluster, en intégrant sept facteurs : distance locale (LD), vitesse relative (RS), rapport de réponse (AR), degré de nœud (ND), distance à la station de base (BSD), confiance (T) et centre de nœud (NC), avec une somme de tous les poids égale à un. Ce processus de sélection optimisé garantit que les chefs de cluster restent stables pour des périodes plus longues, réduisant ainsi la nécessité de refaire les regroupements fréquemment en minimisant la surcharge de communication.

Pour valider l’approche proposée, des simulations sont menées en utilisant le logiciel de simulation de mobilité urbaine (SUMO), avec la gestion de la dynamique des véhicules et des comportements des regroupements assurés via la bibliothèque Traci en Python. Les résultats montrent que la méthode proposée surpasse de manière significative les approches existantes, telles que le regroupement des VANET basé sur la sélection de tête de regroupement de confiance pondérée, le protocole de routage amélioré basé sur le regroupement dans les réseaux ad hoc véhiculaires (ECBLTR) et le routage multicast optimisé basé sur les clusters dans les VANET utilisant un algorithme génétique basé sur les connaissances d'élite. Plus précisément, l'approche proposée a atteint une durée de vie du réseau de 5,78 unités de temps, avec 80 changements de regroupement et 86 changements de tête des regroupements, démontrant une performance supérieure dans ces métriques clés. Cette recherche propose une solution plus robuste et adaptative pour le regroupement dans les VANET, offrant des avancées substantielles pour réduire les changements de chef de cluster, améliorer la stabilité des clusters et prolonger la durée de vie du réseau. Ces contributions ouvrent la voie à des implémentations de VANET plus fiables et efficaces, facilitant leur intégration dans les futurs systèmes de transport intelligents.

ABSTRACT

Vehicular Ad Hoc Networks (VANETs) represent a rapidly advancing technology that improves road safety, optimizes traffic flow, and enables efficient communication between vehicles and infrastructure. However, as the number of connected vehicles grows, managing their mobility effectively in dynamic and unpredictable environments poses significant challenges. One commonly employed solution is vehicle clustering, where vehicles are grouped into clusters, with a designated cluster head coordinating communication. While clustering improves communication efficiency, the highly dynamic nature of VANETs introduces several difficulties, including frequent cluster reformation, short-lived cluster heads, and increased communication delays.

Existing clustering protocols often struggle to maintain cluster stability in rapidly changing environments, leading to frequent changes in cluster heads, elevated communication overhead, and a reduced network lifetime. These challenges limit the practical deployment of VANETs in real-world scenarios where stability and reliable communication are essential. As a result, there is a pressing need for more adaptive and stable clustering algorithms capable of vehicles' high mobility while ensuring efficient and reliable inter-vehicle communication.

This thesis addresses these challenges by proposing an enhanced clustering mechanism and cluster head selection protocol tailored to the dynamic nature of VANETs. The proposed method leverages a K-Means clustering algorithm, combined with the Silhouette Score Method, to dynamically determine the optimal number of clusters based on key vehicular parameters. This ensures more stable cluster formations that can adapt effectively to rapid changes in vehicle mobility. Additionally, a weighted formula is applied for cluster head selection, incorporating seven factors: Local Distance (LD), Relative Speed (RS), Answer Ratio (AR), Node Degree (ND), Base Station Distance (BSD), Trust (T), and Node Center (NC), with the sum of all weights equaling one. This optimized selection process ensures that cluster heads remain stable for longer durations, significantly reducing the need for frequent re-clustering and minimizing communication overhead. To validate the proposed approach, simulations are conducted using the Simulation of Urban MObility (SUMO) software, with vehicle dynamics, and cluster behavior managed through the Traci library in Python. The results demonstrate that the proposed method significantly outperforms existing approaches, such as VANET Clustering Based on Weighted Trusted Cluster Head Selection, the Enhanced Clustering-Based Routing Protocol in Vehicular Ad-hoc Networks (ECBLTR), and Cluster-based multicast optimized routing in VANETs using elite knowledge-based genetic algorithm. Specifically, the proposed approach achieved a network lifetime of 5.78 unit time, with 80 cluster changes and 86 cluster head changes, demonstrating superior performance in these key metrics. This research presents a more robust and adaptive solution for

clustering in VANETs, offering substantial advancements in reducing cluster head changes, enhancing cluster stability, and extending network lifetime. These contributions pave the way for more reliable and efficient VANET implementations, facilitating their integration into future intelligent transportation systems and potentially improving the safety and efficiency of our roads.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

ACO	Ant Colony Optimization
AR	Answer Ratio
AWCP	Adaptive Weighted Clustering Protocol
BCBC	Betweenness Centrality Based Clustering
BES	Bald Eagle Search
BSD	Base Station Distance
BWOA	Beluga Whale Optimization Algorithm
CCM	Cluster Cabinet Member
CH	Cluster Head
CHFHB	Customized Hunger's Foraging Honey Badger Optimization
CHL	Cluster Head Lifetime
CHS	Cluster Head Selection
CM	Cluster Member
CML	Cluster Member Lifetime
CPO	Control Packet Overhead
DA-TRPED	Dynamic Aware Transmission Range Parallel Euclidean Distance
DHC	Double-Head Clustering
DMNSGA	Dynamic Multi-objective Non-Sorted Genetic Algorithm
DPR	Dropped Packet Ratio
DT	Direct Trust
E2E-D	End-to-End Delay
ECBLTR	Enhanced Clustering-Based Routing Protocol with Sugeno Fuzzy Inference System
EKSGA	Elite Knowledge Sharing Genetic Algorithm
FA	Firefly Algorithm
FFOA	Fennec Fox Optimization Algorithm
FIS	Fuzzy Inference System
GA	Genetic Algorithm
GBTR	Graph-Based Trust-Enabled Routing
HFFSCOA	Hybrid Fennec Fox and Sand Cat Optimization Algorithm
HPDBWOA-NC	Hybrid Prairie Dogs and Beluga Whale Optimization-based Node Clustering
HGFA	Hybrid Genetic Firefly Algorithm

I2I	Infrastructure to Infrastructure
IDT	Indirect Trust
IFBE	Intelligent Fuzzy Bald Eagle
ITS	Intelligent Transportation System
LET	Link Expiration Time
LD	Local Distance
MANET	Mobile Ad Hoc Network
MFCA-IoV	Moth Flame Clustering Algorithm for Internet of Vehicle
MFO	Moth Flame Optimization
NC	Node Center
ND	Node Degree
NSGA-II	Non-dominated Sorting Genetic Algorithm version 2
OBU	On-Board Unit
ONC	Optimal Number of Clusters
PDOA	Prairie Dog Optimization Algorithm
PDR	Packet Delivery Ratio
PGG	Public Goods Game
PMC	Priority-based Multi-hop Clustering
RS	Relative Speed
RST	Rough Set Theory
RSU	Road-Side Unit
SCH	Secondary Cluster Head
SCOA	Sand Cat Optimization Algorithm
SNR	Signal-to-Noise Ratio
SUMO	Simulation of Urban Mobility
T	Trust
TACR	Trust Dependent Ant Colony Routing
TB	Threshold-Based
TS	Tabu Search
V2X	Vehicle-to-Everything
V2G	Vehicle-to-Grid
V2I	Vehicle to Infrastructure
V2N	Vehicle-to-Network
V2P	Vehicle-to-Pedestrian
V2V	Vehicle to Vehicle
V2X	Vehicle to Anything
VANET	Vehicular Ad-Hoc Network

VMaSC	Vehicular Multi-hop Algorithm for Stable Clustering
VTD	Virtual Trust-ability Data transmission
WKCA	Weighted K-medoids Clustering Algorithm
WTCHS	Weighted Trusted Cluster Head Selection

CHAPTER 1 INTRODUCTION

1.1 Overview of Vehicular Ad Hoc Networks (VANETs)

Vehicular Ad Hoc Networks (VANETs) are a subset of Mobile Ad Hoc Networks (MANETs), specially designed for vehicular environments [1]. VANETs are integral to the evolution of Intelligent Transportation Systems (ITS), as they enable vehicles to communicate with each other and roadside infrastructure to improve road safety, optimize traffic flow, and provide infotainment services to passengers. Unlike traditional networks, VANETs operate without relying on centralized infrastructure, leveraging dynamic communication between mobile nodes (vehicles) that constantly change position due to high mobility [2], [3], [4].

The fundamental feature of VANETs is their dynamic topology, characterized by frequent changes in network connectivity due to the fast movement of vehicles. This makes maintaining a stable network a critical challenge [5], [6]. VANETs are used in a variety of applications, ranging from safety-related services like collision avoidance to non-safety-related services like entertainment[7]. Given these use cases, VANETs must support high reliability, low latency, and robustness despite the challenges introduced by the dynamic nature of vehicular environments [8].

VANET communication relies on three key hardware components: On-Board Units (OBUs) installed in vehicles, Road-Side Units (RSUs) deployed along roadsides, and Base Stations [9]. These components work in tandem to create the vehicular communication network:

- **On-Board Units (OBUs):** OBUs are communication devices installed in vehicles, enabling them to communicate with other vehicles (in vehicle-to-vehicle communications) and with RSUs (in vehicle-to-infrastructure communications). OBUs transmit information such as vehicle speed, position, and direction to nearby vehicles and RSUs, playing a crucial role in delivering real-time traffic information and safety warnings [9], [10].
- **Road-Side Units (RSUs):** RSUs are fixed devices positioned at specific points along the road, typically at intersections or high-traffic areas. RSUs communicate with vehicles (V2I communication) and infrastructure (I2I communication) to relay critical information such

as traffic signals, accident reports, and road conditions. RSUs also function as gateways for internet access and can store data temporarily for later transmission [9], [11].

- **Base Stations**: While RSUs facilitate communication within a localized area, base stations provide broader network access and connectivity. Base stations can relay data from RSUs to central servers or cloud systems, enabling large-scale data analysis and decision-making for citywide traffic management systems [11].

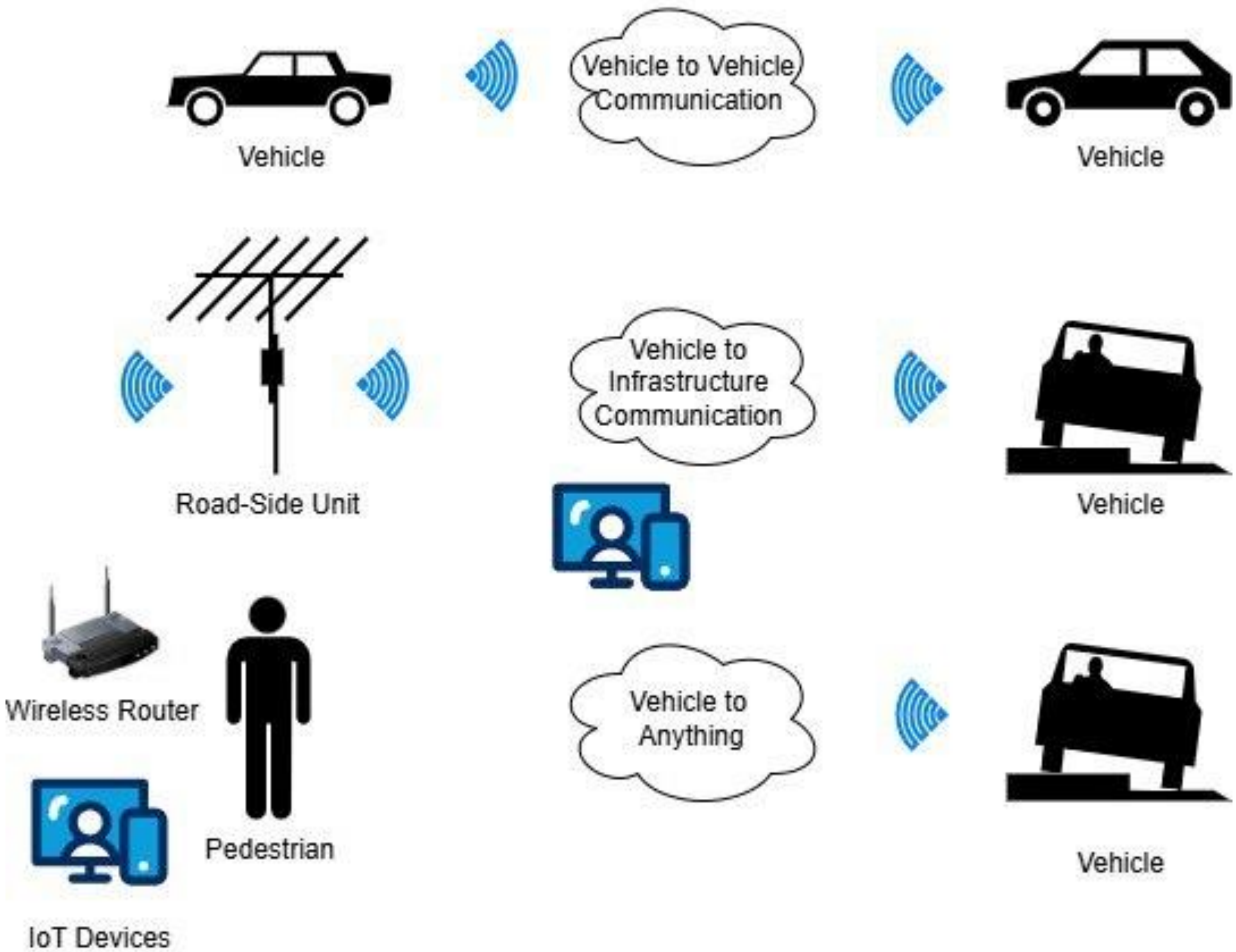

Figure 1-1 A VANET Infrastructure

1.2 Types of Communication in VANETs

As illustrated in Figure.1, there are several types of communication, each serving different purposes in the vehicular ecosystem :

1. **Vehicle-to-Vehicle (V2V) Communication:** V2V communication allows vehicles to communicate with each other within a short range directly. This type of communication is essential for safety applications such as collision avoidance, lane-change warnings, and cooperative driving [12][13].
2. **Vehicle-to-Infrastructure (V2I) Communication:** V2I communication enables vehicles to communicate with roadside infrastructure, such as traffic lights, RSUs, and parking systems. This bidirectional communication facilitates tasks like road toll payments, traffic light coordination, and providing vehicles with information on upcoming road hazards [14].
3. **Vehicle-to-Anything (V2X) Communication:** V2X communication is an umbrella term that includes both V2V and V2I communications, as well as Vehicle-to-Pedestrian (V2P), Vehicle-to-Grid (V2G), and Vehicle-to-Network (V2N) communications. V2X allows vehicles to interact with a wide variety of entities, including pedestrians with mobile devices, charging stations for electric vehicles, and network infrastructures that manage traffic[15].
4. **Infrastructure-to-Infrastructure (I2I) Communication:** I2I communication involves exchanging information between infrastructure elements like RSUs, base stations, and traffic management centers. This communication is vital for large-scale traffic coordination, enabling efficient data transmission between different nodes in the infrastructure network [5].

1.3 Challenges in VANET Communication and Clustering

Clustering is essential in VANETs due to the unique challenges posed by vehicles' high mobility and the network's dynamic nature [16]. In VANET, vehicles are constantly moving, leading to frequent network topology changes [15]. Without clustering, managing communication between numerous vehicles would be inefficient and cause high communication overhead, congestion, and delays [17]. Clustering helps by grouping vehicles into smaller, manageable clusters, each with a designated Cluster Head (CH) that coordinates communication within the cluster and relays information to other clusters or roadside infrastructure [18]. This hierarchical structure reduces the routing complexity, as only the CHs need to communicate with each other or external systems, rather than every vehicle communicating individually [19]. Moreover, clustering enhances scalability, allowing the network to efficiently handle many vehicles, especially in high-density areas like cities [20][21]. It also improves resource management by minimizing redundant communications and saving bandwidth. Clustering helps stabilize communication by selecting CHs based on factors such as vehicle speed, proximity, and direction, which reduces the need for frequent re-clustering and ensures that the network adapts to changes in vehicle positions [22]. Additionally, clustering supports faster and more reliable data transmission, critical for time-sensitive applications like traffic safety alerts and collision avoidance systems, while extending network lifetime through more balanced energy consumption[23]. In essence, clustering is crucial for maintaining an efficient, stable, and scalable VANET capable of operating in complex, rapidly changing environments [24].

