The Journal of Supercomputing (2026) 82:319
https://doi.org/10.1007/s11227-026-08358-z

IMICLiVAN: an improved method to increase cluster lifetime in vehicular ad hoc networks (VANETs)

Erfan Moghadam[1] · **Elham Asghari**[2] · **Seyed Amir Asghari**[3] · **Mohammadreza Binesh Marvasti**[1] · **Yvon Savaria**[2]

Received: 26 April 2025 / Accepted: 13 February 2026

Abstract

Vehicular ad hoc networks (VANETs) represent a transformative technology that enhances road safety, optimizes traffic flow, and enables efficient communication between vehicles and infrastructure. However, vehicular environments' dynamic and unpredictable nature presents significant challenges, including frequent cluster head (CH) changes, increased communication overhead, and limited network lifetime. These issues degrade overall network performance and hinder the practical deployment of VANETs in real-world scenarios. This paper introduces IMICLiVAN, an enhanced clustering mechanism designed to address these challenges. The proposed method integrates the K-means algorithm with the silhouette score to dynamically determine the optimal number of clusters, ensuring compact and stable cluster structures. Additionally, a weighted formula for CH selection is employed, designed to balance multiple metrics to improve cluster stability and reduce unnecessary cluster head changes, and minimize reformation overhead. The performance of IMICLiVAN was evaluated through simulation-based experiments under varying traffic densities. Simulation results demonstrate that IMICLiVAN significantly outperforms existing clustering methods across key metrics, including WTCHS, ECBLTR, and EKSGA. These findings establish IMICLiVAN as a practical and effective solution for enhancing VANET performance in dynamic environments. From a supercomputing perspective, the per-step clustering quality evaluation (silhouette-based selection over multiple candidate-K values) and distance-driven computations become increasingly demanding in dense or city-scale VANETs; therefore, IMICLiVAN is designed to be amenable to parallel and distributed execution (e.g., RSU/MEC or cloud/HPC backends) to help meet strict latency constraints in real-time operation.

Keywords Clustering · Communication overhead · Mobility · Network lifetime · Routing · Vehicular ad hoc network

---

Extended author information available on the last page of the article

1 Introduction

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 [2–4].

The fundamental feature of VANETs is their dynamic topology, characterized by frequent changes in network connectivity due to the fast movement of vehicles. Maintaining a stable network is a critical challenge [5, 6]. VANETs are used in various 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].

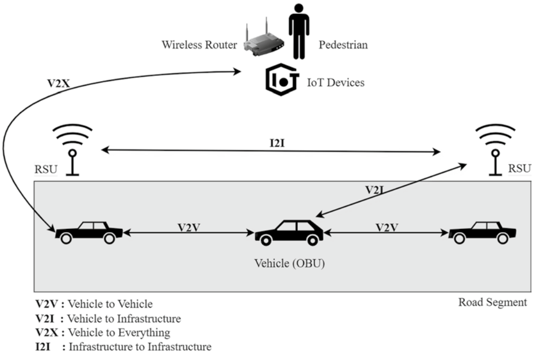
VANET communication relies on three critical hardware components: onboard units (OBUs), roadside units (RSUs), and base stations (BS). OBUs, installed in vehicles, enable vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communication by transmitting real-time data, including speed, position, and direction, to nearby vehicles and RSUs. RSUs are stationary devices placed at strategic locations, such as intersections or high-traffic areas, to facilitate V2I and infrastructure-to-infrastructure (I2I) communication, relay traffic information, and act as temporary data storage gateways [9, 10]. Base stations complement RSUs by providing broader connectivity, linking RSUs to central servers or cloud systems for large-scale data processing and citywide traffic management. Together, these components form a robust vehicular communication network [11].

Relevance to Supercomputing and Real-Time Processing. Many ITS/VANET services require time-sensitive (near real-time) clustering updates as topology changes rapidly. While IMICLiVAN is designed to remain lightweight for onboard feasibility in moderate-density scenarios, dense urban and city-scale deployments can significantly increase per-step computational pressure due to repeated clustering quality evaluation and multimetric cluster head scoring. Consequently, meeting strict latency at scale can motivates the use of parallel and distributed processing, where computations can be offloaded and executed across RSU/MEC (edge) nodes and/or aggregated through cloud/HPC infrastructures, aligning the proposed work with the supercomputing scope.

1.1 Types of communication in VANETs

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

Fig. 1 Infrastructure of VANET



Vehicle-to-Vehicle (V2V) Communication: V2V communication allows vehicles to communicate with each other directly over a short range. This type of communication is essential for safety applications such as collision avoidance, lane-change warnings, and cooperative driving [12, 13].

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].

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].

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].

The importance of clustering in VANETs lies in its ability to improve scalability and reduce communication overhead, especially in dense traffic scenarios. By structuring communication hierarchically, clustering ensures efficient resource utilization and minimizes network congestion. Furthermore, clustering facilitates the management of network dynamics, reducing the frequency of communication disruptions caused by rapid topology changes.

Clustering supports scalable and reliable communication in VANETs by enabling low-latency dissemination for safety-critical messages (e.g., collision warnings) and efficient sharing of real-time traffic information with low overhead. It also benefits non-safety services such as infotainment and cooperative navigation by improving communication reliability. Overall, clustering is a key enabler for practical ITS deployments under dynamic vehicular conditions.

Despite these benefits, VANET clustering is challenging because rapid topology changes and density fluctuations can trigger frequent re-clustering, increasing overhead and reducing stability. In dense urban settings, congestion, delays, and packet loss further degrade performance, while CH selection becomes difficult as static or poorly chosen criteria shorten cluster lifetime. These challenges motivate adaptive and computationally efficient clustering strategies.

1.2 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 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 high mobility and rapid topology changes, neighborhood relationships vary quickly, which can trigger repeated re-clustering and increase control overhead [25]. In practice, an additional work-specific challenge is adapting the cluster structure to traffic density variations: using a fixed or manually tuned number of clusters may lead to fragmented clusters in sparse conditions or overloaded clusters in dense conditions. This motivates online mechanisms that can adjust clustering decisions according to the observed topology and density dynamics.

Cluster Head Longevity: Ensuring CH longevity is challenging because a CH selected based on a single static metric may become unsuitable as relative mobility, connectivity, and neighborhood composition change[9]. Moreover, when interaction-based indicators are considered (e.g., trust- and response-related measures), their estimates may be affected by communication uncertainty such as packet loss and variable delay. Therefore, CH selection benefits from integrating multiple complementary metrics to improve decision robustness and reduce unnecessary CH re-elections.

Communication Latency: Low latency is critical for many ITS services, and excessive clustering overhead can increase delay and reduce responsiveness[26]. Beyond transmission delay, a practical requirement is that clustering and CH decisions should be computed within each decision interval; under dense/city-scale conditions, repeated clustering quality evaluation and per-vehicle CH scoring can become time-critical, motivating efficient and scalable implementations for real-time operation.

1.3 Problem statement and motivation

Problem statement: In vehicular ad hoc networks (VANETs), vehicles must be organized into clusters to support scalable and reliable communication. However, high mobility and rapidly changing traffic density frequently trigger re-clustering and cluster head (CH) switching, which increases control overhead and degrades stability. Therefore, within each time window, given the current set of vehicles and their observable mobility and connectivity conditions, the goal is to (i) determine an appropriate number of clusters, (ii) assign vehicles to clusters, and (iii) select one CH per cluster, such that cluster stability and CH lifetime are maximized while keeping communication and management overhead low and preserving reliable intra-cluster and inter-cluster communication.

Motivation: Many existing VANET clustering schemes assume a fixed or manually tuned number of clusters and rely on a single dominant criterion for cluster head selection. In practice, traffic density and topology vary rapidly, and a fixed K can lead to fragmented clusters in sparse scenarios or overloaded, unstable clusters in dense scenarios. Moreover, single-metric CH selection can become brittle when relative mobility and neighborhood composition change, increasing CH switches and control overhead. These gaps motivate an online mechanism that (1) adapts the cluster count to the observed density, and (2) selects CHs using a robust multimetric scoring strategy to reduce unnecessary CH changes while keeping the process computationally feasible for time-sensitive operation.

Contributions: The main contributions of this work are:

- An online clustering mechanism with specific factors that combines K-means with silhouette-based selection to adaptively determine K_t.
- A weighted multimetric cluster head selection rule that jointly considers mobility, connectivity, communication, and trust indicators to improve CH longevity.
- A simulation-based evaluation under varying traffic densities, demonstrating consistent gains over representative baselines (WTCHS, ECBLTR, and EKSGA) on stability and lifetime-related metrics.

1.4 Clustering in VANETs

This paper aims to address the challenges of clustering and cluster head selection (CHS) in VANETs, focusing on enhancing the stability and efficiency of the network in highly dynamic vehicular environments. The proposed method comprises two main phases: dynamic clustering and optimized 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 CH for each cluster. The CH is responsible for managing communication within the cluster and relaying messages to other clusters or 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 that appears to be most capable of maintaining cluster stability and ensuring efficient communication.

This paper is organized as follows: Sect. 2 surveys existing clustering and CH selection algorithms, highlighting their strengths and weaknesses concerning VANETs. Section 3 describes the methodology, approaches, and proposed method and details the proposed clustering algorithm, including CH selection criteria and simulation setups using the SUMO and TraCI libraries in Python. Section 4 presents and analyzes the results of the simulations, comparing the proposed approach with the best-related methods. Finally, Sect. 5 summarizes the research's key findings and outlines potential enhancements for VANET clustering and communication.

2 Literature review

2.1 Related works

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

2.1.1 Trust-based clustering VANET approaches

Trust metrics have been widely used to improve cluster stability and enhance security in VANETs. Khayat et al. [22] combined trust, velocity, and distance in a weighted formula for CH selection, resulting in improved stability and reduced delays. Mirsadeghi et al. [18] extended this approach by incorporating trustworthiness from RSUs, enhancing reliability but creating dependence on infrastructure. Although scalability issues persist in high-mobility environments, Awan et al. [30]

introduced the StabTrust mechanism, using RSU-collected trust components and backup CHs for stability. Gayathri et al. [31] used fuzzy logic to detect malicious nodes, improving security but facing computational overhead in dense traffic. Sahoo et al. [6] integrated a trust mechanism with Ant Colony Optimization (ACO) for routing, while Alam et al. [23] developed a graph-based trust protocol prioritizing high-trust paths. Ruban et al. [27] employed K-means clustering with trust metrics to improve packet delivery but faced delays in updating trust values in dense scenarios.

More recently, Saleem et al. [32] employed a deep learning-based dynamic cluster head selection scheme, where mobility and link-related features are fed to a neural model to predict stable CHs and reduce re-clustering, at the cost of additional training and inference complexity. More recently, Babu and Rao proposed a secure routing protocol based on trust-based clustering combined with a bionic intelligence algorithm for UAV-assisted VANETs [33]. Their approach integrates trust metrics into the cluster head selection and routing process to improve secure data transmission and network reliability in dynamic vehicular environments.

2.1.2 Fuzzy logic-based clustering

Fuzzy logic has been utilized to optimize CH selection using metrics like distance, velocity, and link stability. Naeem et al. [11] employed Sugeno fuzzy systems to extend network lifetime by 10%, while Aissa et al. [25] introduced a FitFactor-based approach to improve cluster lifetimes, reducing overhead but adding computational complexity. Brindha et al. [24] integrated fuzzy logic with optimization techniques to reduce energy consumption and delays, but encountered challenges in larger networks. More recently,

Malakreddy et al. proposed FLQL-VANET, a hybrid fuzzy logic and Q-learning-based framework that leverages fuzzy inference for decision-making while using reinforcement learning to adapt routing and clustering behavior in dynamic VANET environments [34]. In addition, Balaji et al. introduced a fuzzy-based secure clustering and routing technique that employs enhanced fuzzy reasoning to improve reliability and security in vehicular networks [35].

2.1.3 Clustering-based on a rough set scheme

Dua et al. [16] proposed the RoVAN method, using rough set theory (RST) to minimize CH selection time and enhance stability. Although effective in high-density scenarios, RST's complexity poses implementation challenges.

More recently, Abdulrazzak et al. proposed a hybrid clustering models combining covering rough set with K-means have been proposed to enhance cluster stability and reliability in dynamic vehicular scenarios, indicating ongoing interest in rough set-based clustering frameworks for VANETs and related networks [36].

To facilitate comparison across the above families, we summarize their key trade-offs as follows:

Comparative analysis (trust-based vs. fuzzy/rough set): The reviewed trust-based studies emphasize stability and security improvements, but they also report practical constraints such as infrastructure dependence in RSU-assisted trust designs and increased maintenance overhead in dense traffic [18, 27, 31–33]. By contrast, fuzzy logic approaches focus on multimetric CH decision fusion and report lifetime/overhead benefits, while additional computational complexity and scalability challenges are noted as network size grows [11, 24, 25, 34, 35]. Rough set-based clustering targets faster CH selection and stability, yet implementation complexity remains a key concern [16, 36]. Overall, these families illustrate a recurring trade-off reported across the reviewed studies: Mechanisms that enrich CH decision-making (e.g., trust inference or multimetric fusion) are reported to improve stability/lifetime, while overhead and implementation/scalability constraints are also reported, particularly under dense and dynamic VANET conditions [16, 18, 25, 31].

2.1.4 Multihead clustering algorithm

Multihead clustering approaches use secondary CHs to improve stability. Alsuhli et al. [37] proposed a double-head clustering model (DHC), which reduces re-clustering but increases processing overhead. They optimized DHC using NSGA-III metaheuristics, improving cluster stability but limiting real-time applicability. Vergis et al. [17] allowed vehicles to join multiple clusters, extending cluster lifetimes but facing scalability issues.

More recently, Khayat et al. proposed a VANET clustering scheme based on double cluster head selection to improve robustness when a cluster head fails to deliver packets under damaged infrastructure conditions [38]. Siddiqa et al. introduced a multihead nomination clustering mechanism in vehicular networks, where multiple head vehicles are maintained to mitigate frequent disconnections caused by high mobility and to reduce broadcast overhead [39].

2.1.5 Betweenness centrality-based clustering

Centrality-based methods prioritize CHs with strategic positions in the network. Jabbar et al. [12] introduced betweenness centrality clustering (BCBC) to enhance cluster stability, while Aditya et al. [40] used closeness centrality to balance energy consumption. Choudhary et al. [41] combined centrality measures with link prediction for improved location accuracy but faced computational challenges.

Jabbar et al. [42] proposed a hypergraph-based clustering model for urban VANET scenarios, in which betweenness centrality is employed to construct an evolving traffic graph and identify structurally important vertices based on shortest-path characteristics. In addition, Moura et al. [43] introduced a centrality-based approach to select vehicles as data aggregation points in vehicular sensor networks, where vehicles with central positions are chosen to collect and aggregate data from neighboring vehicles before offloading it to processing stations.

2.1.6 Genetic algorithm-based clustering

Genetic algorithms (GAs) have been widely applied for multiobjective optimization. Hadded et al. [19] used NSGA-II to improve cluster stability and reduce overhead, while Singh et al. [26] combined GAs with firefly algorithms for routing optimization. Hajlaoui et al. [44] enhanced K-medoids clustering with hybrid GAs and Tabu Search for stable clustering. Badole et al. [5] introduced EKSGA, achieving improved throughput but facing computational overhead.

Furthermore, Charoenchai et al. [45] proposed a genetic algorithm-based multihop VANET clustering framework that integrates coalitional game theory to improve cluster stability and communication reliability under high mobility. However, the combination of evolutionary optimization and game-theoretic mechanisms may increase computational complexity and limit real-time applicability in large-scale scenarios. More recently, Madasamy et al. [46] introduced a location-aware GA-based clustering approach aimed at enhancing load balancing and communication efficiency in VANETs, reporting improved packet delivery and delay performance at the cost of additional computational overhead and scalability concerns in dense vehicular environments.

Comparative analysis (multihead vs. centrality vs. GA/metaheuristics): Multihead clustering is reported to reduce re-clustering by introducing secondary CH roles, while increased processing overhead is also reported as a practical cost [20]. When multihead designs are optimized via metaheuristics, improved stability is reported, but higher processing requirements have been noted as a factor that can limit real-time applicability [37]. Centrality-based approaches prioritize strategically positioned CHs and report improvements in stability or energy balancing, whereas computational challenges are reported, particularly when combined with link prediction components [12, 40–43]. GA/metaheuristic methods are reported to improve stability/throughput or support multiobjective optimization, while computational overhead is repeatedly highlighted as a limitation in practice [5, 19, 26, 44–46]. Overall, across these families, the reviewed studies report a trade-off between stability-oriented mechanisms and increased computational/processing overhead that can constrain real-time applicability.