Despite the advantages of clustering, several challenges must be addressed:

- **Frequent Cluster Reformation:** Due to the high mobility of vehicles, clusters often need to be reformed frequently, which increases communication overhead and reduces network efficiency [25].
- **Cluster Head Longevity:** Ensuring the longevity of a CH in a dynamic environment is challenging. A short-lived CH leads to frequent changes, which can disrupt communication within clusters [9].
- **Communication Latency:** Delays in message transmission are critical in VANETs, particularly for safety applications. Inefficient clustering protocols can increase latency, reducing the effectiveness of the system[26].

1.4 Clustering in VANETs

This thesis introduces an approach to clustering in VANETs, addressing the significant challenge of maintaining network stability in highly dynamic vehicle environments. It focuses on enhancing the network's stability and efficiency in these environments. The proposed method is divided into two main phases: dynamic clustering and optimized cluster head selection.

1.4.1 Dynamic Clustering Using K-Means Algorithm, Silhouette Score Method, and Weighted Formula for Cluster Head Selection:

To effectively manage communication between vehicles, a dynamic clustering algorithm is implemented using the K-Means clustering algorithm [27]. The algorithm groups vehicles into clusters based on key vehicular parameters such as position and direction. The optimal number of clusters is determined using the Silhouette Score Method, which evaluates how well each vehicle fits within its assigned cluster. This approach ensures that clusters are well-formed, minimizing overlaps and maintaining stability, even in rapidly changing environments.

Once clusters are formed, the next step is to select the most suitable Cluster Head (CH) for each cluster. The CH is responsible for managing communication within the cluster and relaying messages to other clusters or roadside units (RSUs) [28], [29]. The selection of the CH is based on a weighted formula that considers several critical factors, including [22]: Local Distance (LD), Relative Speed (RS), Trust (T), Base Station Distance (BSD), Answer Ratio (AR), Node Degree (ND), and Node Center (NC)

By optimizing these parameters through a weighted scoring system, the method selects the vehicle most capable of maintaining cluster stability and ensuring efficient communication.

This two-phase approach improves cluster stability and reduces the frequency of re-clustering and cluster head changes, leading to better performance and extended network lifetime. The proposed method is validated through simulations, showing significant improvements in terms of cluster head changes, cluster stability, and overall network efficiency.

1.5 Research Objectives

This research addresses the abovementioned challenges by proposing an enhanced clustering mechanism for VANETs. The objectives of this research are:

1. **Reduce Cluster Head Changes:** Develop an algorithm that optimizes CH selection to minimize the frequency of cluster reformation and increase cluster stability.
2. **Reduce Cluster Changes:** Implement a clustering mechanism that reduces the need for frequent re-clustering, thus improving network efficiency.
3. **Enhance Network Lifetime:** Design a clustering protocol that extends network lifetime by optimizing resource utilization and minimizing unnecessary energy consumption during communication.

1.6 Thesis Roadmap

The structure of the thesis is as follows:

- **Chapter 1:** Introduction – This chapter provides an overview of VANETs, focusing on their significance in Intelligent Transportation Systems, communication types, and the clustering challenges inherent to high-mobility environments. It also outlines the research objectives, and details the study's contributions to clustering methodologies for dynamic vehicular networks.
- **Chapter 2:** Literature Review – A comprehensive review of existing VANET clustering and cluster head selection techniques, assessing approaches including K-means clustering, fuzzy logic-based clustering, trust-based selections, and hybrid optimizations. This chapter examines these methodologies' strengths, limitations, and potential improvements in the context of highly dynamic VANET scenarios.
- **Chapter 3:** Methodology – This chapter explains the proposed clustering framework, which incorporates a K-Means algorithm enhanced with the Silhouette Score for dynamic

cluster determination and a weighted formula for cluster head selection. The methodology is also illustrated through pseudocode and a detailed flowchart

- **Chapter 4:** Results and Discussion – In this chapter, the simulation outcomes are presented and analyzed, and the proposed clustering approach is compared against established techniques such as WTCHS, ECBLTR, and EKSGA. Metrics such as cluster head longevity and network lifetime are discussed to illustrate the advantages of the new methodology.
- **Chapter 5:** Conclusion and Future Work – This concluding chapter summarizes the key findings of the research and suggests potential avenues for future enhancements in VANET clustering, including advanced simulations and additional parameters for cluster head selection.

CHAPTER 2 A REVIEW OF THE LITERATURE

Multiple research groups have investigated different approaches to clustering VANETs and selecting cluster heads. We examine the potential advantages and limitations of different clustering algorithms in VANETs. In this chapter, we will explore some of the state-of-the-art VANET clustering and cluster head selections to build a basis for the reader prior to presenting our novel approach.

2.1 Trust-based Clustering VANET Approaches

In a research paper presented by Khayat et al.[22], they investigated a clustering algorithm that introduces trust as a novel metric in CHS. This approach uses a weighted formula incorporating trust, distance, and velocity to enhance cluster stability and reduce delays. Trust is calculated through direct and indirect trust values, ensuring that reliable and cooperative vehicles are more likely to be selected as CHs. This method addresses the frequent topology changes inherent in VANETs, significantly improving delay metrics and network reliability. The authors conducted simulations on a straight-road scenario with 20 randomly positioned vehicles to test their clustering protocol's effectiveness. The simulation results demonstrated that the algorithm improved cluster stability and reduced delays compared to traditional methods. Clusters formed based on direct trust values showed better performance in terms of delay reduction while incorporating indirect trust helped mitigate the impact of malicious nodes. Despite these advancements, there are areas for further enhancement. Implementing dynamic weight adjustment based on real-time network conditions could optimize CH selection, potentially extending network lifetime and reducing delays. Incorporating residual energy as an additional parameter in the CH selection process can distribute energy consumption more evenly, further extending network longevity and reducing the frequency of CH changes.

Mirsadeghi et al.[18], proposed a trust-based authentication method to address network clusters' security and stability by improving the reliability and longevity of VANET clusters. The proposed method focuses on creating trustworthy and stable clusters by estimating the trust degree of each vehicle. This estimation is done through a combination of direct trust, based on the interactions between vehicles, and indirect trust, derived from information provided by Roadside Units (RSUs). Vehicles with the highest trust scores are then selected as cluster heads, a process designed to minimize the frequency of cluster head changes and enhance the overall network stability. The trust-based method also includes robust monitoring and authentication system. Within each cluster, designated verifier nodes continuously monitor the behavior of vehicles, working in tandem with cluster heads to swiftly identify and isolate any malicious nodes. The authentication process is further secured using a public/private key infrastructure, ensuring that only authenticated and trusted nodes can participate in network communication. The results indicated that the proposed trust-based method significantly enhances the accuracy of detecting malicious nodes, increases the packet delivery ratio, and reduces the delay in authentication processes compared to existing methods.

Awan et al. [30], used StabTrust mechanism that selects the CH through a trust-based approach, which evaluates nodes based on three key trust parameters: knowledge, reputation, and experience. The RSU collects information about all nodes within a cluster and calculates these trust components, where knowledge includes integrity and cooperativeness, reputation reflects honesty and behavior, and experience measures competence and end-to-end packet delivery. The node with the highest trust score is selected as the CH, while the top three nodes with the next highest scores are chosen as backup CHs, ensuring stability in the event of CH failure or departure. This method provides several advantages, including improved cluster stability by reducing the need for frequent CH re-selection, enhanced security through the isolation of malicious or compromised nodes, and greater energy efficiency, as backup CHs minimize the computational overhead associated with CH transitions. In simulations, StabTrust demonstrated performance, with cluster head lifetimes extending up to 130 seconds, outlasting other clustering algorithms. Even under varying conditions, such as node speed and transmission range, StabTrust maintained robust stability and security. However, their approach has limitations. One of the main challenges is that StabTrust is highly dependent on the RSU infrastructure, which may not always be available or scalable in dense urban environments. The system also requires nodes to be IoT-enabled, which could limit its applicability in regions with limited IoT deployment. Additionally, while the backup CH mechanism enhances stability, introducing multiple layers of trust evaluation may increase the system's complexity and lead to additional processing delays in scenarios with high node mobility. These factors could reduce its effectiveness in real-world, large-scale VANETs.

In another study conducted by Gayathri et al. [31], they investigated the subject of VANETs using a Fuzzy-Based Trusted Communication method to detect and prevent malicious nodes from participating in communication. The proposed approach utilizes the Mamdani fuzzy inference system to determine the trustworthiness of nodes based on factors such as distance and trust scores provided by certificate authorities. Based on these metrics, the system classifies nodes as fully trusted, rarely trusted, or malicious, ensuring secure data transmission between vehicles. The main advantage of this method is its ability to offer flexible and secure communication by incorporating trust parameters that handle the dynamic nature of vehicular networks. Fuzzy logic enables the system to manage uncertainties, such as rapidly changing vehicle behavior and provides a degree of truth in selecting reliable nodes for communication. This approach effectively reduces packet drops, minimizes communication delays, and increases throughput by detecting and excluding malicious nodes. Despite these advantages, the method has some limitations. One notable challenge is the processing overhead of the fuzzy logic system, particularly in dense traffic scenarios, which could impact the system's performance. Another limitation is that the method mainly addresses the trustworthiness of nodes based on historical and current behavior. Still, it may not adequately detect sophisticated or zero-day attacks where malicious behavior is not evident from past data.

Sahoo et al.[6], proposed a method for selecting cluster heads by combining a trust-based mechanism and Ant Colony Optimization (ACO) for routing. The cluster head selection is based on factors such as vehicle speed, direction, and trust value. Vehicles move in the same direction and with similar speeds form clusters. The slowest moving vehicle is initially chosen as the cluster head because it is likely to remain in the cluster for a longer time. However, before finalizing the CH, the value of the vehicle is evaluated. The trust value is calculated based on direct trust and indirect trust. Vehicles with trust values higher than a predefined threshold are selected as CHs. After the clusters are formed, the ACO algorithm is employed for routing between CHs. ACO optimizes the communication paths by minimizing routing overhead and ensuring data packets are delivered efficiently across the network. Simulations demonstrate that the Trust Dependent Ant Colony Routing (TACR) algorithm reduces routing overhead. TACR achieves improvements in terms of Packet Delivery Ratio (PDR) and routing efficiency, especially in scenarios with a high number of vehicles.

Alam et al.[23], introduced the Graph-Based Trust-Enabled Routing (GBTR) protocol, designed to enhance security and efficiency in VANETs [23]. The GBTR protocol integrates three types of trust—direct, indirect, and contextual—to evaluate the trustworthiness of nodes and improve routing decisions. Direct trust is based on communication metrics; indirect trust considers feedback and link reliability, and contextual trust accounts for environmental factors like location and traffic density. GBTR's routing algorithm prioritizes paths with the highest trust scores, ensuring secure and reliable data transmission. The protocol also includes a trust update mechanism that dynamically adjusts trust values and maintains a blacklist of untrustworthy nodes, further enhancing network security. Simulations using the Veins simulator showed that GBTR outperforms existing protocols across key metrics, including PDR, Dropped Packet Ratio (DPR), End-to-End Delay (E2E-D), and Throughput. GBTR achieved higher PDR and throughput, with lower delay and routing overhead.

Ruban et al. [27], investigated Cluster-Based Trust Model for Secure Communication in VANET. This study introduces the Virtual Trust-ability Data transmission (VTD) protocol, which enhances secure communication in VANETs by establishing trust between nodes to prevent the transmission of false data. The protocol operates in three stages: virtual topology creation, trust-ability analysis, and data transmission. Using the K-means clustering algorithm, vehicles are organized into clusters based on their proximity, with a central node selected in each cluster. In the second stage, the trustworthiness of each node is evaluated based on its historical communication behavior and feedback from other nodes, and a trust value is assigned to determine whether to accept or reject messages. During data transmission, only trusted nodes are involved, ensuring that unreliable nodes are excluded from the network. The VTD protocol demonstrates several advantages, such as improving the PDR to 94% and significantly reducing packet failure rates to 6%, thus minimizing false data transmission. The trust-based clustering model enhances scalability and network security, as untrustworthy nodes are systematically excluded. Furthermore, the use of K-means clustering efficiently organizes vehicles into stable groups, allowing for better bandwidth utilization and overall network performance. However, continuously calculating and updating trust values for each node can slow down real-time decision-making, which is essential for efficient communication.

2.2 Fuzzy-logic-based Clustering

In another study conducted by Naeem et al.[11], they used an alternate approach that employs a Sugeno model fuzzy inference system for assessing cluster heads (CH) using input parameters such as residual energy, local distance, node degree, concentration, and distance from the base station. In this approach, vehicles are divided into clusters based on their locations. Each cluster is assigned a unique ID. The fuzzy system evaluates each node based on input parameters. The node with optimal characteristics is selected as the CH, responsible for data aggregation and transmission to the base station. CHs maintain a routing table containing information about lifetime, expiry time, associated locations, and CH IDs. When receiving a packet, the CH searches the routing table for the next hop towards the destination. If no immediate CH is found, the packet is stored and forwarded later. The proposed protocol showed a 10% increase in network lifetime, along with improvements in packet delivery ratio, average end-to-end delay, and overhead transmission compared to existing methods. However, we can make considerable improvements by optimizing the fuzzy inference system by incorporating machine learning techniques to adaptively adjust the input parameters based on changing network conditions.

Aissa et al. [25], uses a fuzzy logic approach to select the cluster head. Their method uses metrics like average distance, relative velocity, and link connectivity duration, to evaluate each vehicle's suitability to serve as a CH. These factors are processed using a fuzzy inference system, which calculates a node's FitFactor. Vehicles with the lowest FitFactor values are elected as CHs, ensuring they maintain a safe distance from neighboring vehicles, travel at similar speeds, and exhibit stable link connectivity. The advantages of this fuzzy logic-based method include the creation of more stable clusters, which reduce the need for frequent re-clustering. This stability is achieved by prioritizing vehicles that maintain safe distances and have consistent speeds, thereby minimizing the risk of collisions and interruptions in communication. Additionally, this approach extends the cluster lifetime by 20-30% compared to existing methods, reducing overhead and enhancing overall network efficiency. However, the limitations of this work include its dependency on a precise calculation of metrics, which may introduce computational complexity, particularly in highly dynamic environments where vehicle speeds and positions change rapidly.

Brindha et al.[24], investigated the Intelligent Fuzzy Bald Eagle (IFBE) optimization algorithm. The study integrates fuzzy logic with the Bald Eagle Search (BES) optimization algorithm to improve the accuracy and efficiency of CH selection, which is critical for maintaining network stability in the dynamic environments typical of VANETs. The main advantages of this method include a significant reduction in energy consumption and end-to-end delay, as well as an improvement in the packet delivery ratio (PDR) and overall clustering efficiency. The results of the IFBE optimization showed better performance compared to existing. However, the limitations of this approach include the added computational complexity of using fuzzy logic combined with BES, which may increase processing time in large-scale networks.