2.1.7 Hybrid dog and beluga whale optimization algorithm

Nithyanandam et al. [9] proposed a hybrid prairie dog and beluga whale optimization algorithm (HPDBWOA-NC), combining prairie dog optimization and beluga whale optimization to balance exploration and exploitation during cluster formation. This hybrid algorithm optimizes CH selection, reducing energy consumption and mean delay in dynamic VANET environments. However, it is highly sensitive to parameter configuration, which may limit its adaptability in real-world scenarios. Begum et al. proposed a multiobjective prairie dog optimization-based cluster routing scheme for VANETs, where visible light communication is integrated to enhance energy efficiency while reducing delay [47]. By balancing exploration and exploitation during optimization, their approach improves packet delivery ratio, delay, and throughput compared to existing traffic-aware routing schemes, demonstrating the effectiveness of prairie dog-inspired multiobjective optimization for vehicular clustering and routing. Hijjawi et al. proposed a hybrid prairie dog and Harris hawks optimization algorithm (HPDO) to address the resource allocation problem in wireless networks, aiming to enhance exploitation capability and convergence behavior in NP-hard optimization scenarios [48]. Their results demonstrate superior optimization performance and faster convergence compared to standalone prairie dog and Harris hawks algorithms, highlighting the effectiveness of prairie dog-based hybridization for complex wireless network optimization tasks. Husnain et al. proposed WOACNET, an intelligent clustering framework based on the whale optimization algorithm for cluster head selection in VANETs, where transmission range, node density, mobility characteristics, and grid-based spatial factors are jointly considered [49]. Their results show notable improvements in cluster stability and lifetime compared to other metaheuristic approaches, such as gray wolf and ant lion optimization, demonstrating the effectiveness of whale-inspired optimization for dynamic vehicular environments. Venkatasubramanian et al. proposed an improved Beluga Whale Optimizer-based cluster head selection scheme for FANETs, aiming to address premature convergence and limited population diversity in the original BWO [50]. By incorporating opposition-based learning and an adaptive cluster maintenance mechanism, their approach achieves improved packet delivery ratio, extended network lifetime, and reduced energy consumption under high-mobility UAV scenarios.

2.1.8 Moth flame optimization (MFO) algorithm

Moth flame optimization (MFO) has been employed to enhance cluster formation and communication efficiency in VANETs. Ramlee et al. [8] combined the MFO algorithm with K-means clustering to optimize CH selection and improve network coverage. This method mimics moth behavior to find optimal solutions for clustering, but it is sensitive to node density and transmission range, which can lead to overlapping clusters. Shah et al. [51] refined the MFO algorithm for reduced routing costs and improved network stability, while Khan et al. [1] developed MFCA-IoV, an MFO-based clustering algorithm for the Internet of vehicles, minimizing the number of CHs and improving energy efficiency and communication reliability.

Tariq et al. [52] proposed the IMOC scheme, where an MFO-based clustering mechanism is used in drone-assisted VANETs to increase vehicular coverage while reducing the number of required cluster heads in highly dynamic topologies. In a large-scale IoT context, Sadrishojaei et al. [53] designed a clustered routing method that employs MFO for cluster head selection, showing that MFO-driven clustering can serve as an effective basis for routing decisions in dense deployments, while still incurring the typical metaheuristic tuning and computational costs.

2.1.9 Ant colony optimization using fittest node clustering

Ant colony optimization (ACO) techniques have been applied to improve clustering and routing in VANETs. Bijalwan et al. [15] introduced a heuristic clustering algorithm integrated with an enhanced ACO approach. Their method, dynamic-aware transmission range parallel Euclidean distance (DA-TRPED), dynamically adjusts transmission ranges based on real-time distance calculations. This approach improves the packet delivery ratio (PDR), reduces packet drop rates, and lowers end-to-end delays. However, its reliance on accurate parallel Euclidean distance estimations poses limitations in environments with rapidly changing traffic conditions or irregular node distributions.

Ramamoorthy and Thangavelu [54] introduced the IDBACOR protocol, an improved distance-based ACO routing algorithm for vehicular ad hoc networks that determines inter-vehicular distances and then triggers a modified ant colony optimization process to construct routes, with simulation results indicating better throughput, higher packet delivery ratio, and lower average communication cost, propagation delay, and routing overhead than conventional ACO, opposition-based ACO, and greedy ACO routing variants. Khan et al. [55] proposed a street-centric routing scheme (SCRS) for bus-based VANETs, in which a multipath routing component based on the probability of street consistency and path consistency is combined with an ACO-based clustering mechanism for relay-bus selection, and their results show that this design improves packet delivery ratio while reducing end-to-end delay, computational cost, and unnecessary beacon messages compared with earlier bus-based routing schemes. These ACO-based designs retain the iterative and parameter-sensitive nature of ant colony metaheuristics, which may introduce non-negligible computational and signaling overhead that challenges real-time applicability in very dense or large-scale VANET deployments.

2.1.10 Hybrid game theory-based clustering

Hybrid game theory-based methods have been explored to optimize clustering and resource allocation in VANETs. Alsarhan et al. [2] combined fuzzy logic and game theory to enhance CH selection by ranking candidates based on mobility metrics such as speed, signal strength, and stability. Game theory was used to manage spectrum sharing, creating an efficient and cost-effective clustering framework. Similarly, Shivshankar et al. [56] integrated evolutionary game theory (EGT) with the public goods game (PGG) model to promote cooperation among vehicles for packet forwarding. This model allows vehicles to evolve cooperative behavior naturally, improving packet delivery and adapting to dynamic traffic scenarios. Both approaches demonstrate scalability and adaptability but may face challenges in computational efficiency and real-time decision-making in dense networks.

2.1.11 Hybrid fennec fox and sand cat optimization algorithm

Meera et al. [57] introduced the hybrid fennec fox and sand cat optimization algorithm (HFFSCOA) for improving cluster stability and CH selection in VANETs. The algorithm combines the strengths of the fennec fox optimization algorithm (FFOA) and the sand cat optimization algorithm (SCOA) to enhance clustering performance. A novel fitness function based on grid size, node orientation, velocity, and communication range ensures the selection of the most suitable CHs. This approach reduces the number of clusters, optimizes routes, and improves energy efficiency and link stability. However, the combined computational requirements of FFOA and SCOA may limit the algorithm's applicability in real-time and large-scale VANET environments.

Jahangeer et al. [58] proposed a hybrid cat swarm optimization–TOPSIS algorithm (HCSTA) for seamless vertical handoff in urban VANETs, where a hybrid cat swarm optimizer is combined with a TOPSIS-based decision module to execute vertical handoffs across heterogeneous wireless technologies, and its effectiveness is demonstrated through mobility and network simulations that analyze handoff occurrences, technology utility, and throughput under different vehicle speeds, densities, and movement directions. However, HCSTA is evaluated only in simulated urban VANET scenarios with a specific mix of wireless technologies, and the reported results still show a noticeable degradation in packet delivery ratio at higher vehicle densities and speeds, indicating that its performance remains sensitive to heavy network load and mobility patterns.

2.1.12 Hybrid customized hunger's foraging honey badger with dynamic multiobjective non-sorted genetic algorithm (CHFHB-DMNSGA)

Badole et al. [14] proposed the hybrid customized hunger's foraging honey badger optimization combined with a dynamic multiobjective non-sorted genetic algorithm (CHFHB-DMNSGA) for cluster-based VANET routing. This model optimizes multiple objectives, including packet delivery ratio (PDR), end-to-end delay, throughput, and control overhead. The hybrid approach improves residual energy utilization, alive node counts, and convergence rates. CHFHB-DMNSGA outperforms existing methods such as NSGA-II and HBA in terms of cluster stability and network performance. However, its increased computational complexity can lead to higher processing times, limiting its suitability for real-time and large-scale applications.

Badole and Thakare [59] proposed an optimized VANET routing framework that leverages Digital Twin technology and a hybrid hunger's foraging behavior customized honey badger optimization (HFCHBO) model formed by combining the standard honey badger algorithm with hunger games search–to perform multiobjective cluster head selection based on mean routing load, packet delivery ratio, throughput, end-to-end delay, and control packet overhead, and to construct routing paths between vehicles and the base station in dynamic vehicular environments.

Comparative analysis (hybrid swarm/optimization vs. ACO/Game-theoretic): Hybrid swarm/metaheuristic designs are reported to improve performance indicators such as energy consumption, delay, stability, and multiobjective network metrics through iterative optimization [9, 14, 57, 58]. At the same time, the reviewed studies report practical constraints such as sensitivity to parameter configuration in hybrid optimization [9], sensitivity to node density and transmission range in MFO-based clustering [8], and increased computational complexity that can limit real-time and large-scale applicability in hybrid multiobjective designs [14, 57]. In comparison, ACO- and game-theoretic frameworks are reported to improve packet delivery/cooperation and resource sharing decisions, yet they may rely on accurate distance-related estimations or face computational efficiency constraints for real-time decision-making in dense scenarios [2, 15, 54–56]. Collectively, the reviewed works highlight that stronger optimization and richer decision frameworks can yield reported performance gains, while complexity/efficiency constraints remain recurring concerns under dynamic and dense VANET conditions [14, 15, 57, 59].

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

Proposed method	Limitations	Approach family	optimization basis	Key constraint type
Weighted trusted formula [22], TVR (trust based on vehicles and roadside units) [18], Stab-Trust mechanism [30], fuzzy-based trusted communication [31], a trust-based clustering with ant colony routing [6], graph-based trust-enabled routing [6], 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	Trust-based/trust-assisted	Trust inference and trust-weighted selection may include RSU-assisted trust and routing integration	Range sensitivity; infrastructure availability/scalability limits
ugeno model fuzzy inference system [11], enhanced fuzzy logic-based clustering scheme [25], fuzzy logic with the bald eagle Search [24]	Its difficulty quickly adapting to frequent topology changes in dynamic VANET environments leads to potentially suboptimal cluster head selection and high computational complexity	Fuzzy logic-based	Fuzzy inference/rule-based multimetric decision may integrate metaheuristic search	Topology-change adaptation difficulty; 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	Rough set-based	Rough set theory (RST)-based selection/decision rules	Computational complexity; data sensitivity; dense-network scalability
Double-head clustering (DHC) [20], evolutionary approach for optimized VANET clustering, [56] multihead clustering algorithm [17]	The complexity of maintaining several CHs. It may require significant computational resources	Multihead clustering	Multiple CH roles; evolutionary/metaheuristic optimization in some variants	Maintenance overhead; computational resource demand

Table 1 (continued)

Proposed method	Limitations	Approach family	optimization basis	Key constraint type
Betweenness centrality-based clustering [12, 40, 41]	Calculating metrics such as betweenness centrality can be computationally intensive, especially in networks with high vehicle density	Centrality-based	Graph centrality metrics (e.g., betweenness/closeness)	Metric computation cost; density-driven overhead
Genetic algorithm-based clustering [5, 19, 26, 44]	Potential computational overhead due to the need to continually optimize the population with each iteration	GA / evolutionary-based	Genetic population-based iterative optimization	Iterative optimization overhead
Hybrid dog and beluga whale optimization [9]	This hybrid algorithm is sensitive to the configuration of the parameters	Hybrid swarm optimization	Hybrid swarm optimization (parameterized search)	Parameter sensitivity
MFO algorithm [1, 8, 36]	This algorithm is sensitive to increasing transmission range and node density	Swarm optimization (MFO)	Moth flame optimization-based search/selection	Range and density sensitivity
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	ACO-based / heuristic + distance-driven	ACO heuristic with PED estimation for selection/adjustment	Dependence on accurate PED; instability under rapid/irregular dynamics
Game theory-based clustering [2, 56]	If market conditions are unstable, the efficiency of spectrum allocation could suffer	Game-theoretic	Game-theoretic resource/spectrum decision mechanisms	Spectrum allocation efficiency sensitivity (as stated)
Hybrid fennec fox and sand cat optimization algorithm [57]	Combining FFOA and ISCOA is time- and power-consuming	Hybrid swarm optimization	Hybrid optimizer combining two swarm methods	Time and power consumption
CHFHB-DMNSGA) [14]	The computational complexity of the hybrid CHFHB-DMNSGA may lead to higher processing time	Hybrid multiobjective optimization	Hybrid multiobjective optimization (DMNSGA-based)	Computational complexity; higher processing time

Table 1 presents a comparative overview of the reviewed VANET clustering and cluster head selection approaches, including their methodological family, optimization basis, key constraint type, and the main limitations reported in the literature.

2.2 Research gaps and positioning

The surveyed literature shows that numerous approaches and algorithms have been proposed for clustering and routing in VANETs. Nevertheless, several research gaps and open issues remain that warrant further investigation.

- *Limited joint use of mobility, connectivity, and interaction quality in CH selection:* Many clustering and cluster head selection schemes mainly rely on high mobility or basic connectivity indicators, such as relative speed, direction to improve stability. Conversely, several trust-enabled or security-oriented approaches model node reliability and misbehavior using trust, reputation, or graph-based trust, while handling mobility and fine-grained link dynamics more coarsely. Overall, relatively few works define a transparent multimetric CH score that simultaneously fuses mobility, local connectivity, and communication quality with the explicit goal of reducing CH switches and cluster reconfigurations over time.
- *Energy and overhead are often optimized without explicitly managing cluster or CH dynamics:* Several protocols incorporate energy consumption, network lifetime, or control overhead into the design of clustering and routing, frequently using metaheuristic optimization to search for efficient configurations. While these methods can reduce energy use or improve throughput and delay, cluster head and cluster stability metrics (such as cluster head duration, CH change rate, or cluster lifetime) are often treated as secondary outputs and evaluated under limited traffic scenarios. As a result, the direct contribution of reduced CH or cluster changes to long-term overhead and implicit energy savings is not always made a primary design objective, particularly across different density regimes.
- *Algorithmic and tuning complexity in fuzzy logic and metaheuristic clustering schemes:* Fuzzy logic-based clustering and decision-making, as well as hybrid fuzzy–metaheuristic approaches, have been proposed to capture non-linear relationships among metrics and enhance stability or QoS in VANETs. In parallel, a variety of metaheuristic schemes (e.g., moth-flame optimization, prairie dog optimization, genetic and hybrid evolutionary algorithms) have been applied to clustering and routing decisions. However, fuzzy schemes typically require carefully designed membership functions and rule bases that are tuned to specific traffic patterns, and metaheuristic optimizers perform iterative population-based search. Under fast-changing topology and dense urban conditions, such tuning effort and computational overhead can be difficult to reconcile with strict real-time requirements for frequent clustering and CH updates.
- *Limited density-aware, online determination of the number of clusters.* Many clustering protocols either assume a fixed number of clusters or determine the cluster count offline for a given scenario, and then reuse this choice across differ-

ent traffic conditions. In practice, approaches such as multihead and double-head clustering, optimization-based weighted clustering, and several dynamic clustering schemes typically control the cluster count through static design parameters (for example, preconfigured values of K) rather than an online validity criterion that reacts to the current traffic density and topology. Although silhouette-based criteria have been studied for selecting the number of clusters in general k-means applications, they are usually applied in static or offline contexts, not as an online mechanism for adapting the number of clusters K_t at each decision interval in a highly dynamic VANET.

In summary, the reviewed literature shows that, while clustering and CH selection mechanisms for VANETs have made significant progress in terms of scalability, stability, and QoS, there is still a need for lightweight, density-aware solutions that: (i) jointly exploit mobility, connectivity, and communication or trust indicators in a single CH scoring model, (ii) explicitly target the reduction of CH switches and cluster reconfigurations to extend cluster lifetime and reduce overhead, and (iii) remain computationally feasible for frequent, real-time decisions under varying traffic densities. The proposed IMICLiVAN framework is positioned to address these gaps.

3 Proposed method

3.1 Overview of the proposed method

This section presents the method proposed to improve clustering and cluster head selection in VANETs. We aim to optimize the clustering process dynamically and ensure effective CHS 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. In practice, “real-time” operation means that clustering and CH decisions must be completed within each simulation/decision interval despite rapid vehicle mobility. As vehicle density increases, the workload of repeatedly evaluating clustering quality and computing per-vehicle metrics grows substantially, which may motivate parallel/distributed execution to preserve latency in large-scale deployments. Importantly, this scalability layer complements the lightweight design of IMICLiVAN rather than changing its algorithmic steps.

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

In Phase 1, the algorithm determines the optimal number of clusters and assigns each vehicle to a cluster. In Phase 2, the method computes the CH score for vehicles within each cluster and selects the highest-scoring vehicle as the cluster head, resulting in a set of clusters each associated with a CH.

For clarity, IMICLiVAN operates at each simulation time step as follows: (1) collect the current vehicle state from SUMO/TraCI, (2) build the clustering