2.3 Clustering Based on a Rough Set Scheme

In another study conducted by Dua et al., they used an alternate approach, a rough set scheme (RoVAN), which leverages Rough Set Theory (RST) to optimize the process of cluster head selection in VANETs [16]. Traditional CH selection methods often consider all cluster members, which can be time-consuming and inefficient, particularly as the number of nodes in a cluster increases. RoVAN addresses this issue by reducing the number of nodes involved in the CH selection process, thereby minimizing the time required for selection while enhancing cluster stability. RoVAN operates by forming equivalence classes within a cluster, where nodes that share similar attributes—such as position and speed—are grouped. Only one node from each equivalence class, termed a Cluster Cabinet Member (CCM), participates in the CH selection process. This reduction in participating nodes accelerates selection and leads to more stable clusters. The CH is chosen based on the calculated dependency of each node using RST, with the node receiving the highest number of votes from other CCMs being elected as the CH. The results demonstrated that RoVAN reduces CH selection time compared to existing methods, especially as vehicle density increases. Additionally, the scheme improved the reliability of CHs, maintaining cluster stability even as vehicle speed increased. However, there are some drawbacks to the RoVAN scheme, including implementation complexity due to the use of Rough Set Theory (RST), which can be computationally intensive, especially in large-scale networks. Its effectiveness relies on accurate data exchange, making it sensitive to inaccuracies or delays. Additionally, environmental changes can affect performance, and managing equivalence classes may introduce overhead, potentially offsetting some benefits.

2.4 Multi-Head Clustering Algorithm

In another study conducted by Alsuhli et al. [20], they looked into the Mobility-Based Double-Head Clustering (DHC) algorithm for dynamic VANETs, which introduces a new clustering method that enhances stability by selecting two cluster heads (CHs) for each cluster: a primary CH and a Secondary Cluster Head (SCH). The primary CH is responsible for leading the cluster and managing communications. At the same time, SCH serves as a backup, preventing unnecessary re-clustering when communication with the primary CH is temporarily lost. The selection of the cluster heads is based on several mobility metrics, including the vehicle's speed, position, direction, popularity, signal-to-noise ratio (SNR), and link expiration time (LET). These metrics are combined into an eligibility score that determines the CH selection, with vehicles that maintain the highest eligibility scores becoming the primary and secondary CHs. The advantages of this approach include improved cluster stability, as the presence of the SCH reduces the need for re-clustering, especially in dynamic environments where vehicles may frequently lose temporary connections. The algorithm also reduces the control packet overhead since fewer re-clustering events occur. Simulations show that the DHC algorithm outperforms traditional single-head clustering algorithms, such as the Threshold-Based (TB) algorithm, by increasing the cluster head lifetime (CHL) and cluster member lifetime (CML), thus ensuring more consistent communication within clusters. However, limitations of this approach include the added complexity of maintaining two CHs, which could increase the processing load on vehicles and the overhead of managing dual CH communication.

Alsuhli et al. [32], investigated an evolutionary approach for optimizing clustering in VANETs, framing the problem as a many-objective optimization challenge. They apply the NSGA-III metaheuristic algorithm to optimize critical parameters of the Double Head Clustering (DHC) algorithm, including Cluster Head Lifetime (CHL), Cluster Member Lifetime (CML), Control Packet Overhead (CPO), and the total number of clusters formed. The optimized version, called Optimized Configuration DHC (OCDHC), shows substantial improvements, with cluster head lifetime increasing by up to 134% and control packet overhead reducing by 30%. The advantages of this approach lie in its ability to enhance cluster stability, reduce the need for frequent re-clustering, and lower communication overhead, resulting in more efficient VANET operations. The optimized configuration also scales effectively across different vehicle densities, making it suitable for highly dynamic environments. However, a notable limitation is the computational complexity involved in the many-objective optimization process, as it requires significant computational resources and time—up to a week in their simulations—posing challenges for real-time applications in VANETs.

Vergis et al. [17], introduced a multi-head clustering algorithm to improve the stability and longevity of clusters in VANETs. This algorithm addresses the challenges posed by the high mobility of vehicles and the dynamic topology of VANETs by enabling vehicles to be part of multiple clusters simultaneously. By distributing the responsibilities of the CH across several vehicles, the algorithm reduces the frequency of CH changes, which is a critical factor in maintaining network stability and ensuring efficient communication. The algorithm uses mobility-based metrics such as speed, acceleration, and direction to form clusters of vehicles with similar movement patterns. The ability of the algorithm to allow vehicles to belong to multiple clusters simultaneously means that even if a vehicle moves out of range of one CH, it can remain part of the network through another CH, thereby maintaining the integrity and efficiency of the network. The Simulation of Urban MObility (SUMO) was used to simulate realistic vehicular traffic conditions, with the road topology based on the actual network of the town of Corfu. This provided a realistic and challenging testbed for evaluating the algorithm. The results indicated that the multi-head clustering algorithm outperformed the previously proposed double-head clustering algorithm, particularly regarding cluster lifetime, cluster size, and reduced time vehicles spent outside clusters.

2.5 Betweenness Centrality Based Clustering

In a research paper presented by Jabbar et al.[12], a Betweenness Centrality Based Clustering (BCBC) approach was proposed to improve the stability of clusters within VANETs. This study aims to address high vehicle mobility and frequent changes in network topology by introducing a novel clustering technique that leverages betweenness centrality, a concept from graph theory, to select the most stable CHs, thereby enhancing the overall stability of the network. The BCBC approach focuses on creating a more stable clustering structure using a fixed clustering model based on vehicles' geographical area and transmission range. The clustering process is initiated using the

K-means method, a widely used partitioning strategy employed here to form clusters based on the Optimal Number of Clusters (ONC). Once the clusters are formed, the betweenness centrality method is utilized to select the CH. Betweenness centrality measures how often a node (vehicle) acts as a bridge along the shortest path between two other nodes in a graph. The vehicle with the highest betweenness centrality is chosen as the CH, ensuring that the most strategically positioned vehicle assumes the leadership role, contributing to greater cluster stability. The simulation results revealed that the BCBC approach outperforms existing clustering algorithms, such as the Priority-based Multi-hop Clustering (PMC) and the Vehicular Multi-hop Algorithm for Stable Clustering (VMaSC), particularly in terms of CH and CM lifetimes. The BCBC demonstrated a longer CH lifetime, which indicates its effectiveness in maintaining cluster stability despite the dynamic nature of vehicular networks. The approach also showed a higher CM lifetime, meaning that vehicles remained within their respective clusters for longer, reducing the need for frequent cluster reformation. Despite these advantages, there are several drawbacks. The computational overhead associated with centrality calculations can pose challenges, especially in highly dense networks, where the betweenness centrality method may struggle with processing delays. Furthermore, reliance on fixed clustering techniques might limit adaptability to fluctuating traffic conditions, such as sudden vehicle speed or density increases.

In another study, Aditya et al.[33], introduced a method to select CHs using closeness centrality. Closeness centrality measures how close a node is to all other nodes in the network, prioritizing nodes with the shortest average path to others. This metric allows for selecting cluster heads that can effectively communicate with other nodes in their vicinity, reducing latency and improving overall network performance. The advantages include faster communication and extended network lifetime due to balanced energy consumption among CHs. Simulations show improvements in energy consumption, latency, and network lifetime compared to other methods.

Choudhary et al.[34], investigated the subject of VANETs using a centrality-based clustering and link prediction method to select cluster heads based on centrality measures such as degree centrality, closeness centrality, and betweenness centrality. Betweenness centrality, in particular, measures the extent to which a node lies on the shortest paths between other nodes, indicating its role in facilitating communication. The advantages of this centrality-based method include identifying the most strategically positioned vehicles in the network, ensuring that the selected cluster heads maintain stable connections, and facilitating efficient data transmission. This approach is effective in non-GPS environments, such as tunnels, where conventional localization systems like GPS fail. By relying on the centrality of vehicles in the communication network, the algorithm ensures robust connectivity and localization without needing external systems. The results of this approach, as demonstrated through simulations using NS-2 and SUMO, show improved accuracy in determining the location of vehicles in tunnels. The cluster-based communication system enables vehicles outside the tunnel to provide resources and information to vehicles inside the tunnel, ensuring uninterrupted communication and efficient resource allocation. However, a limitation of this work is calculating metrics such as betweenness centrality, which can be computationally intensive, especially in networks with high vehicle density and frequent changes in topology. Additionally, while the model addresses non-GPS environments like tunnels, its performance in more complex urban scenarios with varying traffic patterns is not fully explored.

2.6 Genetic-Algorithm-Based Clustering

In another study conducted by Hadded et al. [19], they used an alternate approach of using genetic algorithm for selecting cluster heads. In this study, they introduced the Adaptive Weighted Clustering Protocol (AWCP). AWCP selects cluster heads based on several key factors, such as highway ID, vehicle direction, position, speed, and number of neighboring vehicles. To optimize the performance of AWCP, the authors applied the Non-dominated Sorting Genetic Algorithm version 2 (NSGA-II). This algorithm addresses the multi-objective optimization problem of clustering in VANETs, where the goals are to maximize cluster stability, increase the data delivery rate, and reduce clustering overhead. NSGA-II is used to fine-tune the parameters of AWCP, ensuring that the protocol can efficiently manage the dynamic nature of VANETs while maintaining stable communication between vehicles. The advantages of AWCP include its ability to significantly improve cluster stability by ensuring that vehicles moving in the same direction on the same highway remain in the same cluster. This minimizes frequent re-clustering, often necessary in highly mobile networks. Additionally, using multi-objective genetic algorithms to select the best parameters enhances the overall efficiency of data transmission, reducing delays and increasing the packet delivery ratio. The protocol also reduces control packet overhead, making it more scalable and suitable for large-scale vehicular networks. AWCP achieved better results regarding cluster stability, data delivery rate, and reduced overhead, proving its effectiveness in enhancing the overall performance of VANETs. However, there is a limitation to the AWCP method. Firstly, computational time remains a challenge due to the complexity of the multi-objective optimization process, particularly when running simulations in scenarios with high vehicle density. For example, the authors reported that running multiple independent simulations to identify the optimal parameter set took considerable time, which may limit the practicality of real-time deployment.

Singh et al. [26], investigated the Hybrid Genetic Firefly Algorithm-Based Routing Protocol (HGFA) for VANETs, which integrates the Genetic Algorithm (GA) with the Firefly Algorithm (FA) to optimize routing in both sparse and dense network scenarios. The method uses firefly-inspired behavior, where fireflies are attracted to each other based on the brightness of their flashes, to determine the best routes for data transmission. The methodology starts with GA to initialize the population of vehicle nodes and generate diverse routing paths through genetic operations like crossover. After this, the FA refines the search by treating each vehicle as a "firefly" that is attracted to brighter fireflies, representing better routes. The algorithm minimizes transmission delays and improves the packet delivery ratio by continuously adjusting based on real-time network conditions.

In another study conducted by Hajlaoui et al. [35], they proposed an enhanced version of the Weighted K-medoids Clustering Algorithm (WKCA), integrating a hybrid genetic algorithm and tabu search to improve the stability of VANETs. The key method used is the Genetic Algorithm (GA), which Tabu Search (TS) enhances to avoid getting trapped in local optima. The Enhanced WKCA (E-WKCA) works by improving the assignment of vehicles to clusters. The GA is responsible for creating an initial population of clusters and applying crossover and mutation operators to optimize the selection of CHs. To further enhance the genetic algorithm, the tabu search technique is applied to explore unvisited areas of the solution space, improving the diversity of solutions and leading to better cluster stability. The hybrid approach brings several benefits. The combination of GA and TS enables the algorithm to quickly converge on optimal solutions without being stuck in suboptimal ones. This results in better cluster head lifetime, improved PDR, and reduced end-to-end delay. Simulations showed that the E-WKCA outperforms the original WKCA and other clustering algorithms regarding network throughput and stability, especially under high-density conditions.

Badole et al.[5], investigated another study that uses elite knowledge-based genetic algorithm, which introduces an innovative approach to improving multicast communication in VANETs by employing an enhanced genetic algorithm known as the Elite Knowledge Sharing Genetic Algorithm (EKSGA). The EKSGA algorithm identifies the top 5% of nodes with the highest fitness scores and uses their genetic information to create a new, more fit population. This approach significantly reduces computational overhead and improves network performance. This algorithm enhances the clustering process by optimizing the selection of cluster heads based on multi-objective fitness functions. These fitness functions consider factors such as energy distribution, inter- and intra-cluster distances, load balancing, and coverage. The EKSGA method reduces the need for excessive iterations by effectively using the top 5% of the population to generate more fit solutions, leading to improved network stability, increased network lifetime, and higher throughput compared to existing methods. The results demonstrated that EKSGA outperformed the methods compared, showing improvements in network lifetime and throughput. However, a notable limitation of EKSGA is the potential computational overhead due to the need to continually optimize the population with each iteration, which could impact performance in larger networks or under real-time constraints.

2.7 Hybrid Dog and Beluga Whale Optimization Algorithm

In another study conducted by Nithyanandam et al.[9], they used an alternate approach of using Hybrid Prairie Dogs and Beluga Whale Optimization-based Node Clustering (HPDBWOA-NC), by combining two optimization algorithms: the Prairie Dog Optimization Algorithm (PDOA) and the Beluga Whale Optimization Algorithm (BWOA). This hybrid algorithm effectively balances exploration and exploitation during the cluster formation process, thereby preventing premature convergence and ensuring optimal CHs selection. The simulation involved evaluating is focusing on metrics such as the number of clusters formed, mean throughput, PDR, energy consumption, and mean delay. The simulations used a freeway mobility mode. The results demonstrated that HPDBWOA-NC consistently outperformed benchmark methods by forming fewer clusters, optimizing resource use, and achieving higher throughput and PDR. Additionally, the mechanism showed reduced energy consumption and lower mean delay, indicating its efficiency and reliability in dynamic VANET environments. However, while the PDOA and BWOA provide robust solutions for preventing premature convergence, the method is highly sensitive to the configuration of the parameters governing the balance between exploration and exploitation.

2.8 Moth Flame Optimization (MFO) Algorithm

In another research conducted by Ramlee et al.[8], they introduced a hybrid approach that combines the Moth Flame Optimization (MFO) algorithm and K-Means clustering to improve the performance of VANETs by optimizing cluster formation and communication. The MFO algorithm mimics moths' behavior of flying towards flames to find optimal solutions, and in this context, it helps select the most suitable CHs based on fitness values. On the other hand, the K-Means algorithm groups vehicles into clusters based on their proximity, ensuring that each cluster has a well-placed CH that can efficiently manage intra-cluster communication. The MFO-K-Means hybrid system improved network performance by ensuring better energy distribution among cluster heads, minimizing the overhead associated with frequent cluster head re-selection, and providing scalability in dynamic traffic scenarios. By continuously updating CH positions according to vehicle movements, the approach leads to better packet delivery ratios, lower communication delays, and more stable clusters, especially in high-mobility scenarios. The combination of these two methods enhances clustering efficiency and improves network coverage. The limitation of this method is its sensitivity to increasing transmission range and node density, which can lead to overlapping clusters and interference. As more nodes join the network or vehicles move across different transmission ranges, the algorithm might experience difficulties maintaining optimal cluster formations, which could decrease communication efficiency.

Shah et al. [36], investigated the subject of clustering using MFO algorithm. The method takes inspiration from the natural navigation behavior of moths, where moths follow a spiral trajectory toward a light source. It applies this concept to optimize the selection of CHs and reduce the number of clusters. The algorithm initializes the positions of vehicles randomly within a defined grid and calculates the Euclidean distance between them to form clusters. Using the MFO algorithm, vehicles follow a spiral path toward potential CHs, continuously adjusting their positions based on fitness value. This fitness value is derived from vehicle speed, transmission range, and node proximity, which helps optimize cluster formation. As the algorithm iterates, the flames are updated, with the best solutions selected as CHs, ensuring that the number of clusters is minimized while maintaining stable and efficient communication. CAMONET's optimization process ensures robust clustering by dynamically adjusting to vehicle density and improving overall network performance. The CAMONET algorithm improves network stability and reduces routing costs by minimizing the number of clusters.