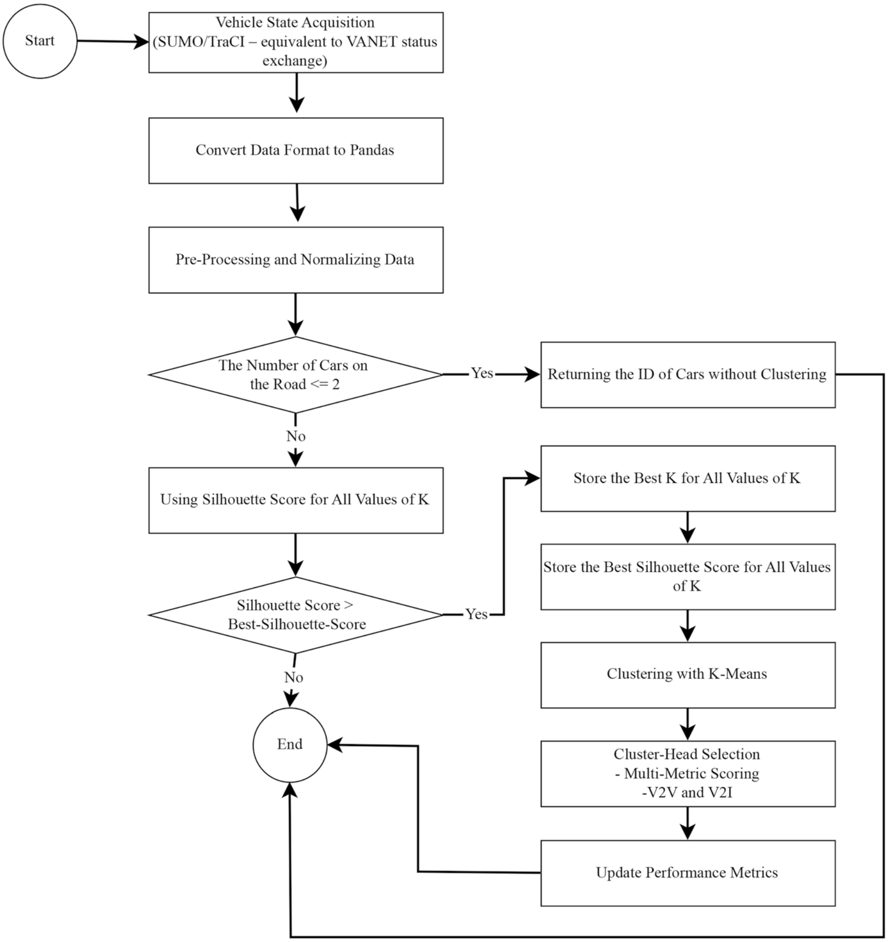


Fig. 2 Clustering vehicles in VANET using the K-means algorithm and the silhouette score method

feature matrix, (3) run K-means for candidate-K values and select $K^\star$ using the silhouette score, (4) assign vehicles to clusters using $K^\star$, and (5) compute the CH score in each cluster and select the highest-scoring vehicle as the CH. This step-by-step workflow is summarized in Fig. 2 and Algorithm 1.

3.1.1 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 groups vehicles into clusters, while the silhouette score determines the optimal number of clusters.

3.1.2 a. Data collection and preprocessing

We collect data from each vehicle during the simulation. The data include:

Vehicle Position (X, Y): The vehicle's position in the road network.

Lane ID: The lane number where the vehicle is currently located.

The collected data are formatted into a Pandas data frame for easier manipulation during the clustering process. Each vehicle's position is normalized using the standard scaler from Scikit-learn to ensure uniform scaling across all features.

3.1.3 b. Clustering architecture

The flowchart of Fig. 2 represents the process of clustering vehicles in VANET using the K-means algorithm and the silhouette score method. The process begins by initializing the clustering mechanism. At this step, real-time data on vehicles are collected from the simulation. The data include vehicle positions (X, Y coordinates) and lane numbers. This information is crucial for clustering as it defines the current state of the network. Once the data are collected, they are converted into a format that can be processed efficiently. In this case, the Pandas library organizes the data into a DataFrame, which facilitates the manipulation and analysis of vehicle information.

3.1.4 c. K-means clustering with silhouette score

Once the data are 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.

Specifically, we evaluate candidate values $K \in \{2, \ldots, 11\}$, compute the silhouette score for each clustering result, and select the value $K^{\star}$ that maximizes the overall silhouette score for that time step. The final clustering assignment for that time step is then obtained by running K-means with $K^{\star}$.

The choice of K-means in IMICLiVAN is motivated by the geometric nature of our scenario and the real-time constraints of VANET clustering. Vehicles move along a bidirectional multilane road segment, and the clustering features are strictly spatial and lane-level (position (x, y) and lane ID). In this setting, Euclidean distance provides a natural similarity measure, and K-means offers a lightweight, well-understood baseline for grouping vehicles into spatially compact clusters. Its per-iteration complexity $O(K \times n \times d)$ makes it suitable for execution at every simulation step ($\Delta t = 1$ s) under moderate and high densities, and it is widely adopted in VANET/IoV clustering studies as a practical geometry-driven clustering mechanism.

A key challenge in K-means-based clustering is the choice of the number of clusters K. Many existing VANET schemes fix K or tune it offline per scenario, which can lead to fragmented clusters in sparse conditions or overloaded, unstable clusters in dense conditions. In contrast, IMICLiVAN uses the silhouette score as an online clustering quality criterion. At each decision interval, we evaluate candidate values $K \in \{2, \ldots, 11\}$, compute the corresponding silhouette scores, and select the value

$K^\star$ that maximizes the overall score. The silhouette score jointly captures intra-cluster compactness and inter-cluster separation using only pairwise distances, which makes it a natural internal validity index for our Euclidean, geometry-driven setting. Without such a criterion, the number of clusters would have to be fixed or tuned manually per scenario, contradicting our goal of an online, density-aware clustering mechanism. Overall, this yields a density- and topology-aware adaptation of the cluster count K_t over time while keeping the clustering step lightweight enough for real-time operation.

3.1.5 Cluster head selection phase

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

3.1.6 a. 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 are local distance (LD), relative speed (RS), trust (T), base station distance (BSD), answer ratio (AR), node degree (ND), and node center (NC).

The weighted formula for CH selection is given by Eq. 1:

$$CH = W_{\mathrm{LD}} \cdot \mathrm{LD} + W_{\mathrm{RS}} \cdot \mathrm{RS} + W_{\mathrm{T}} \cdot \mathrm{T} + W_{\mathrm{BSD}} \cdot \mathrm{BSD} + W_{\mathrm{AR}} \cdot \mathrm{AR} + W_{\mathrm{ND}} \cdot \mathrm{ND} + W_{\mathrm{NC}} \cdot \mathrm{NC} \tag{1}$$

where the sum of all weights equals 1. We use $W_{\mathrm{T}} = 0.4$ and set all other weights in Eq. (1) to 0.1 each, selected empirically to prioritize trust for CH stability while keeping the remaining factors balanced. The vehicle with the highest CH score in each cluster is selected as the cluster head. Although T and AR are both related to communication reliability, they capture complementary aspects: T reflects long-term forwarding cooperation and reputation as perceived by the cluster, while AR reflects the short-term responsiveness and availability of the node for its own communication. Including both terms in the CH score allows IMICLiVAN to avoid selecting nodes that are either historically untrustworthy (low T) or temporarily overloaded/unresponsive (low AR), even if they score well on the other metrics.

3.1.7 b. Metrics calculation

Each of the above metrics is calculated as follows:

Local Distance (LD): This metric captures the *total* distance between a vehicle and the other vehicles in its cluster, that is computed by Eq. 2:

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

The formula calculates the Euclidean distance between vehicle i and each other vehicle i in the same cluster. A vehicle with a smaller total distance is more central within the cluster 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.

A simple and consistent way to compute this metric is to aggregate the speed deviation of vehicle *i* from the other vehicles in its cluster, which is shown in Eq. 3:

$$RS_i = \sum_{v \in C} \left| s_i - s_v \right| \tag{3}$$

where s_i and s_v denote the speeds of vehicles *i* and *v*, respectively. Smaller deviations indicate better speed similarity within the cluster.

Trust (T): Trust measures the reliability of a vehicle in terms of communication (e.g., forwarding messages). Vehicles with a higher trust score are more reliable for inter-vehicle communication. Trust is divided into direct trust and indirect trust. Direct trust is calculated based on a vehicle's past success in forwarding packets within the cluster that as shown in Eq. 4:

$$T_{\text{direct}} = \frac{\text{Packets forwarded successfully}}{\text{Total packets sent}} \tag{4}$$

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, i.e., how much they trust the current vehicle based on their interactions, which is shown in Eq. 5:

$$T_{\text{indirect}} = \sum_{v \in C} T_{\text{direct}}(v) \times D(i, v), \quad T = \lambda \times T_{\text{direct}} + (1 - \lambda) \times T_{\text{indirect}} \tag{5}$$

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

Conceptually, T_i is designed as a longer-term reputation indicator of node *i*'s cooperative forwarding behavior. The direct trust component is accumulated from the success ratio of packets that *i* forwards or transmits on behalf of others over time, while the indirect trust aggregates the opinions of neighbors about *i*'s past behavior. Therefore, T_i primarily reflects how reliably node *i* has participated in multihop forwarding for other vehicles over a longer observation horizon, rather than its short-term responsiveness to its own traffic.

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. The BSD computation is in Eq. 6:

$$BSD_i = \sqrt{(x_i - x_{BS})^2 + (y_i - y_{BS})^2} \tag{6}$$

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. The AR computation is in Eq. 7:

$$AR_i = \frac{\text{Packets received}}{\text{Packets sent}} \tag{7}$$

In contrast to trust, the answer ratio AR_i captures the short-term responsiveness of node i to its own communication activities. Here, "packets sent" and "packets received" refer to the application-level requests and corresponding replies that are originated by node i and addressed to it within the current observation window. A high AR_i indicates that node i is currently reachable and able to successfully complete its own communication exchanges, whereas a low AR_i reveals temporary overload, local channel problems, or lack of responsiveness, even if its long-term forwarding reputation T_i is still high.