In another study Muhammad Fahad Khan et al. [1], introduced Moth Flame Clustering Algorithm for Internet of Vehicle (MFCA-IoV). The MFCA-IoV algorithm treats each vehicle in the network as a "moth," and the optimal CH is represented as the "flame." The moths move towards the flames by optimizing their positions based on a fitness function calculating the Euclidean distance between vehicles. The algorithm aims to reduce the number of clusters by selecting the optimal cluster heads that can handle a larger number of vehicles without compromising communication efficiency. The algorithm iteratively updates the positions of the moths and flames until an optimal cluster formation is achieved. The fitness function used in MFCA-IoV is multi-objective, considering the number of clusters and the distance between cluster members and their respective CHs. The main advantage of the MFCA-IoV is its ability to minimize the number of CHs, which reduces the overhead in managing the clusters. This leads to better energy efficiency, resource utilization, and network performance.

2.9 Ant Colony Optimization Using Fittest Node Clustering

Bijalwan et al. [15], investigated a novel approach that leverages a heuristic-based clustering algorithm integrated with an enhanced Ant Colony Optimization (ACO) technique. The study introduces the Dynamic Aware Transmission Range Parallel Euclidean Distance (DA-TRPED) method, which dynamically adjusts transmission ranges based on real-time parallel Euclidean distance calculations. By enhancing the traditional ACO algorithm with a dynamic evaporation factor, the proposed method addresses key challenges such as routing exploration time, network congestion, and the formation of stable clusters. The study also advances the ACO algorithm by introducing a dynamic evaporation factor, accelerating convergence and enhancing searchability, resulting in more efficient cluster formation. The proposed method improves the PDR, reduces packet drop rates, and lowers end-to-end delays compared to traditional ACO. It adapts dynamically to changes in vehicular density and transmission ranges, ensuring stable and efficient clustering in varying traffic conditions. The use of the PED metric allows for more accurate distance-based evaluations, improving CH selection. Despite its advantages one limitation is the reliance on accurate PED estimations, which can become less effective in scenarios with rapidly changing traffic conditions or irregular node distributions.

2.10 Hybrid Game-Theory-Based Clustering

In another research conducted by Alsarhan et al. [2], they introduced a method that integrates fuzzy logic and game theory to enhance decision-making in the CH selection process. Fuzzy logic ranks candidate CHs based on multiple factors: speed, signal strength, location, stability, and spectrum price. These factors are evaluated using a fuzzy inference system, which provides a ranking for the most suitable CH. In parallel, game theory is applied to manage the limited spectrum of resources. CHs compete to offer the best spectrum leasing prices to attract cluster members, creating an efficient and cost-effective spectrum-sharing system.

The hybrid method enhances network scalability and stability by dynamically adjusting to vehicular movement patterns and optimizing spectrum usage. The fuzzy logic ensures that only the most stable and efficient vehicles are selected as CHs. At the same time, game theory adds a competitive mechanism that helps balance the demand and supply of spectrum resources, ensuring more reliable communication. The game theory assumes that cluster heads can bid for spectrum in a competitive market. However, this can be problematic in real-world scenarios where regulations, sudden fluctuations in demand, or varying levels of interference might constrain spectrum availability. If market conditions are unstable, the efficiency of spectrum allocation could suffer.

Shivshankar et al. [37], proposed a method for cooperation in VANETs using Evolutionary Game Theory (EGT) combined with a Public Goods Game (PGG) model. This method focuses on stimulating cooperation among vehicles to improve packet forwarding. Instead of forcing cooperation, the model allows cooperation to evolve naturally through interactions influenced by network conditions like clustering, node density, and connectivity. The Public Goods Game (PGG) is employed to model the interactions between vehicles. In this game, cooperating vehicles contribute to a shared pool, and the benefits are distributed among all participants, including those who did not contribute. The evolutionary game dynamics allow vehicles to adjust their strategies based on the payoffs they receive from cooperating or defecting, with the goal of maximizing their benefits. Cooperation is encouraged by increasing the benefits shared among nodes. The EGT framework ensures that nodes can adapt to dynamic network conditions, evolving toward cooperation when it provides higher payoffs than defection. This method offers several advantages.

By using EGT and PGG, cooperation is not forced but evolves naturally, making it more sustainable in highly dynamic environments like VANETs. It adapts to different network topologies, such as urban or highway scenarios, by considering factors like node density and clustering.

2.11 Hybrid Fennec Fox and Sand Cat Optimization Algorithm

Meera et al. [38], introduced a novel Hybrid Fennec Fox and Sand Cat Optimization Algorithm (HFFSCOA). The proposed algorithm combines the strengths of the Fennec Fox Optimization Algorithm (FFOA) and the Sand Cat Optimization Algorithm (SCOA) to enhance cluster stability, reduce the number of clusters, and optimize the selection of CHs. The HFFSCOA introduces a new fitness function that considers grid size, node orientation, velocity, and communication range to ensure that only the most suitable vehicles serve as CHs. Additionally, the algorithm incorporates a comprehensive clustering strategy that optimizes the routing paths between vehicles, considering link stability, energy efficiency, and transmission range. The paper's contributions are validated under various VANET scenarios. The results demonstrate that HFFSCOA outperforms several existing clustering algorithms across key performance metrics such as the number of clusters generated, cluster stability, and network performance. The simulations show that HFFSCOA significantly improves cluster optimization rate and stability, making it a promising solution for enhancing the performance and reliability of VANETs. Despite its advantages, combining FFOA and ISCOA requires more processing power and time, which may limit the algorithm's applicability in real-time scenarios, especially in large-scale VANETs

2.12 Hybrid Customized Hunger's Foraging Honey Badger with Dynamic Multi-objective Non-Sorted Genetic Algorithm (CHFHB-DMNSGA)

The paper by Badole et al. introduced a new cluster-based VANET routing model that combines Customized Hunger's Foraging Honey Badger Optimization (CHFHB) and Dynamic Multi-objective Non-Sorted Genetic Algorithm (DMNSGA) [14]. The model optimizes multiple objectives such as PDR, end-to-end delay, throughput, and control packet overhead. CHFHB improves CHS by balancing exploration and exploitation, while DMNSGA enhances multi-objective optimization to meet VANET's dynamic requirements. The hybrid model effectively reduces routing overhead and delay and increases throughput and PDR by improving the selection of cluster heads and gateways. It also adapts to the dynamic nature of VANETs, ensuring optimal routing performance even in fluctuating network conditions. The results showed significant improvements in residual energy, alive node count, and convergence, which validate the efficiency of CHFHB-DMNSGA over other algorithms like NSGA-II and HBA. Despite its advantages, there is one limitation to this method. Computational complexity of the hybrid CHFHB-DMNSGA model involves multiple optimization stages. This could lead to higher processing time, particularly in large-scale networks, challenging real-time implementation.

Having discussed some of the novel and base approaches for clustering and cluster head selection in VANETs, we hope to shed light on the current state-of-the-art techniques in the field while highlighting the gaps and potential areas for improvement. To present an overview of all the discussed methods and their limitations, we have included a brief summary of the methods and their identified limitations in Table 2-1.

Table 2-1Comparison table listing some of the discussed clustering and cluster head selection methods in VANETs as well as listing their limitations

Proposed Method	Limitations
Weighted trusted formula [22], TVR (Trust based on Vehicles and Roadside units)[18], StabTrust mechanism [30], Fuzzy-Based Trusted Communication[31], A Trust-Based Clustering with Ant Colony Routing [6] Graph-Based Trust-Enabled Routing[23], Cluster based Trust Model[27]	Detection accuracy can decrease as the transmission range increases, which affects the ability to detect malicious nodes effectively. RSU infrastructure may not always be available or scalable in dense urban environments
Sugeno model fuzzy inference system [11], Enhanced Fuzzy Logic-based Clustering Scheme[25], fuzzy logic with the Bald Eagle Search [24]	Its difficulty in quickly adapting to frequent topology changes in dynamic VANET environments leads to potentially suboptimal cluster head selections. computational complexity.
RoVAN Scheme for Optimizing the Process of Cluster Head Selection [16]	High computational complexity from Rough Set Theory, sensitivity to data inaccuracies, scalability challenges in dense networks.
Double-Head Clustering (DHC) [20], Evolutionary Approach for Optimized VANET Clustering [32] Multi-Head Clustering Algorithm[17]	The complexity of maintaining several CHs. It may require significant computational resources.
Betweenness Centrality Based Clustering [12], [33], [34]	Calculating metrics such as betweenness centrality can be computationally intensive, especially in networks with high vehicle density.
Genetic Algorithm Based Clustering [5], [19], [26], [35]	Potential computational overhead due to the need to continually optimize the population with each iteration.
Hybrid Dog and Beluga Whale Optimization[9]	This hybrid algorithm is sensitive to the configuration of the parameters.
MFO Algorithm[8], [36], [1]	This algorithm is sensitive to increasing transmission range and node density.
Ant Colony Optimization Using Fittest Node Clustering[15]	Reliance on accurate PED estimations can become less effective in rapidly changing traffic conditions or irregular node distribution scenarios.
Game Theory Based Clustering[2], [37]	If market conditions are unstable, the efficiency of spectrum allocation could suffer.
Hybrid Fennec Fox and Sand Cat Optimization Algorithm[38]	Combining FFOA and ISCOA is time and power-consuming
CHFHB-DMNSGA)[14]	The computational complexity of the hybrid CHFHB-DMNSGA may lead to higher processing time.

CHAPTER 3 PROPOSED METHOD

3.1 Overview of the Proposed Method

This chapter presents the methodology employed to improve clustering and cluster head selection in VANETs. Our goal is to optimize the clustering process dynamically and ensure effective CH selection in a highly dynamic vehicular environment. We use a hybrid approach involving the K-Means algorithm and Silhouette Score to determine the optimal number of clusters in real time. The simulation uses SUMO and the Traci library for Python, allowing for real-time traffic simulation and vehicle interaction.

The methodology is structured into two main phases:

- **Phase 1: Clustering using K-Means and Silhouette Score**
- **Phase 2: Cluster Head Selection using a Weighted Formula**

3.2 Clustering Phase

The primary goal of clustering is to organize vehicles into groups (clusters) based on their current location and behavior on the road. This phase involves data collection, data preprocessing, and clustering. The K-Means algorithm is used to group vehicles into clusters, while the Silhouette Score determines the optimal number of clusters.

3.2.1 K-Means algorithm

The K-Means algorithm is widely recognized for its simplicity and effectiveness in clustering data, especially in dynamic environments like VANETs. The K-Means algorithm works by partitioning data points (vehicles) into k clusters based on their proximity to predefined centroids. The goal is to minimize the distance between each vehicle and its corresponding cluster centroid[8], [39].

To calculate the cluster centroids, K-Means uses an objective function that minimizes the sum of squared distances between each vehicle and its centroid. This function can be calculated as in Equation (3.1) [39]:

$$J = \sum_{i=1}^{k} \sum_{x \in C_i} \parallel x - \mu_i \parallel^2 \quad (3.1)$$

Here, x represents the position of a vehicle, C_i is the set of vehicles in cluster i, and μ_i is the centroid of that cluster. The algorithm iteratively recalculates the centroids by averaging the positions of all vehicles in each cluster, aiming to reduce the distance between vehicles and their respective centroids with each iteration. This process continues until the centroids stabilize, meaning there is minimal change in their positions, and the clusters are well-formed.

The K-Means algorithm must be adapted to handle the constantly changing network topology. Vehicles frequently move in and out of clusters, so the centroids need to be recalculated dynamically. Additionally, the algorithm must account for factors like vehicle's position and lane ID to maintain efficient communication within clusters, making K-Means a powerful tool for managing network traffic in such environments. In the traditional K-Means algorithm, the value of K (the number of clusters) is fixed and must be specified by the user before the algorithm begins. This means that the algorithm clusters the data into exactly K groups, regardless of the natural distribution of the data[39].

3.2.2 Silhouette Score Method

The Silhouette Score evaluates how well data points are grouped into clusters. It helps determine if the points within a cluster are similar and if different clusters are distinct. Essentially, the score gives us a sense of how well our clustering has performed, measuring how close each point is to others in its group and how far it is from points in other groups[40].

The score for each data point is computed based on two factors:

1. **Cohesion ($\boldsymbol{a(i)}$):** This measures how close a data point is to the other points within the same cluster. The closer the points are to each other, the more compact the cluster is. For each data point i, the cohesion score $a(i)$ is the average distance between i and all other points in its cluster. A smaller value of $a(i)$ indicates that the point is well-clustered with its peers, reflecting good intra-cluster compactness.

2. **Separation ($\boldsymbol{b(i)}$)**: This evaluates how well-separated the point is from other clusters. Specifically, $b(i)$ is the smallest average distance from the data point to all points in the nearest neighboring cluster. A larger value of $b(i)$ implies better inter-cluster separation, as the point is far from neighboring clusters.

Once these values are computed, the Silhouette score for each point $S(i)$ is calculated as Equation (3.2)[40]:

$$S(i) = \frac{b(i) - a(i)}{\max\big(a(i), b(i)\big)} \tag{3.2}$$

The Silhouette score ranges from -1 to 1:

- A score close to 1 indicates that the point is well-clustered, meaning it is close to the points in its cluster and far from points in other clusters.
- A score around 0 suggests that the point lies on or near the boundary between clusters.
- A negative score means the point is likely assigned to the wrong cluster, as it is closer to points in a different cluster than to those in its own cluster.

3.2.3 The Proposed Clustering Architecture

In the proposed method, we aim to combine the K-Means algorithm with the Silhouette method to determine the optimal number of clusters, ensuring well-separated and cohesive groupings within the data. This flowchart represents the process of clustering vehicles in VANET using the K-Means algorithm and the Silhouette Score Method. The flowchart of this process is shown in Figure 3-1.

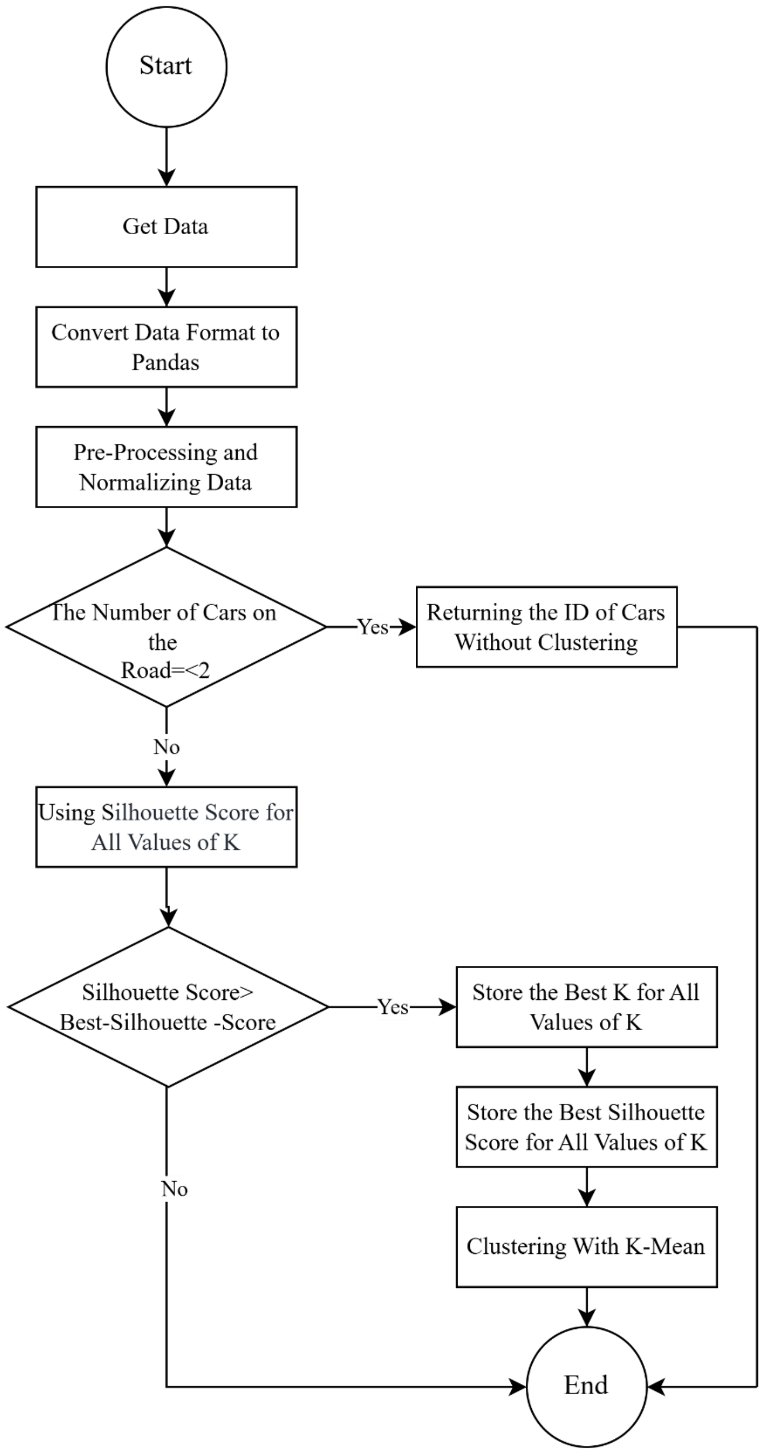

Figure 3-1Flowchart Illustrating the Flow of Our Clustering Process

The process begins by initializing the clustering mechanism. At this step, real-time data on vehicles is collected from the simulation. The data includes vehicle location (X, Y coordinates) and lane ID. This information is crucial for clustering as it defines the current state of the network. Once the data is collected, it is converted into a format that can be processed efficiently. In this case, the Pandas library structures the data into a DataFrame, making manipulating and analyzing the vehicles' information easier.

Each vehicle's position and speed are normalized using the Standard Scaler from Scikit-learn to ensure uniform scaling across all features.

Once the data is preprocessed, we apply the K-Means algorithm to cluster the vehicles. The challenge here is selecting the optimal number of clusters, denoted by K. To solve this, we use the Silhouette Score to evaluate how well the vehicles fit within each cluster.

The silhouette score measures the clustering quality by comparing the intra-cluster distance with the nearest-cluster distance. We run K-Means multiple times with different values of K, ranging from 2 to 11, and calculate the silhouette score for each iteration. We select the value of K that maximizes the silhouette score, ensuring that the clusters are well-formed and stable. The process continues iteratively, and in each iteration, K is optimized.

3.3 Cluster Head Selection Phase

Once the vehicles are clustered, we must select a Cluster Head (CH) for each cluster. The CH manages communication within the cluster and with other clusters or RSUs.

3.3.1 Weighted Formula for Cluster Head Selection

The cluster head is selected based on a weighted formula that considers several key factors, which are calculated for each vehicle in the cluster. These factors include:

- **Local Distance (LD):** Proximity to other vehicles within the cluster.
- **Relative Speed (RS):** Speed relative to other vehicles in the cluster.
- **Trust (T):** Reliability in forwarding data efficiently.
- **Base Station Distance (BSD):** Distance to the nearest base station or RSU.
- **Answer Ratio (AR):** Success rate in data communication.
- **Node Degree (ND):** The number of connections a vehicle has within the cluster.
- **Node Center (NC):** The centrality of the vehicle in the cluster.

The weighted formula for CH selection is given by Equation (3.3) :

$$CH = W_{LD}.LD + W_{RS}.RS + W_{T}.T + W_{BSD}.BSD + W_{AR}.AR + W_{ND}.ND + W_{NC}.NC \qquad (3.3)$$

where the sum of all weights equals 1. The vehicle with the highest CH Score in each cluster is selected as the Cluster Head.

Each of these factors plays a vital role in ensuring the selected cluster head can manage the cluster effectively while minimizing cluster head changes, reducing cluster reformation, and extending the network's lifetime. Local Distance (LD) is critical in selecting a cluster head that is centrally located within the cluster, which helps maintain strong communication links and reduces the likelihood of vehicles drifting apart, thus minimizing the need for frequent cluster head changes. Relative Speed (RS) is equally important, as selecting a cluster head that moves at a similar speed to other vehicles in the cluster ensures that the cluster remains intact for longer periods. This directly reduces cluster reformation and the need to re-elect cluster heads, contributing to overall network stability. Additionally, Trust (T) ensures the reliability of the selected cluster head by prioritizing vehicles with a proven track record of efficient data handling. A trusted vehicle is less likely to cause disruptions, thereby maintaining stable communication within the cluster and reducing unnecessary cluster head replacements.

Base Station Distance (BSD) affects energy consumption and communication efficiency. A cluster head closer to base stations consumes less energy when communicating with infrastructure, allowing it to conserve resources and remain operational longer. Answer Ratio (AR) is essential in identifying vehicles capable of handling communication efficiently without overwhelming the cluster head. This, in turn, extends the cluster head’s operational time by avoiding overburdening it with data requests, which would otherwise shorten its lifespan. Similarly, Node Degree (ND), or the number of direct connections a vehicle has within the cluster, ensures that the selected cluster head can manage multiple connections efficiently, reducing energy consumption and prolonging the network's overall lifetime. Finally, Node Center (NC) emphasizes the selection of a vehicle positioned centrally within the cluster, ensuring efficient communication with all members, reducing unnecessary energy expenditure, and ultimately supporting longer-lasting network operations. By carefully balancing these factors, the network can maintain stable clusters, minimize the frequency of cluster head and cluster changes, and extend the overall lifetime of the network.

3.3.2 Metrics Calculation

Each of the above metrics is calculated as follows:

- **Local Distance (LD)**: This is the average distance between the vehicle and all other vehicles in the cluster Equation (3.4)[22]:

$$LD_i = \sum_{v \in C} \sqrt{(x_i - x_v)^2 + \left(y_i - y_v\right)^2} \tag{3.4}$$

The formula calculates the Euclidean distance between the two vehicles i and v where vehicle i and v have locations of attributes $(x_i\ , y_i)$ and $(x_v\ , y_v)$ respectively. A vehicle with a smaller total distance is considered more central and is therefore a better candidate for the CH role.

- **Relative Speed (RS)**: Relative Speed measures how close a vehicle's speed is to the average speed of other vehicles in the cluster. A vehicle with a similar speed to the rest of the cluster is less likely to cause frequent cluster reformation due to sudden speed changes. Relative speed is calculated by summing the absolute differences in speed between the given vehicle and each of the other vehicles. The formula for the relative speed can be expressed as Equation (3.5)[22]:

$$RS = \sum_{v \in vehicle_{ids}} |speed(vehicle_n) - speed(vehicle_v)| \tag{3.5}$$

- **Trust (T)**: Trust reflects the performance of vehicular nodes in the network. When a vehicle behaves cooperatively by forwarding packets with minimal delays, it earns a higher trust value compared to non-cooperative vehicles that introduce delays in packet transmission. The trust value is divided into two components: Direct Trust and Indirect Trust. Trust is calculated based on Equation (3.6)[22]:

$$T = \lambda \times T_{direct} + (1 - \lambda) \times T_{indirect} \tag{3.6}$$

This formula combines both direct and indirect trust. The value of λ balances the importance of a vehicle's own behavior (direct trust) and the perspective of other vehicles (indirect trust). A higher trust score means that the vehicle is more reliable and a better candidate for CH

Direct trust is calculated based on a vehicle's past success in forwarding packets within the cluster Equation (3.7)[22]:

$$T_{direct} = \frac{Packets\ forwarded\ successfully}{Total\ packets\ sent} \tag{3.7}$$

Indirect trust is the trust a vehicle has from the perspective of neighboring vehicles. It is calculated based on the communication behavior of the neighboring vehicles and how much they trust the current vehicle based on their interactions (3.8) [22] :

$$T_{indirect} = \sum_{v \in C} T_{direct}(v).D(i, v) \tag{3.8}$$

Base Station Distance (BSD): Base Station Distance measures the Euclidean distance between a vehicle and the nearest base station. Vehicles closer to a base station are preferred because they can act as reliable intermediaries between vehicles and infrastructure Equation (3.9)[11] :

$$BSD_i = \sqrt{(x_i - x_{BS})^2 + (y_i - y_{BS})^2} \tag{3.9}$$

- **Answer Ratio (AR)**: The ratio of successful responses to the total number of requests sent by the vehicle. A higher answer ratio indicates a more reliable vehicle for inter-vehicle communication Equation (3.10)[31] :

$$AR_i = \frac{Packets\ received}{Packets\ sent} \tag{3.10}$$

- **Node Degree (ND)**: Node degree is the number of direct connections a vehicle has with others in its communication range Equation (3.11)[32]:

$$ND_i = \sum_{v \in C} \mathbb{I}(D(i, v) \leq R) \quad (3.11)$$

- **Node Center (NC)**: In our project, we designed a two-way road with three lanes. The vehicles that are in the middle lane out of the three lanes are not likely to exit the designed street (those on the right side are usually likely to exit the street sooner or later) and are not likely to overtake. Because the vehicles of this line are usually more stable than the cars of other lines, it is likely that if they are chosen as the cluster head, they will lose this cluster head position late.

3.4 Real-World Considerations and Challenges in VANET Clustering and Information Transfer

3.4.1 Real-World Data Collection and Communication Framework

In real-world VANETs, clustering implementation relies heavily on practical data acquisition and communication frameworks. Unlike the controlled environment of simulations, real-world clustering involves significant variability in vehicle density, mobility patterns, and environmental factors. Vehicles collect critical clustering parameters—such as position, velocity, and direction—through integrated sensors and Global Positioning System (GPS) modules. This data is communicated via V2V and V2I communication protocols.

- **Role of Infrastructure:** RSUs and base stations facilitate clustering in long-distance scenarios by bridging connectivity gaps and aggregating vehicular data over broader areas.
- **Dynamic Information Sharing:** Vehicles exchange status and clustering data periodically via beacon messages. However, ensuring low latency and reliability is critical for real-time updates in high-mobility environments, especially in urban areas with dense traffic.

3.4.2 Estimating the Number of Clusters in Real-World Scenarios

The number of vehicles in the simulation is predefined, and a range for K is specified before clustering. However, real-world scenarios lack this certainty. Adaptive methods to estimate the optimal number of clusters are crucial for effective clustering in dynamic traffic environments.

1. **Dynamic Adjustments Using Localized Information:**
   Vehicles independently monitor their local density and proximity to neighbors, determining a rough estimate for the optimal number of clusters. This distributed approach reduces reliance on centralized systems and increases scalability.
   Clustering algorithms like dynamic K-Means or hierarchical clustering can adjust K based on real-time vehicle density metrics.

2. **Utilizing Roadside Infrastructure:**
   RSUs and base stations gather data from connected vehicles to calculate cluster parameters dynamically. These values can be broadcast to vehicles, helping to establish efficient clusters in long-distance and high-density scenarios.

3. **Machine Learning Models for Cluster Prediction:**
   Advanced machine learning techniques can be applied to predict the optimal number of clusters by analyzing traffic flow data and historical vehicular movement patterns. These models improve responsiveness to fluctuating vehicle counts in highways and urban grids.

3.4.3 Long-Distance Clustering Challenges

Clustering vehicles over extended geographical areas introduces unique challenges not present in simulation environments:

- **Connectivity Stability:** Ensuring vehicles within a cluster maintain consistent communication becomes increasingly difficult as vehicle density fluctuates. This is particularly challenging in rural highways with low vehicle density or urban intersections with extremely high density.

- **Inter-Cluster Communication Overhead:** In real-world deployments, communication between distant clusters often relies on RSUs or base stations. This introduces potential bottlenecks in data relay and processing, especially in areas with limited infrastructure coverage.

- **Energy Efficiency Considerations:** Clustering protocols' energy efficiency becomes crucial in long-distance scenarios, particularly for electric vehicles. Reducing communication overhead while maintaining reliable connections is essential for sustainable VANET deployments.

3.4.4 Adaptive Techniques for Real-World Cluster Formation

Given that real-world vehicular traffic is highly dynamic, predefining a range for K, as done in simulations, is not practical. The following approaches address this limitation:

- **Real-Time Adjustments Based on Traffic Density:** The clustering algorithm iteratively evaluates traffic density metrics to adjust K. For instance, if vehicles experience rapid changes in density due to merging traffic, the algorithm recalibrates cluster boundaries to maintain stability.

- **Hierarchical Clustering Frameworks:** Hierarchical clustering, supported by RSUs, can dynamically merge or split clusters based on vehicular movement trends. This ensures stability across long distances while minimizing the need for frequent re-clustering.

- **Incorporation of Silhouette Score in Real-Time:** Vehicles can dynamically adjust the clustering process to improve cohesion and separation in real-world deployments by continuously evaluating the silhouette score.

3.5 Pseudo Code of our Cluster Head Selection and Clustering

The pseudocode in Figure 3-2 below concludes the algorithm followed throughout the Cluster Head Selection and Clustering simulation. As seen in line 4 of the pseudo code, the algorithm begins by setting up a time counter, which updates at each simulation step. At each step, it gathers information on every active vehicle's position and lane ID. Lane position plays an essential role here, as vehicles in the middle lane are less likely to change lanes frequently, making them more stable and ideal candidates for leading a cluster. This stability is particularly valuable, as it helps keep clusters intact over time, reducing the need for constant reorganization.

As seen in line 11, the collected data is then formatted in Pandas, which makes it easier to manage and analyze. It is also standardized using a scaling technique in line 12. Normalizing this data is critical, ensuring that each attribute contributes evenly to the clustering process, preventing any attribute from dominating. With this cleaned data, in line 14, the algorithm uses the K-Means clustering method to group vehicles into clusters. It tries out different numbers of clusters and calculates a silhouette score for each configuration as demonstrated in line 16. This score measures how well each vehicle fits within its assigned cluster compared to others, helping the algorithm pick the most effective number of cohesive and distinct clusters.

In line 20, once clusters are in place, the algorithm selects a cluster head within each group. As seen in line 24, this leader is chosen based on a weighted score, which considers multiple factors such as proximity to other vehicles (local distance), similarity in speed, trustworthiness in data handling, closeness to a base station, communication reliability (answer ratio), number of direct connections (node degree), and lane position (node center). Among these factors, lane position is especially crucial; vehicles in the middle lane are favored as they are less likely to veer off, ensuring stable leadership within the cluster.

Each factor plays a distinct role in determining the most suitable cluster head: local distance favors vehicles that are centrally located within the cluster; relative speed prefers vehicles moving at a similar pace to others, helping maintain group cohesion; trust prioritizes vehicles with reliable data handling; base station distance favors those closer to infrastructure, reducing energy costs for communication; answer ratio values vehicles with a high rate of successful communication; node degree accounts for connectivity within the cluster; and node center gives preference to stable middle-lane vehicles. The vehicle with the highest combined score is chosen as the cluster head.

Throughout the simulation, the algorithm monitors the stability of cluster heads and tracks changes in cluster membership. As seen in line 33, whenever there is a shift in cluster leadership or membership, it updates counters and notes each cluster's operational lifespan. This tracking helps the algorithm evaluate how often clusters need to be reorganized, a key indicator of stability in a dynamic environment.