Node Degree (ND): The number of direct connections (D (i,v)) a vehicle has with others in its communication range (R). The ND equation is given in Eq. 8:

$$ND_i = \sum_{v \in C} I(D(i, v) \leq R) \tag{8}$$

Here, $D(i, v)$ denotes the Euclidean distance between vehicles i and v, and R is the communication range.

Node Center (NC): This metric reflects whether a vehicle is positioned on a central lane of the road segment (a lane-centric indicator). Vehicles located on the designated central lane are favored, as they tend to maintain more stable neighborhood relations under lane-level mobility patterns. For example, vehicles in the left and right lanes are more likely to overtake other vehicles or leave the roadway, making them less stable than vehicles in the center lane.

Algorithm 1 summarizes the time step execution of IMICLiVAN. Lines 9 through 14 collect and normalize the current vehicle state. Lines 15 through 19 perform candidate-K evaluation and select $K^\star$ using the silhouette score, followed by clustering with $K^\star$. Lines 20 through 27 compute the per-vehicle CH score using Eq. 1 and select the highest-scoring vehicle as the CH in each cluster. Finally, lines 28 through 36 track CH and cluster changes and update the reported evaluation measures.

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. The silhouette score S(i) for a vehicle is computed as in Eq. 9:

$$S(i) = \frac{b(i) - a(i)}{\max\left(a(i), b(i)\right)} \tag{9}$$

where:

$a(i)$ is the mean intra-cluster distance (average distance between i and all other vehicles in its cluster), and $b(i)$ is the mean nearest-cluster distance (average distance between i and vehicles in the nearest neighboring cluster).

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.

3.1.8 Parallelization and distributed execution opportunities

IMICLiVAN can be executed in a fully onboard manner for moderate densities; however, for dense/city-scale scenarios under strict latency constraints, several steps are naturally parallelizable:

- *Candidate-K evaluation:* each K-means run and its corresponding silhouette score can be computed independently (e.g., $K = 2$ to 11).
- *Distance/silhouette computations:* the distance calculations involved in silhouette scoring can be parallelized across vehicles or spatial partitions (e.g., road segments).
- *Per-cluster CH scoring:* the weighted CH score can be computed concurrently for vehicles within each cluster, and clusters can be processed in parallel.

These opportunities enable parallel and distributed realizations on RSU/MEC nodes or cloud/HPC backends when required, without changing the algorithmic steps or assumptions of IMICLiVAN.

Algorithm 1 Cluster head selection and clustering

```
Input:
$\mathcal{V}(t)$                                              ▷ Set of active vehicles at time step $t$
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 $k = 2$ to $k = 11$ do
        Perform K-means
        Perform Silhouette Score          ▷ $Silhouette\ Score = \frac{b(i)-a(i)}{\max(a(i),b(i))}$
    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
            $CHScore(i) = W_{LD} \times LD(i) + W_{RS} \times RS(i) + W_T \times T(i) + W_{BSD} \times BSD(i) + W_{AR} \times AR(i)$
                          $+ W_{ND} \times ND(i) + W_{NC} \times NC(i)$
        end for
        Select vehicle with the highest $CHScore$ as cluster head
    end for
    Step 6: Track Cluster-Head and Cluster Changes
    if prevCH[c] $\neq \emptyset$ and CH$_c(t) \neq$ prevCH[c] then
        $N_{\mathrm{CH}} \leftarrow N_{\mathrm{CH}} + 1$                  ▷ cluster-head change counter
    end if
    if prevMembers[c] $\neq \emptyset$ and Member$_c(t) \neq$ prevMembers[c] then
        $N_{\mathrm{Clus}} \leftarrow N_{\mathrm{Clus}} + 1$              ▷ cluster membership change counter
    end if
    prevCH[c] $\leftarrow$ CH$_c(t)$;  $prevMembers$[c]$\leftarrow$ Member$_c(t)$   ▷ update previous state for next step
    Finalize lifetimes for clusters that disappeared since $t - 1$       ▷ Calculating destroyed clusters
end while
Return final statistics: cluster head changes, cluster changes, lifetime of clusters over time
```

3.2 Computational complexity of the proposed method

The computational complexity of the proposed method is analyzed to quantify the overhead introduced by the clustering process and the cluster head selection mechanism. The clustering process involves three key steps. First, data normalization processes n data points with d features, resulting in a complexity of $O(n \times d)$. Second, the K-means clustering algorithm divides n vehicles into K clusters over t iterations, with each data point having d features, yielding a complexity of $O(K \times n \times t \times d)$. Third, the silhouette score computation evaluates the clustering quality by calculating pairwise distances for all n data points within clusters. This step dominates the clustering process with a complexity of $O(K \times n^2 \times d)$. Consequently, the overall complexity of the clustering process is $O(K \times n^2 \times d)$, as the silhouette score computation is the most computationally intensive step.

For the cluster head selection mechanism, seven metrics are calculated for each vehicle in a cluster. These metrics include local distance, relative velocity, trust (direct and indirect), base station distance, answer ratio, node degree, and node center. Most of these computations are neighbor-aggregation operations within the same cluster; thus, for a cluster of size m, the per-vehicle cost is $O(m)$, and scoring all vehicles in that cluster costs $O(m^2)$. Therefore, over c clusters, the total cost of cluster head selection is $O\left(\sum_{i=1}^{c} m_i^2\right)$, which is upper-bounded by $O(c \cdot m_{\max}^2)$ (and by $O(n^2)$ in the worst case), where m_i is the size of cluster i, $m_{\max}$ is the maximum cluster size, and n is the total number of vehicles. By combining the clustering process and cluster head selection mechanism, the proposed method's total computational complexity is $O(K \times n^2 \times d) + O\left(\sum_{i=1}^{c} m_i^2\right)$. This analysis highlights IMICLiVAN's efficiency and scalability for both moderate, and high-density vehicular networks, ensuring its applicability in real-world scenarios. The complexity analysis also indicates why parallel/distributed processing becomes relevant at scale: the silhouette-based evaluation term $O(K \times n^2 \times d)$ can dominate as n increases in dense/city-scale deployments, while CH scoring across clusters/vehicles is inherently parallel. Therefore, although IMICLiVAN remains practical for moderate densities, dense scenarios with strict latency constraints can motivate parallelizing candidate-K evaluation, distance/silhouette computations, and per-cluster CH scoring using RSU/MEC or cloud/HPC resources.

4 Experimental results and discussion

Section 4 evaluates the performance of the proposed hybrid clustering approach called "An Improved Method to Increase Cluster Lifetime in Vehicular Ad Hoc Networks (IMICLiVAN)" method through simulations conducted under two vehicular density scenarios: one with 20 vehicles and another with 100 vehicles, representing moderate and high-density urban traffic conditions, respectively. The proposed method is compared against three state-of-the-art clustering approaches: weighted trusted cluster head selection (WTCHS), enhanced clustering-based routing protocol with Sugeno fuzzy inference system (ECBLTR), and cluster-based multicast optimized routing with elite knowledge-sharing genetic algorithm (EKSGA).

To improve clarity, we define the main clustering-level stability metrics used throughout Sect. 4. Let t_k denote the k-th simulation time instant ($k = 1, \ldots, T$). At each time instant, vehicles are partitioned into clusters, each with a designated cluster head (CH). We evaluate stability using three metrics:

(i) *Cluster lifetime:* for each observed cluster j, with creation time $t_{\text{start},j}$ and termination time $t_{\text{end},j}$, its lifetime is $L_j = t_{\text{end},j} - t_{\text{start},j}$, and the reported value is the average $\overline{L}$ over all observed clusters.

(ii) *Cluster head (CH) changes:* For each cluster c, a CH change is counted whenever $\text{CH}_c(t_k) \neq \text{CH}_c(t_{k-1})$ between consecutive instants. The reported value is the total number of CH changes accumulated over all clusters during the simulation.

(iii) *Cluster changes:* a cluster change is counted whenever the membership set of a cluster changes between two consecutive instants, i.e., $M_c(t_k) \neq M_c(t_{k-1})$. The reported value is the total number of membership changes accumulated over all clusters.

Detailed formulations are provided in the corresponding subsections for completeness. Key metrics include network lifetime, cluster head changes, cluster changes, energy efficiency, delay, overhead, throughput, PDR, packet loss, and jitter.