Moreover, in line 35, the algorithm monitors clusters that lose all their members, marking them as inactive and recording their lifetimes. At the end of the simulation, the algorithm provides statistics on the number of cluster head changes, membership changes, and each cluster's lifetime in line 38. By favoring vehicles in the middle lane and adapting clusters based on silhouette scores, this approach aims to create stable, long-lasting clusters. This minimizes the need for re-clustering and ensures continuous, reliable communication even in rapidly changing traffic conditions.

Algorithm 1 Cluster Head Selection and Clustering

Input:
Vehicle_list ▷ Total number of active vehicles
Weights ▷ Our weights are $W_{\mathrm{LD}}, W_{\mathrm{RS}}, W_T, W_{\mathrm{BSD}}, W_{\mathrm{AR}}, W_{\mathrm{ND}}, W_{\mathrm{NC}}$
Output:
Divide clusters with cluster heads
Evaluation Measures:
Cluster head changes, cluster changes, lifetime of clusters
Step 1: Initialize Time Step Counter
$t \leftarrow 0$ ▷ Starting time
while simulation is running **do**
  Step 2: Update Time Step
  $t \leftarrow t+1$ ▷ Update time step
  Step 3: Normalizing Data of Vehicles
  Get position and lane ID of vehicles
  Convert data to Pandas format
  Normalize data using StandardScaler
  Step 4: Clustering the Vehicles on the Road
  for each $k = 1$ to ClusterNumber **do**
    Perform K-Means
    Perform Silhouette Score ▷ *Silhouette Score* $= \frac{b-a}{\max(a,b)}$
  end for
  Store the k with the best Silhouette Score
  Perform clustering
  Step 5: Cluster Head Selection
  for each cluster in clusters **do**
    for each vehicle in the cluster **do**
      Calculate local distance, relative velocity, trust, base station distance, answer ratio, node degree, node center
      $PCH = W_{LD} \times LD + W_{RS} \times RS + W_T \times T + W_{BSD} \times BSD + W_{AR} \times AR + W_{ND} \times ND + W_{NC} \times NC$
    end for
    Select vehicle with the highest PCH as cluster head
  end for
  Step 6: Track Cluster Head and Cluster Changes
  if cluster head changes **then**
    Increment cluster head changes
  end if
  if cluster changes **then**
    Increment cluster changes and update lifetime
  end if
  Check for destroyed clusters and update lifetime
end while
End while:
Return final statistics: cluster head changes, cluster changes, lifetime of clusters over time

Figure 3-2Pseudocode of the proposed Cluster Head Selection and Clustering method

CHAPTER 4 EXPERIMENTAL RESULTS AND DISCUSSION

In this section, we compare the performance of our proposed hybrid clustering approach called "An Improved Method to Increase Cluster Lifetime in Vehicular Ad Hoc Networks (VANETs) (IMICLiVAN)" with three existing methods: WTCHS, ECBLTR, and EKSGA. We have selected to compare our method with the existing methods as these are the current state-of-the-art approaches in the field and can be considered solid baseline methods for performance analysis. Furthermore, each uses a distinct clustering selection approach, which could potentially inform us of the most effective approach. The evaluation focuses on three key metrics: Network Lifetime, Cluster Changes, and Cluster Head Changes. The results are also supported with visual data presented in charts to provide a clear understanding of the improvements achieved by the IMICLiVAN method.

4.1 Clustering Methodology

The clustering methodology combines K-means clustering with a silhouette score method, enabling dynamic and optimized grouping of vehicles. The core of this process includes the following:

- **Dynamic Cluster Formation with K-means:** The K-means algorithm is applied to group vehicles based on their position and lane ID. Each vehicle's data, such as coordinates and current velocity, is processed in real time through SUMO's Traci library in Python, which retrieves vehicle states at each simulation step. The optimal number of clusters is determined by evaluating silhouette scores across different K values, ensuring well-separated and cohesive clusters that adapt to vehicular movement changes.

- **Silhouette Score Calculation:** The **s**ilhouette scoring evaluates the quality of clusters by balancing intra-cluster compactness with inter-cluster separation. This score assists in dynamically adjusting the number of clusters based on vehicle density and movement patterns, which helps prevent over-clustering and maintains robust communication structures.

4.2 Cluster Head Selection Methodology

The cluster head (CH) selection algorithm dynamically evaluates each vehicle's suitability to serve as the CH by calculating a weighted score from multiple factors that impact stability and connectivity within the cluster. Each factor is weighted, summed, and compared across vehicles to select the highest-scoring vehicle as the CH, thus ensuring stable cluster communication with minimized re-clustering requirements. This selection process is iteratively performed in each simulation step, updating CHs as vehicles move in and out of clusters.

4.3 Simulation Environment

One point of discussion that should be mentioned is the choice of simulation environment. SUMO software was used as the primary simulation environment for all our experiments, and it was integrated with Python via the Traci library for real-time control and management of vehicular traffic. SUMO offers detailed models for vehicular mobility, which is essential for simulating highly dynamic environments like VANETs [17]SUMO was chosen over other simulators because of its scalability, high precision in vehicle behavior modeling, and open-source nature, which allows extensive customization.

We used Python for its flexibility and ease of integration with SUMO. The use of Traci allowed for real-time control of vehicles during the simulation, enabling dynamic decision-making processes such as cluster formation and cluster head selection. This framework was ideal for testing various VANET routing protocols in a controlled but realistic environment. Regarding simulation time, Python's integration with SUMO offers notable advantages. Python, mainly when used with libraries like NumPy and Pandas, handles large datasets efficiently and speeds up processes like real-time cluster formation and decision-making during simulations. This is crucial for handling hundreds of vehicles without significant delays. Compared to OMNeT++ or MATLAB, the Python-SUMO combination typically results in shorter simulation times.

OMNeT++ is a widely used simulator for network communication[17]. However, when vehicular mobility simulation is needed, Veins integrates SUMO with OMNeT++ to handle vehicle movement realistically. While this combination is powerful, it requires managing two simulators simultaneously (SUMO for traffic and OMNeT++ for communication), which increases complexity and setup time compared to directly using SUMO with Python for mobility control and real-time decisions. This results in longer preparation times and higher simulation overhead when compared to SUMO-Python integration, which offers a more streamlined process for vehicular simulations.

For MATLAB-based simulators [16], while strong in terms of network algorithm simulations, their mobility modeling is less detailed. MATLAB often requires the development of custom vehicle movement models or integration with external datasets, making it less suited for simulating real-world traffic dynamics when compared to SUMO, which has a built-in capacity for simulating urban traffic, intersections, and various modes of transport . As a result, SUMO with Python offers faster and more realistic simulations, reducing preparation and running time.

4.4 Implementation Details

The simulation environment in SUMO was constructed to replicate a realistic road network where vehicles move dynamically. This setup involved defining nodes, edges, routes, and a base station, enabling vehicles to navigate through the network and communicate with each other.

Node and Edge Configuration:

- Each node represents intersections or key locations where vehicles can enter or exit. Two nodes were configured at (0.0, 0.0) and (1000.0, 0.0) to form the endpoints of the network
- Edges represent the roadways connecting these nodes. A three-lane, bidirectional road connects nodes 1 and 2, allowing vehicles to travel in both directions, essential for testing cluster dynamics and vehicle interactions.

Route Configuration:

The routes for vehicles were defined in route.xml, specifying the sequence of edges each vehicle follows, along with staggered departure times. This setup simulates various levels of traffic density and movement patterns, providing a robust test for the clustering and cluster head algorithms under different network conditions.

Key Simulation Parameters:

The following parameters are configured for the simulation:

- **Communication Range**: 250 meters
- **Cluster Distance**: 45 meters
- **Neighber_Distance_Threshold** :100 meters

4.5 Cluster Lifetime

The cluster lifetime refers to the duration for which a cluster remains stable without major re-clustering events or cluster head changes. This metric is critical for maintaining network efficiency and reducing communication overhead in VANETs. In Figure 4-1, we compared the cluster lifetime for each method. We simulated over 200 iterations to present a basis for comparison for this specific metric. As shown in Figure 4-1, the IMICLiVAN clustering algorithm significantly outperforms existing methods in terms of network lifetime. The IMICLiVAN hybrid method, which combines K-Means clustering with the Silhouette Score for optimal cluster formation, achieved a cluster lifetime of 5.78 rounds. This represents a significant improvement over WTCHS, which recorded only 2.01 rounds, ECBLTR with 4.65 rounds, and EKSGA, which achieved 1.03 rounds.

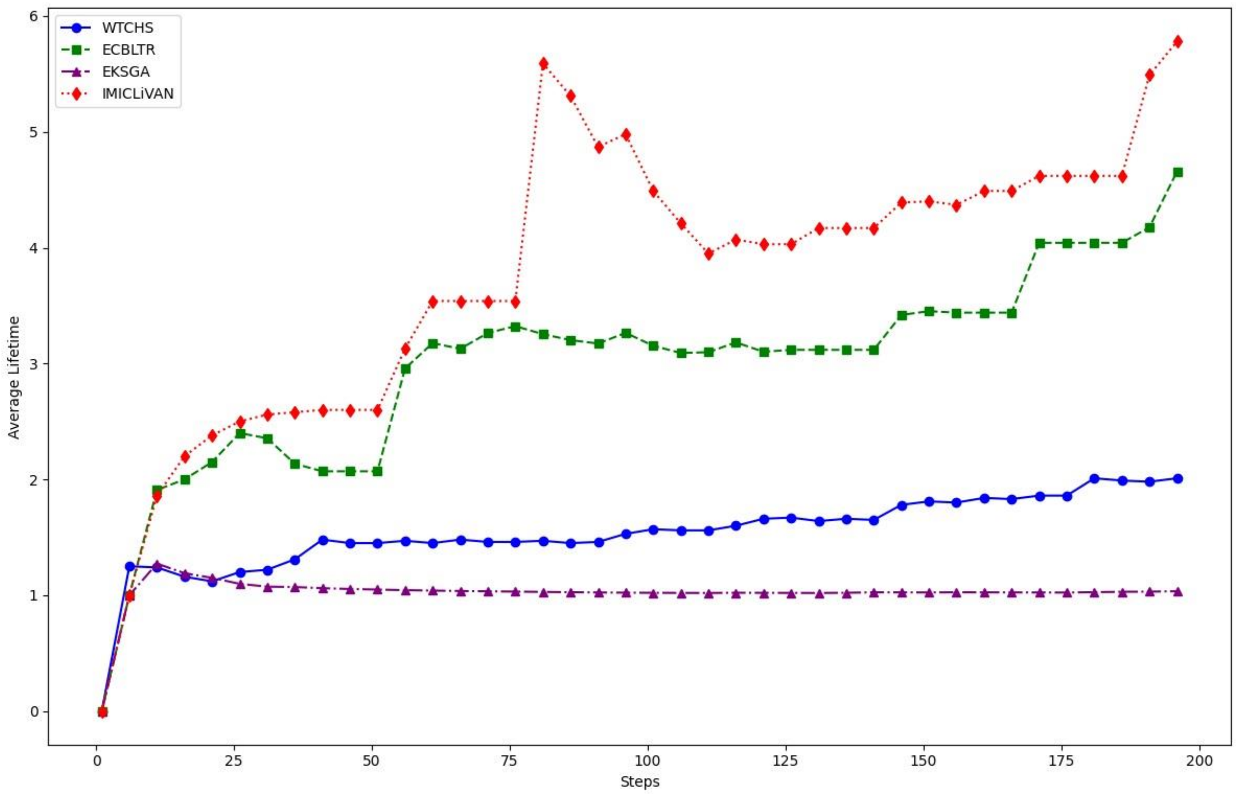

Figure 4-1Comparison of the Lifetime of IMICLiVAN with WTCHS, ECBLTR, and EKSGA

The K-Means algorithm dynamically adapts to node mobility, ensuring the clusters remain stable and cohesive for longer durations. Since we minimized clustering changes using the K-Means algorithm and Silhouette Score, it is evident that during the simulation, the lifetime of the clusters also increased, leading to a higher average lifetime. By reducing unnecessary re-clustering, the IMICLiVAN method enhances network stability, critical in highly dynamic VANET environments where vehicle positions and velocities change frequently.

The IMICLiVAN method integrates several factors—particularly Base Station Distance and Answer Ratio—to optimize cluster head selection, ensuring that chosen cluster heads consume minimal energy when communicating with infrastructure. This design choice enables the network to function for longer periods without frequent re-clustering or cluster head replacements. In contrast, WTCHS primarily focuses on trust as the key metric, leading to a significant energy drain caused by constant trust recalculations and trust-based re-clusterings. While important for data reliability, the emphasis on trust in WTCHS often overlooks energy efficiency, particularly when cluster heads are located farther from the base station. As a result, the network lifetime in WTCHS is shorter, as CHs expend more energy maintaining long-distance communications with base stations.

The ECBLTR method improves energy efficiency by incorporating a fuzzy inference system that accounts for node degree and residual energy. However, fuzzy logic tends to slow down decision-making in highly dynamic environments, particularly when there is rapid fluctuation in vehicle density. This lag in response time leads to unnecessary energy consumption as the system struggles to adapt quickly enough to changing conditions. In contrast, the K-Means clustering algorithm used in the IMICLiVAN method adapts more efficiently to these dynamic conditions, ensuring that clusters are formed optimally and maintained with minimal overhead, thereby extending the network lifetime.

The EKSGA algorithm, which uses genetic optimization to select cluster heads, performs well in conserving energy by choosing the best-performing nodes. However, the genetic algorithm's iterative nature introduces additional computational overhead, particularly when it continuously recalibrates the network to identify the fittest nodes. This frequent recalibration, especially in high-mobility environments, contributes to energy wastage and shortens network lifespan. By contrast, the IMICLiVAN method avoids such overhead by dynamically adjusting clusters in real-time using a weighted selection process, which is less computationally demanding and more energy efficient. As a result, the network lifetime in the IMICLiVAN method outperforms both EKSGA and WTCHS, while demonstrating greater adaptability and efficiency compared to ECBLTR.

The IMICLiVAN method improves cluster lifetime by 24% compared to ECBLTR, and by 187% compared to WTCHS. It performs nearly 461.16% better than EKSGA, demonstrating its superior ability to maintain stable clusters over extended periods.

4.6 Cluster Head Changes

Cluster head (CH) changes reflect the frequency with which the cluster head is reassigned. Fewer CH changes indicate better cluster stability, reducing communication delays and overhead. We have compared the performance of the methods on this metric over 200 iterations and presented the results in Figure 4-2. As observed from Figure 4-2, the IMICLiVAN method achieves a reduction in cluster head changes, with a total CH change number equal to 86. This is a significant improvement compared to 422 CH changes in EKSGA, 139 CH changes in ECBLTR, and 257 CH changes in WTCHS.

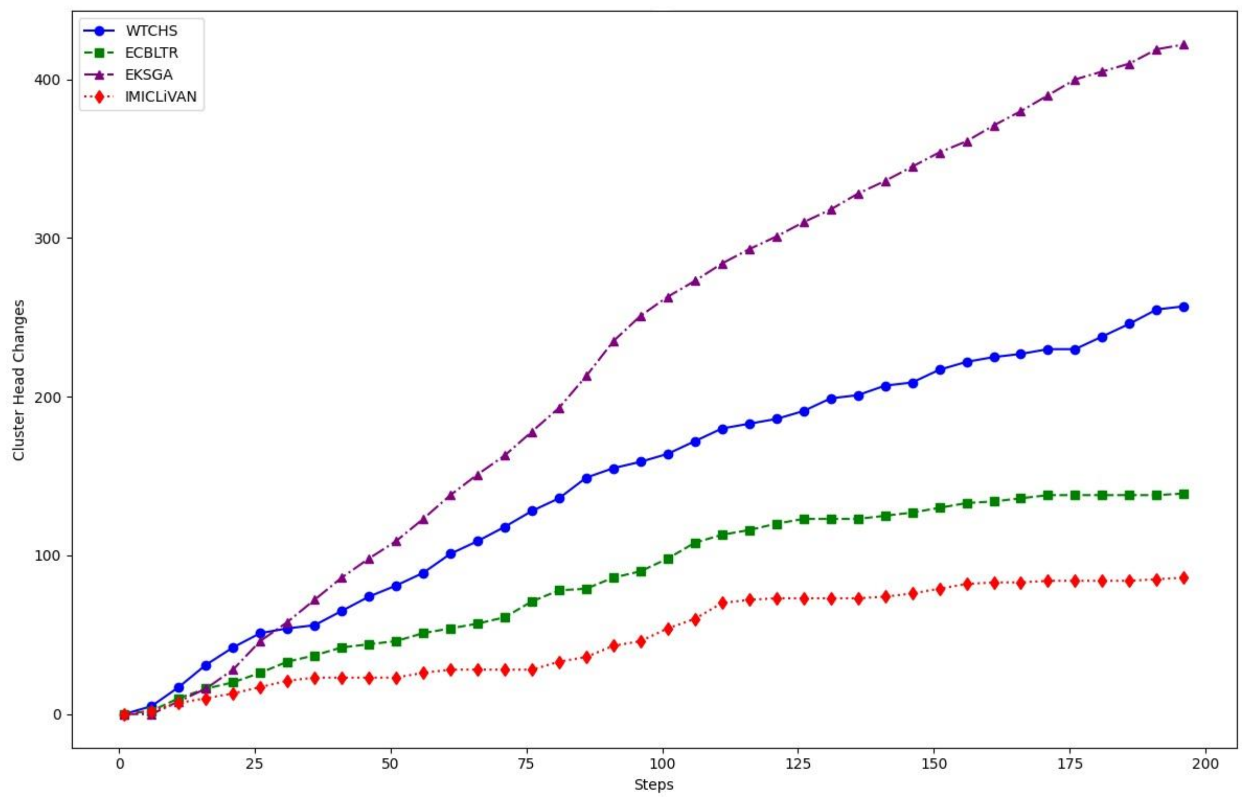

Figure 4-2Comparison of the Cluster Head Changes of IMICLiVAN with WTCHS, ECBLTR, and EKSGA

The frequency of cluster head changes is a critical factor that affects network performance. Frequent changes in cluster heads disrupt communication, increase delays, and raise the overall control overhead. The IMICLiVAN method effectively addresses this issue by considering a balanced set of criteria for selecting cluster heads. Our method proved to be superior to others for several reasons. First, when we begin with improved clustering that has minimal changes, the selected cluster heads will also be better due to the stable clustering structure. The second point is that we considered seven factors to ensure the best cluster head selection, unlike the three factors used in the WTCHS paper. As a result, the weights in our formula are distributed more evenly, and in situations where there are slight changes in vehicles with each update, the likelihood of a single metric changing and causing the cluster head to shift from one vehicle to another is significantly reduced, since we have more metrics for evaluating cluster head candidates.

In contrast, WTCHS experiences frequent CH changes due to its overreliance on trust scores. In dynamic VANET environments, trust scores can fluctuate rapidly, leading to constant re-evaluations and re-selections of cluster heads. This trust-based volatility often results in CHs being replaced prematurely, increasing control message overhead and degrading overall network stability. The IMICLiVAN method avoids this issue by balancing trust with other important metrics, ensuring that CHs are not selected or replaced based on volatile factors alone.

The ECBLTR method also faces issues with cluster head changes, as its reliance on fuzzy logic for CH selection introduces delays in decision-making, mainly when vehicle topology changes quickly. CHs are replaced more frequently than necessary as the fuzzy system struggles to adapt in real time. By contrast, the IMICLiVAN method’s weighted formula for CH selection ensures that CHs remain stable for longer periods, as it continuously monitors the relative positions, speeds, and energy levels of vehicles, avoiding the slow recalibration issues present in fuzzy systems.

Finally, EKSGA, while improving CH selection through genetic algorithms, suffers from frequent CH replacements due to its iterative optimization process. The genetic algorithm continuously seeks to identify the fittest vehicle to serve as CH, which can lead to frequent replacements as new "elite" vehicles are identified. This frequent switching increases network instability and control overhead. The IMICLiVAN method, with its more direct and responsive approach, minimizes CH changes by selecting cluster heads based on multiple stable factors, ensuring that once a vehicle is chosen as CH, it remains in that position for longer periods, resulting in fewer disruptions and more stable communication.

The IMICLiVAN method reduces CH changes by 38% compared to ECBLTR, 79.6% compared to EKSGA, and by 66.53% compared to WTCHS, demonstrating its superior ability to maintain cluster stability and reduce communication overhead.

4.7 Cluster Changes

Cluster changes measure how frequently a cluster needs to be reorganized or reformed due to shifts in vehicle positions or cluster head changes. Fewer cluster changes mean greater stability and less overhead in managing the dynamic nature of the network. Since this is an essential metric in comparing clustering performance, we calculated this value for each tested method over 200 iterations and compared the results in Figure 4-3. As Figure 4-3 demonstrates, the IMICLiVAN method reduces cluster changes, with only 80 cluster changes in total. This number is considerably

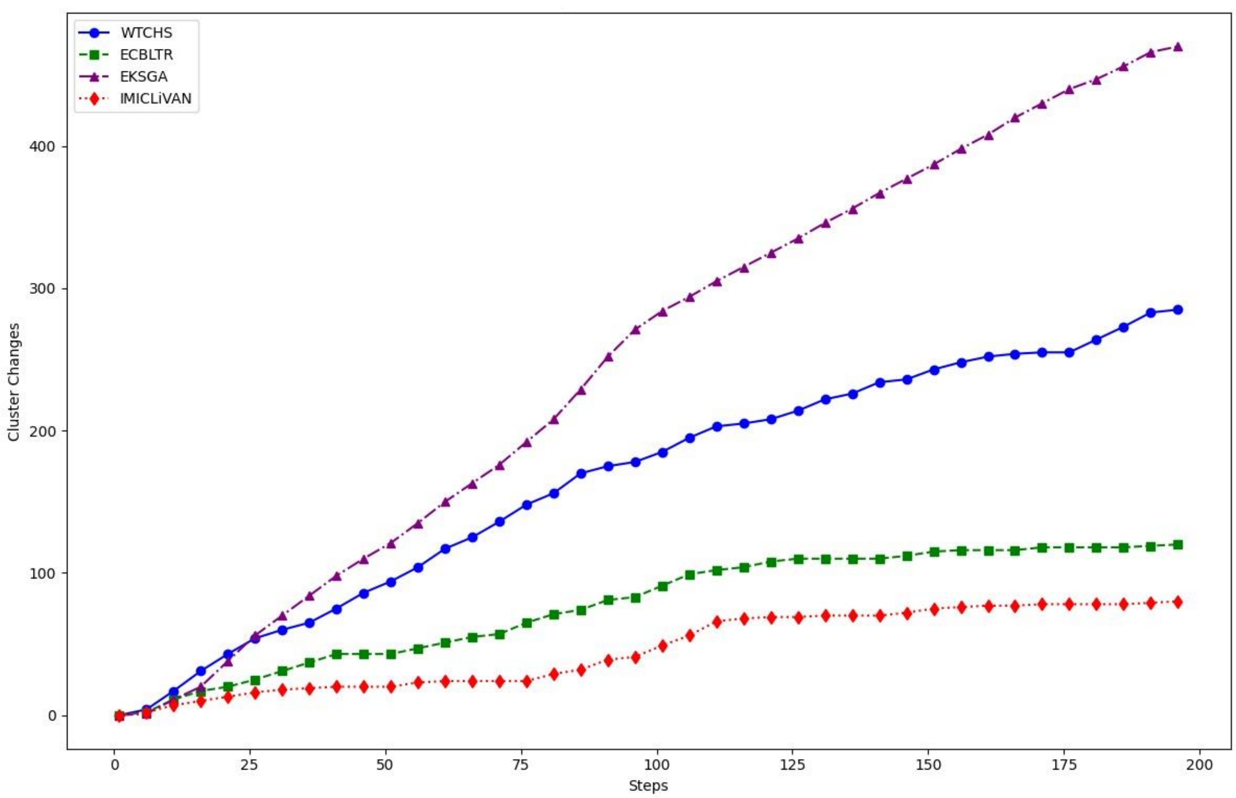

Figure 4-3 Comparison of Cluster Changes of IMICLiVAN with WTCHS, ECBLTR, and EKSGA

lower when compared to 285 changes in WTCHS, 120 changes in ECBLTR, and 470 changes in EKSGA.

In our clustering process, where we utilized the K-Means method with Silhouette Score, the optimal number of clusters, k, is initially determined by this method. For each value of k (ranging from 2 to 11), the clustering is performed, and the best value of k is returned in each iteration. This approach ensures that we achieve the optimal number of clusters in every iteration, which is highly impactful for our clustering process, as the clusters are formed in their most optimal state. Consequently, the likelihood of changes in clustering is lower than that of other methods.

Initially, we considered several factors in the clustering process, such as vehicle lane ID, location, and speed. However, since the differences in vehicle speed disrupted the clustering order and had a stronger effect than location, we refined the clustering process by prioritizing location and vehicle IDs as the main elements. This stands in contrast to WTCHS, where trust and velocity are overemphasized, often leading to frequent cluster fragmentation as vehicles with fluctuating trust scores are moved in and out of clusters. WTCHS 's dependence on trust introduces instability because small variations in trust values can trigger re-clustering, even when the physical distance between vehicles remains unchanged. This results in a higher number of cluster changes, undermining the network's overall efficiency.

Similarly, the ECBLTR method, while considering factors such as node degree and distance from the base station, relies heavily on fuzzy logic, which does not respond quickly enough to rapid changes in vehicle topology. The inherent delay in recalculating membership functions in fuzzy systems causes frequent cluster reshuffling, especially when vehicle density fluctuates significantly. In contrast, the IMICLiVAN method's dynamic clustering approach adapts more fluidly to real-time vehicular movement, reducing unnecessary cluster changes by ensuring that clusters remain stable even when vehicles move at varying speeds.

While EKSGA uses a genetic algorithm to form clusters, its iterative process can lead to frequent re-clusterings as new fittest nodes are continuously selected to serve as cluster heads. This frequent recalibration of clusters introduces instability, especially when the genetic algorithm identifies new high-fitness vehicles that were not initially considered. The IMICLiVAN method avoids this issue by using multiple selection criteria (e.g., local distance, relative speed, node degree) to form stable clusters that do not require constant recalibration. As a result, the number of cluster changes is significantly lower compared to EKSGA, while also outperforming WTCHS and ECBLTR in maintaining cluster stability.

The IMICLiVAN method shows a 33% improvement in reducing cluster changes compared to ECBLTR, an 82.97% improvement over EKSGA, and a 72% reduction compared to WTCHS, indicating its superior ability to maintain stable clusters with lower overhead.

In conclusion, as illustrated in Figure 4-4, the IMICLiVAN clustering method offers significant advantages over existing approaches by holistically addressing the key challenges of VANETs in a balanced way.

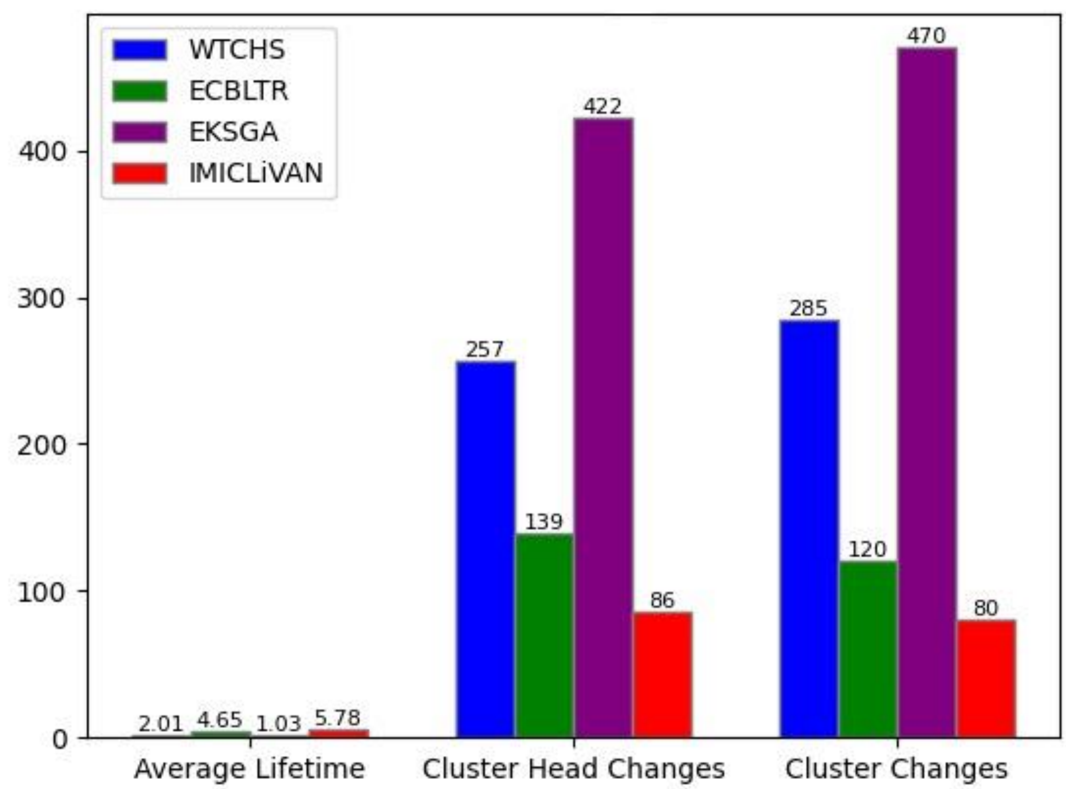

Figure 4-4 Comparison of Simulation Results

Integrating multiple metrics—such as Local Distance, Relative Speed, Trust, and Base Station Distance, the IMICLiVAN method ensures a longer network lifetime, fewer cluster changes, and more stable cluster head selections. This results in a more efficient, reliable, and adaptable VANET system compared to the WTCHS, ECBLTR, and EKSGA algorithms, which tend to over-rely on specific metrics and face challenges in dynamic environments. Through its dynamic clustering architecture and balanced decision-making process, the IMICLiVAN method outperforms these traditional approaches in maintaining network stability and extending network operational time.

4.8 Overview of the IMICLiVAN Hybrid Clustering Approach

Our IMICLiVAN hybrid approach, which combines the K-Means algorithm and Silhouette Score for optimal cluster formation, followed by a weighted formula for cluster head selection, significantly improved network performance in all three metrics. To obtain an overview of the method's performance compared to other state-of-the-art methods, we have included the individual performance of each method over the three metrics in Table 4-1. Table 4-1 results indicate that the IMICLiVAN method improves network stability, reducing the frequency of cluster and cluster head changes. The IMICLiVAN method achieved a network lifetime of 5.78 rounds in the simulation, with 86 cluster head changes and 80 cluster changes. This performance demonstrates the effectiveness of the K-Means clustering algorithm dynamically adjusting to changes in node positions while maintaining stable clusters over time. Thus, it can be observed that our proposed methods improve performance over all three of the target metrics.