As discussed in Sect. 2, a wide spectrum of VANET clustering and CH selection approaches has been proposed, including trust-based, fuzzy logic, evolutionary/metaheuristic, hybrid swarm-based, and game-theoretic designs. In this work, our main goal is to evaluate IMICLiVAN as a stability-oriented, single-head clustering mechanism under a homogeneous SUMO/TraCI-based setting. For the detailed simulation comparison, we therefore focus on three representative schemes that are both closely related to our problem formulation and practically implementable in our framework: a fuzzy-based clustering protocol, an evolutionary (genetic) scheme, and a trust-based weighted CH selection method.

The fuzzy-based protocol (ECBLTR) operates in a bidirectional multilane environment and aims to improve network lifetime using a fuzzy inference system that combines, among others, the number of neighbors, the distance to the base station, and residual energy in the CH decision. In its clustering stage, however, the features are primarily location-based with a fixed clustering dimension; lane and driving direction are not explicitly modeled. As a result, vehicles that are momentarily close in space but travel in opposite directions (e.g., on different lanes) may be grouped into the same cluster, which can lead to frequent cluster and CH reconfigurations as they diverge. In IMICLiVAN, we explicitly incorporate direction- and mobility-related indicators in the clustering process (via SUMO lane/direction information and relative speed–based metrics) to mitigate this effect, and we also make use of connectivity and infrastructure-related quantities such as the distance to the base station and the local node density in the CH scoring rule.

The EKSGA scheme represents an evolutionary/metaheuristic approach that further develops the fuzzy-based design and compares itself against it, with the primary objective of improving energy consumption and throughput. Its CH selection relies on a five-factor weighted fitness function, with a strong emphasis on energy-related terms and with intra-/inter-cluster quantities encouraging compact CH-centric structures. In our SUMO/TraCI-based implementation, we also observe improved throughput and energy indicators for EKSGA compared with the fuzzy baseline, but at the cost of substantially higher end-to-end delay. The original paper mainly emphasizes the energy/throughput gains and does not analyze this delay trade-off in detail. IMICLiVAN shares the underlying intuition of favoring compact clusters, but implements it at the clustering level via K-means and an online silhouette-based selection of the number of clusters, while explicitly reporting CH/cluster changes and delay/jitter as part of the evaluation.

Finally, the trust-based method introduces a weighted CH scoring rule that combines vehicle speed, a trust factor, and pairwise distance between vehicles. IMICLiVAN builds upon this idea by retaining the trust component and generalizing the mobility and distance terms: instead of using only pairwise quantities, we compute relative speed and local distance aggregated over all neighbors within a cluster. This aims to select CHs that are more stable with respect to the entire cluster neighborhood. In addition, our scoring rule integrates further connectivity and communication quality indicators, which are then evaluated jointly in terms of CH stability and network-level performance.

Taken together, WTCHS (trust-based), ECBLTR (fuzzy-based), and EKSGA (evolutionary/metaheuristic) thus provide three representative baselines from different methodological families, all of which explicitly address clustering and CH selection under dynamic vehicular conditions. Other families surveyed in Table 1 are certainly relevant and provide complementary perspectives; however, many of them rely on multihead or backup-CH models that differ from our single-head formulation, or employ highly complex hybrid metaheuristics with substantial tuning and computational demands that make a fully homogeneous reimplementation in our simulation framework beyond the scope of the present study. Extending the evaluation of IMICLiVAN to such additional families is an interesting direction for future work.

4.1 Simulation environment

This study used SUMO (Simulation of Urban Mobility) as the primary simulation environment, integrated with Python via the TraCI library for real-time control and management of vehicular traffic [60]. SUMO was selected due to its scalability, high precision in modeling vehicular behavior, and open-source nature, which enables extensive customization for dynamic VANET scenarios. The SUMO scenario files were generated with SUMO v1.18.0. The online controller was implemented in Python using NumPy, Pandas, scikit-learn (K-means and silhouette score), and Matplotlib.

In SUMO, the simulation advances in discrete time steps of duration Δt. We set the step length to $\Delta t = 1.0$ s (i.e., `step length = 1.0` in the SUMO configuration). At each time step, we call `simulationStep()` via TraCI and

Table 2 Tools and software environment

Component	Details
Mobility simulator	SUMO v1.18.0
Online control	TraCI (SUMO interface) + Python controller
SUMO movement model	Krauss car-following model
Clustering library	scikit-learn (K-means, silhouette score)
Data processing/plots	NumPy, Pandas, Matplotlib

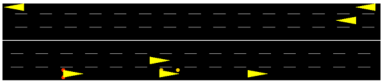


Fig. 3 One snapshot of vehicles traveling

then execute clustering and network-level evaluation. Therefore, the total simulation time is given by

$$T_{\text{sim}} = N_{\text{steps}} \times \Delta t = N_{\text{steps}} \times 1.0 \text{ s}. \tag{10}$$

Table 2 summarizes the tools and software environment used in this paper.

4.1.1 Topology and mobility

The road network comprises a 1000-unit bidirectional stretch with three lanes per direction, featuring priority junctions at both ends and internal turning lanes for smooth transitions. Vehicles were simulated with staggered departure times at 1-second intervals. Lane speeds were set to 13.89 m/s (~50 km/h) under normal conditions and reduced to 3.65 m/s (~13 km/h) at junctions to mimic realistic slowdowns. Vehicular movement was modeled using SUMO's Krauss car-following model to capture acceleration, deceleration, and safe-following behavior [61]. A single base station (BS) was deployed near the middle of the road at $(x, y) = (500, 100)$. Figure 3 is one snapshot of a vehicle flow scenario.

4.1.2 Densities

We evaluate two vehicular densities: 20 vehicles (moderate density) and 100 vehicles (high density), representing moderate and congested urban traffic conditions, respectively. Table 3 presents the parameters of the simulation environment.

4.1.3 Network-level communication model

We embed a fair traffic load and a consistent standard PHY/MAC communication model on top of SUMO/TraCI to evaluate the impact of clustering on network-level performance. At each simulation step (Δt = 1 s), we generate a constant offered load of up to 50 application-layer data reports, independently of the current number

Table 3 Simulation parameters

Parameter	Value
Number of nodes	20, 100
Number of lanes per direction	3
Road length	1000 units (~ 1 km)
Vehicle speed	13 to 50 km/h
Number of base stations	1
Base station coordinates	(500, 100)
Base station transmission range	~ 250 m
Simulation step length	1 s

of clusters or cluster sizes. Report sources are uniformly selected from the active vehicles. Each report is delivered using a cluster-assisted two-hop pattern: a member vehicle forwards its packet to its current cluster head (vehicle–to–vehicle, V2V), and the cluster head relays the packet to a roadside base station (vehicle–to–infrastructure, V2I); if the source is itself the cluster head, only the V2I hop is used. This pattern is identical for all compared schemes. For each hop, we apply a PHY/MAC model. We assume a one-hop radio communication range of 250 m and use a log-distance path-loss model with log-normal shadowing and a fixed transmit power and receiver sensitivity. Links beyond 250 m or below the sensitivity threshold are treated as failed. In addition, the packet success probability is coupled to the estimated channel utilization (based on the aggregate data + control bytes scheduled in the current step) through an exponential load-dependent term, to emulate collision and queue-overflow effects in a congested random-access channel. The per-hop delay includes propagation, transmission time (for a 512-byte data payload at 6 Mbps), a MAC/queuing component modeled as an exponential random variable whose mean increases with channel utilization and cluster churn, and a small processing delay. The network parameters are noted in Table 4. We record, on a per-report basis, whether the packet is successfully delivered to the base station, as well as its end-to-end delay. From these records, we derive the packet delivery ratio (PDR), average end-to-end delay, average jitter, and throughput. We also accumulate the total number of transmitted data and control bytes over the simulation. This quantity serves as a measure of the communication load, is used to compute the control overhead, and provides the basis for deriving the TX/RX times in the state-based energy model introduced in Sect. 4.5.6. All model parameters and mechanisms in this network layer are shared across WTCHS, ECBLTR, EKSGA, and IMICLiVAN, ensuring a fair comparison; the only difference between schemes is their clustering and cluster head selection behavior, which was our purpose from the first place.