Table 4-1Comparison of IMICLiVAN with WTCHS (Trust), ECBLTR (Fuzzy), and EKSGA (Genetic Algorithm)

Metrics	WTCHS (Trust)	ECBLTR (Fuzzy)	EKSGA (Genetic Algorithm)	IMICLiVAN
Network Lifetime (rounds)	2.01	4.65	1.03	5.78
Cluster Head Changes	257	139	422	86
Cluster Changes	285	120	470	80

CHAPTER 5 CONCLUSIONS AND FUTURE WORKS

Vehicular Ad Hoc Networks (VANETs) are essential in advancing road safety, optimizing traffic systems, and improving vehicle-to-vehicle and vehicle-to-infrastructure communication. However, due to their high mobility and dynamic environments, managing clustering and maintaining network stability remain significant challenges. Existing clustering protocols commonly face frequent re-clustering and short-lived cluster heads, limiting their practical deployment.

This research introduced the IMICLiVAN method, which addresses these challenges by utilizing a K-Means clustering algorithm combined with the Silhouette Score to optimize cluster formation. In addition, IMICLiVAN applies a weighted selection process for cluster heads, taking into account seven key features: Relative Speed, Answer Ratio, Trust, Base Station Distance, Node Degree, Node Center, and Local Distance. This multi-factor approach results in more stable clusters, reducing the frequency of re-clustering events and prolonging network lifetime.

Simulation results show that IMICLiVAN outperforms existing methods such as WTCHS, ECBLTR, and EKSGA across several critical metrics. IMICLiVAN achieves an average network lifetime of 5.78 rounds, significantly higher than ECBLTR (4.65 rounds), WTCHS (2.01 rounds), and EKSGA (1.03 rounds). Additionally, IMICLiVAN demonstrates superior performance in terms of cluster stability, with only 86 cluster head changes compared to 257 for WTCHS, 139 for ECBLTR, and 422 for EKSGA. Furthermore, IMICLiVAN results in the fewest cluster changes, with just 80, while WTCHS, ECBLTR, and EKSGA experience 285, 120, and 470 changes, respectively.

These results underscore IMICLiVAN's ability to enhance network stability and efficiency in highly dynamic VANET environments. With fewer re-clustering events and more prolonged cluster stability, IMICLiVAN is a promising solution for the next generation of intelligent transportation systems, ensuring reliable communication while minimizing overhead.

In future work, we plan to explore integrating enhanced security mechanisms into the proposed clustering protocol. While the current focus is on optimizing clustering efficiency, incorporating robust security features could further strengthen the resilience of VANETs against malicious attacks. One potential direction is the development of trust-based protocols that utilize cryptographic methods or blockchain technology to secure the selection of cluster heads and protect communication within clusters. This would ensure that only trusted and verified vehicles participate in the network, thereby improving the stability and security of VANETs in dynamic environments. By addressing these security challenges, the protocol could better handle scenarios involving compromised or uncooperative vehicles, ultimately contributing to safer and more reliable vehicular communication.

REFERENCES

[1] M. F. Khan, F. Aadil, M. Maqsood, S. H. R. Bukhari, M. Hussain, and Y. Nam, “Moth flame clustering algorithm for internet of vehicle (MFCA-IoV),” *IEEE Access*, vol. 7, pp. 11613–11629, 2019, doi: 10.1109/ACCESS.2018.2886420.

[2] A. Alsarhan, Y. Kilani, A. Al-Dubai, A. Y. Zomaya, and A. Hussain, “Novel Fuzzy and Game Theory Based Clustering and Decision Making for VANETs,” *IEEE Trans Veh Technol*, vol. 69, no. 2, pp. 1568–1581, Feb. 2020, doi: 10.1109/TVT.2019.2956228.

[3] M. J. N. Mahi *et al.*, “A Review on VANET Research: Perspective of Recent Emerging Technologies,” *IEEE Access*, vol. 10, pp. 65760–65783, 2022, doi: 10.1109/ACCESS.2022.3183605.

[4] K. Kandali, L. Bennis, O. El Bannay, and H. Bennis, “An Intelligent Machine Learning Based Routing Scheme for VANET,” *IEEE Access*, vol. 10, pp. 74318–74333, 2022, doi: 10.1109/ACCESS.2022.3190964.

[5] M. H. Badole and A. D. Thakare, “Cluster-based multicast optimized routing in VANETs using elite knowledge-based genetic algorithm,” *Knowl Based Syst*, vol. 294, p. 111773, Jun. 2024, doi: 10.1016/J.KNOSYS.2024.111773.

[6] R. R. Sahoo, R. Panda, D. K. Behera, and M. K. Naskar, “A trust based clustering with Ant Colony Routing in VANET,” *2012 3rd International Conference on Computing, Communication and Networking Technologies, ICCCNT 2012*, 2012, doi: 10.1109/ICCCNT.2012.6395939.

[7] M. A. Al-Shareeda and S. Manickam, “A Systematic Literature Review on Security of Vehicular Ad-Hoc Network (VANET) Based on VEINS Framework,” *IEEE Access*, vol. 11, pp. 46218–46228, 2023, doi: 10.1109/ACCESS.2023.3274774.

[8] S. R. Bin Ramlee, S. Hasan, and S. K. Subramaniam, “Cluster Optimization in VANET using MFO Algorithm and K-Means Clustering,” *2023 13th International Conference on Information Technology in Asia, CITA 2023*, pp. 70–75, 2023, doi: 10.1109/CITA58204.2023.10262579.

[9] G. Nithyanandam, C. Ambiyaram, and S. Prabathkumar, "An intelligent hybrid prairie dog optimization algorithm-based stable cluster reliable routing scheme for VANETs," *International Journal of Communication Systems*, vol. 36, no. 14, p. e5549, Sep. 2023, doi: 10.1002/DAC.5549.

[10] H. Amari, Z. A. El Houda, L. Khoukhi, and L. H. Belguith, "Trust Management in Vehicular Ad-Hoc Networks: Extensive Survey," *IEEE Access*, vol. 11, pp. 47659–47680, 2023, doi: 10.1109/ACCESS.2023.3268991.

[11] A. Naeem *et al.*, "Enhanced clustering based routing protocol in vehicular ad-hoc networks," *Wiley Online LibraryA Naeem, M Rizwan, S Alsubai, A Almadhor, M Akhtaruzzaman, S Islam, H RahmanIET Electrical Systems in Transportation, 2023•Wiley Online Library*, vol. 13, no. 1, Mar. 2023, doi: 10.1049/els2.12069.

[12] M. Jabbar, H. T.-2022 15th I. C. on, and undefined 2022, "A betweenness centrality based clustering in VANETs," *ieeexplore.ieee.orgMK Jabbar, H Trabelsi2022 15th International Conference on Security of Information and, 2022•ieeexplore.ieee.org*, 2022, doi: 10.1109/SIN56466.2022.9970553.

[13] A. Amirshahi, M. Romoozi, M. A. Raayatpanah, and S. A. Asghari, "Modeling Geographical Anycasting Routing in Vehicular Networks," *KSII Transactions on Internet and Information Systems (TIIS)*, vol. 14, no. 4, pp. 1624–1647, Apr. 2020, doi: 10.3837/TIIS.2020.04.012.

[14] M. Badole, A. Thakare, and D. Oliva, "An Improved Multi-objective Optimization Framework with a Hybrid Model for Vehicular Adhoc Network Routing Services," *Wirel Pers Commun*, pp. 1–28, Aug. 2024, doi: 10.1007/S11277-024-11245-6/TABLES/8.

[15] A. Bijalwan, I. Hussain, K. C. Purohit, and M. A. Kumar, "Enhanced Ant Colony Optimization for Vehicular Ad Hoc Networks Using Fittest Node Clustering," *Sustainability 2023, Vol. 15, Page 15903*, vol. 15, no. 22, p. 15903, Nov. 2023, doi: 10.3390/SU152215903.

[16] A. Jindal *et al.*, "RoVAN: A rough set-based scheme for cluster head selection in vehicular ad-hoc networks," *ieeexplore.ieee.orgA Dua, P Sharma, S Ganju, A Jindal, GS Aujla, N*

Kumar, JJPC Rodrigues2018 IEEE Global communications conference (GLOBECOM), 2018•ieeexplore.ieee.org, 2018, doi: 10.1109/GLOCOM.2018.8647576.

[17] S. Vergis, … G. T.-2021 6th S.-E., and undefined 2021, "A proposed multi-head clustering algorithm for VANET environments," *ieeexplore.ieee.orgS Vergis, G Tsoumanis, K Oikonomou2021 6th South-East Europe Design Automation, Computer Engineering, 2021•ieeexplore.ieee.org*, doi: 10.1109/SEEDA-CECNSM53056.2021.9566263.

[18] F. Mirsadeghi, M. K. Rafsanjani, and B. B. Gupta, "A trust infrastructure based authentication method for clustered vehicular ad hoc networks," *Peer Peer Netw Appl*, vol. 14, no. 4, pp. 2537–2553, Jul. 2021, doi: 10.1007/S12083-020-01010-4.

[19] M. Hadded, R. Zagrouba, A. Laouiti, P. Muhlethaler, and L. A. Saidane, "A multi-objective genetic algorithm-based Adaptive Weighted Clustering Protocol in VANET," *2015 IEEE Congress on Evolutionary Computation, CEC 2015 - Proceedings*, pp. 994–1002, Sep. 2015, doi: 10.1109/CEC.2015.7256998.

[20] G. H. Alsuhli, A. Khattab, and Y. A. Fahmy, "A Mobility-Based Double-Head Clustering Algorithm for Dynamic VANET," *2018 Proceedings of the Japan-Africa Conference on Electronics, Communications, and Computations, JAC-ECC 2018*, pp. 91–94, Jul. 2018, doi: 10.1109/JEC-ECC.2018.8679569.

[21] A. Amirshahi, M. Romoozi, M. A. Raayatpanah, and S. A. Asghari, "Anycast routing in time-expanded vehicular networks," *Computers & Electrical Engineering*, vol. 82, p. 106560, Mar. 2020, doi: 10.1016/J.COMPELECENG.2020.106560.

[22] G. Khayat, … C. M.-2020 I., and undefined 2020, "VANET clustering based on weighted trusted cluster head selection," *ieeexplore.ieee.org*, Accessed: Jul. 31, 2024. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9148339/

[23] I. Alam *et al.*, "Efficient and Secure Graph-Based Trust-Enabled Routing in Vehicular Ad-Hoc Networks," *Mobile Networks and Applications*, pp. 1–21, Jan. 2024, doi: 10.1007/S11036-023-02274-9/FIGURES/10.

[24] S. B.-V. Communications and undefined 2024, "Maximizing VANET performance in cluster head selection using Intelligent Fuzzy Bald Eagle optimization," *Elsevier*, Accessed:

Aug. 22, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2214209623000906

[25] M. Aissa, B. Bouhdid, and A. Ben Mnaouer, “Enhanced Fuzzy logic-based Cluster Stability in Vehicular ad hoc Network,” *2021 International Symposium on Networks, Computers and Communications, ISNCC 2021*, 2021, doi: 10.1109/ISNCC52172.2021.9615872.

[26] G. D. Singh, M. Prateek, S. Kumar, M. Verma, D. Singh, and H. N. Lee, “Hybrid Genetic Firefly Algorithm-Based Routing Protocol for VANETs,” *IEEE Access*, vol. 10, pp. 9142–9151, 2022, doi: 10.1109/ACCESS.2022.3142811.

[27] C. B. Ruban and B. Paramasivan, “VTD Protocol: A Cluster based Trust Model for Secure Communication in VANET,” *International Journal of Intelligent Engineering and Systems*, vol. 13, no. 1, 2020, doi: 10.22266/ijies2020.0229.04.

[28] A. F. E. Abadi, S. E. Asghari, S. Sharifani, S. A. Asghari, and M. B. Marvasti, “A Survey on Utilizing Reinforcement Learning in Wireless Sensor Networks Routing Protocols,” *Proceedings of the 13th International Conference on Information and Knowledge Technology, IKT 2022*, 2022, doi: 10.1109/IKT57960.2022.10039013.

[29] A. F. E. Abadi, S. A. Asghari, M. B. Marvasti, G. Abaei, M. Nabavi, and Y. Savaria, “RLBEEP: Reinforcement-Learning-Based Energy Efficient Control and Routing Protocol for Wireless Sensor Networks,” *IEEE Access*, vol. 10, pp. 44123–44135, 2022, doi: 10.1109/ACCESS.2022.3167058.

[30] K. A. Awan, I. Ud Din, A. Almogren, M. Guizani, and S. Khan, “StabTrust-A Stable and Centralized Trust-Based Clustering Mechanism for IoT Enabled Vehicular Ad-Hoc Networks,” *IEEE Access*, vol. 8, pp. 21159–21177, 2020, doi: 10.1109/ACCESS.2020.2968948.

[31] M. Gayathri and C. Gomathy, “Fuzzy based Trusted Communication in Vehicular Ad hoc Network,” *2022 2nd International Conference on Intelligent Technologies, CONIT 2022*, 2022, doi: 10.1109/CONIT55038.2022.9847823.

[32] G. H. Alsuhli, A. Khattab, and Y. A. Fahmy, “An evolutionary approach for optimized VANET clustering,” *Proceedings of the International Conference on Microelectronics, ICM*, vol. 2019-December, pp. 70–73, Dec. 2019, doi: 10.1109/ICM48031.2019.9021941.

[33] V. M. V. S. Aditya, S. Dhuli, P. L. Sashrika, K. K. Shivani, and T. Jayanth, "Closeness Centrality Based Cluster Head Selection Algorithm for Large Scale WSNs," *Proceedings - 2020 12th International Conference on Computational Intelligence and Communication Networks, CICN 2020*, pp. 107–111, Sep. 2020, doi: 10.1109/CICN49253.2020.9242639.

[34] D. Choudhary, D. Yadav, J. Joshi, M. Deka, and S. Jha, "A Model Based Connectivity and Localization Strategy for Vehicular Ad Hoc Networks," *Proceedings - International Conference on Intelligent Systems, Modelling and Simulation, ISMS*, vol. 0, pp. 331–336, Jul. 2016, doi: 10.1109/ISMS.2016.17.

[35] R. Hajlaoui, E. Alsolami, T. Moulahi, and H. Guyennet, "Construction of a stable vehicular ad hoc network based on hybrid genetic algorithm," *Telecommun Syst*, vol. 71, no. 3, pp. 433–445, Jul. 2019, doi: 10.1007/S11235-018-0513-6/FIGURES/15.

[36] Y. A. Shah, H. A. Habib, F. Aadil, M. F. Khan, M. Maqsood, and T. Nawaz, "CAMONET: Moth-Flame Optimization (MFO) Based Clustering Algorithm for VANETs," *IEEE Access*, vol. 6, pp. 48611–48624, Sep. 2018, doi: 10.1109/ACCESS.2018.2868118.

[37] S. Shivshankar and A. Jamalipour, "An evolutionary game theory-based approach to cooperation in VANETs under different network conditions," *IEEE Trans Veh Technol*, vol. 64, no. 5, pp. 2015–2022, May 2015, doi: 10.1109/TVT.2014.2334655.

[38] V. Krishna Meera and C. Balasubramanian, "A hybrid fennec fox and sand cat optimization algorithm for clustering scheme in VANETs," *Sustainable Computing: Informatics and Systems*, vol. 42, p. 100983, Apr. 2024, doi: 10.1016/J.SUSCOM.2024.100983.

[39] M. Ramalingam and R. Thangarajan, "Mutated k-means algorithm for dynamic clustering to perform effective and intelligent broadcasting in medical surveillance using selective reliable broadcast protocol in VANET," *Comput Commun*, vol. 150, pp. 563–568, Jan. 2020, doi: 10.1016/J.COMCOM.2019.11.023.

[40] A. Naghizadeh and D. N. Metaxas, "Condensed Silhouette: An Optimized Filtering Process for Cluster Selection in K-Means," *Procedia Comput Sci*, vol. 176, pp. 205–214, Jan. 2020, doi: 10.1016/J.PROCS.2020.08.022.