PHY/MAC parameters follow IEEE 802.11p/DSRC settings and widely used VANET simulation configurations. The transmission range, data rate, and packet sizes are aligned with Table 3 of [62], which also uses 250 m range, 6 Mbps, and 512-byte packets for IEEE 802.11p VANET simulations. The path-loss anchor is set to the free-space loss at 1 m and 5.9 GHz ($PL_0 \approx 47$ dB) with exponent $n = 2.5$, following the 5.9 GHz V2V channel model recommended in [63], and the log-normal

Table 4 Network parameters used in the SUMO/TraCI-based communication model

Parameter	Value
Transmission range R_{comm}	250 m
packet size B_{data}	512 bytes
Control message sizes	HELLO: 64 B, JOIN/LEAVE: 96 B, TABLE: 256 B
Data rate R_{data}	6 Mbps
Transmit power P_{tx}	20 dBm
Path-loss at 1 m, PL_0	47 dB
Path-loss exponent n	2.5
Shadowing σ	4 dB
Receiver sensitivity $P_{\text{rx,min}}$	−90 dBm
Base MAC/queuing delay (mean)	10 ms
Processing delay per hop	1 ms
Maximum application reports per step	50
Load, success coupling parameter α	3.0
Load, delay scaling parameter β	4.0
Radio receive power P_{rx}	70 mW
Radio idle/listen power P_{idle}	50 mW

shadowing standard deviation $\sigma = 4$ dB is chosen within the 3.3 dB to 6.8 dB range reported there. The transmit power of 20 dBm and receiver sensitivity around −90 dBm follow validated IEEE 802.11p/ITS-G5 configurations in simulation and hardware studies [64]. The TX/RX/idle power ratios in our analytical energy model are consistent with per-state power measurements for IEEE 802.11 network interfaces.

4.1.4 Online clustering and control

The integration of Python and TraCI enables dynamic decision-making during simulations, including real-time cluster formation and cluster head selection. This online control loop is essential for evaluating the proposed IMICLiVAN method under realistic mobility dynamics.

This section evaluates the performance of the proposed IMICLiVAN method through simulations conducted under two vehicular density scenarios (20 and 100 vehicles). The proposed method is compared against three state-of-the-art clustering approaches: WTCHS, ECBLTR, and EKSGA. In addition to clustering-level indicators (cluster lifetime, cluster head changes, and cluster changes), Sect. 4.5 reports network-level metrics (PDR, throughput, end-to-end delay, jitter, overhead, and energy efficiency) to assess the dissemination efficiency of clustering in VANETs.

4.2 Cluster lifetime

The cluster lifetime refers to the duration for which a cluster remains stable without major re-clustering events or cluster head changes.

Let $t_{\mathrm{start},j}$ and $t_{\mathrm{end},j}$ denote, respectively, the creation and termination times of cluster j. The lifetime of cluster j is:

$$L_j = t_{\mathrm{end},j} - t_{\mathrm{start},j}. \tag{11}$$

Let N_{cl} be the total number of clusters observed during the simulation. The average cluster lifetime is then given by:

$$\overline{L} = \frac{1}{N_{\mathrm{cl}}} \sum_{j=1}^{N_{\mathrm{cl}}} L_j, \tag{12}$$

where L_j is the lifetime of cluster j and $\overline{L}$ denotes the average cluster lifetime.

The reported value corresponds to the average lifetime over all clusters observed during the simulation. This metric is critical for maintaining network efficiency and reducing communication overhead in VANETs. As illustrated in Fig. 4, the results for the 20-vehicle scenario demonstrate that IMICLiVAN achieved a network lifetime of 5.78 time units, significantly outperforming WTCHS (2.01 time units, a 187% increase), ECBLTR (4.65 time units, a 24% increase), and EKSGA (1.03 time units, a 461% increase). This substantial improvement highlights the method's ability to maintain stable clusters under moderate vehicular density. Similarly, as shown in Fig. 5, the results for the 100-vehicle scenario confirm IMICLiVAN's scalability, achieving a network lifetime of 2.54 units. This performance remained significantly better than WTCHS (1.41 time units, an 80% improvement), ECBLTR (2.26 time units, a 12% improvement), and EKSGA (0.83 time units, a 206% improvement). Although a reduction in lifetime was observed due to increased traffic density, IMICLiVAN's ability to maintain stability under high-density conditions demonstrates its robustness and adaptability.

The significant improvement in network lifetime is rooted in IMICLiVAN's K-means-based clustering process and its integration with the silhouette score. By dynamically determining the optimal number of clusters using the silhouette score,

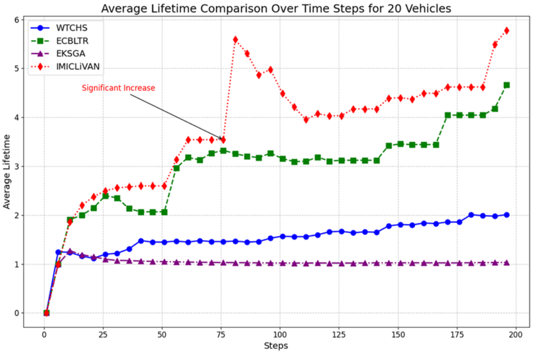


Fig. 4 Comparison of the lifetime of IMICLiVAN with WTCHS, ECBLTR, and EKSGA with 20 vehicles

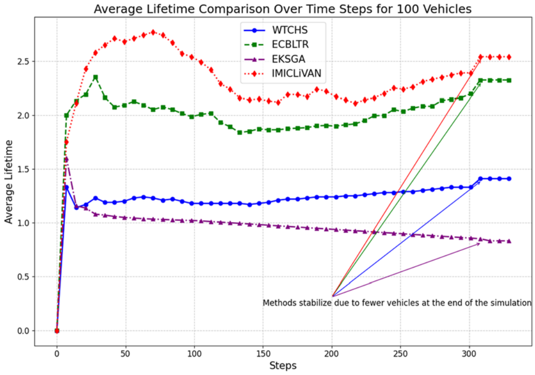


Fig. 5 Comparison of the lifetime of IMICLiVAN with WTCHS, ECBLTR, and EKSGA with 100 vehicles

IMICLiVAN ensures clusters are compact and well-separated, leading to more efficient communication within clusters and fewer reformation events. Unlike other methods, which rely on predefined or static clustering parameters, this adaptive approach reduces instability caused by suboptimal clustering. Furthermore, the weighted formula for cluster head selection, which evaluates candidates based on a combination of metrics (e.g., mobility patterns and connectivity), ensures that the most stable and influential nodes are chosen as cluster heads. This minimizes unnecessary cluster head changes, further extending the network lifetime. As illustrated in the charts, Initial Similarity is observed at the beginning of both simulations, where all methods exhibit comparable network lifetime values due to uniform initial conditions. Over time, IMICLiVAN transitions into stable trends, where it consistently outperforms other methods by stabilizing at a higher network lifetime, indicating equilibrium in clustering performance. IMICLiVAN's Dominance is evident throughout the simulations, maintaining significantly longer network lifetimes than competing methods in both density scenarios. These results validate IMICLiVAN as a robust and scalable solution for vehicular clustering, ensuring prolonged network stability even under challenging conditions.

Cluster stability plays an important role in improving communication efficiency in VANETs. Metrics such as cluster lifetime and average cluster lifetime are used to evaluate stability. These metrics highlight how prolonged cluster lifetimes reduce communication overhead, enhance routing consistency, and improve packet delivery efficiency. Our simulations demonstrate that IMICLiVAN significantly outperforms existing methods by achieving higher average cluster lifetimes, directly correlating with reduced communication latency.

4.3 Cluster head changes

Cluster head changes reflect the frequency with which the cluster head is reassigned.

Let t_k denote the k-th simulation time instant (or step), $k = 1, \dots, T$, and let $\mathrm{CH}_c(t_k)$ be the cluster head of cluster c at time t_k. The number of cluster head changes for cluster c is:

$$N_{\mathrm{CH},c} = \sum_{k=2}^{T} \mathbf{1}\big(\mathrm{CH}_c(t_k) \neq \mathrm{CH}_c(t_{k-1})\big), \tag{13}$$

where $\mathbf{1}(\cdot)$ is the indicator function, equal to 1 if its argument is true and 0 otherwise. The total number of cluster head changes in the network is:

$$N_{\mathrm{CH,total}} = \sum_{c=1}^{N_{\mathrm{cl}}} N_{\mathrm{CH},c}, \tag{14}$$

where N_{cl} denotes the total number of clusters.

Fewer CH changes indicate better cluster stability, reducing communication delays and overhead. As illustrated in Fig. 6, the results for the 20-vehicle scenario show that IMICLiVAN recorded 86 CH changes, significantly fewer than WTCHS (257 CH changes, a 66% reduction), ECBLTR (139 CH changes, a 38% reduction), and EKSGA (422 CH changes, an 80% reduction). This demonstrates IMICLiVAN's superior ability to maintain stable cluster leadership under moderate vehicular density. Similarly, as shown in Fig. 7, the results for the 100-vehicle scenario highlight that IMICLiVAN sustained its performance, recording 176 CH changes. This is significantly lower than WTCHS (573 CH changes, a 69% reduction), ECBLTR (422 CH changes, a 58% reduction), and EKSGA (730 CH changes, a 76% reduction). IMICLiVAN consistently outperformed the competing methods despite the increased vehicular density, demonstrating its scalability in high-density traffic conditions.

The reduction in CH changes is attributed to IMICLiVAN's weighted method for cluster head selection, which evaluates nodes based on a combination of metrics such as mobility patterns and connectivity. This multicriteria approach ensures

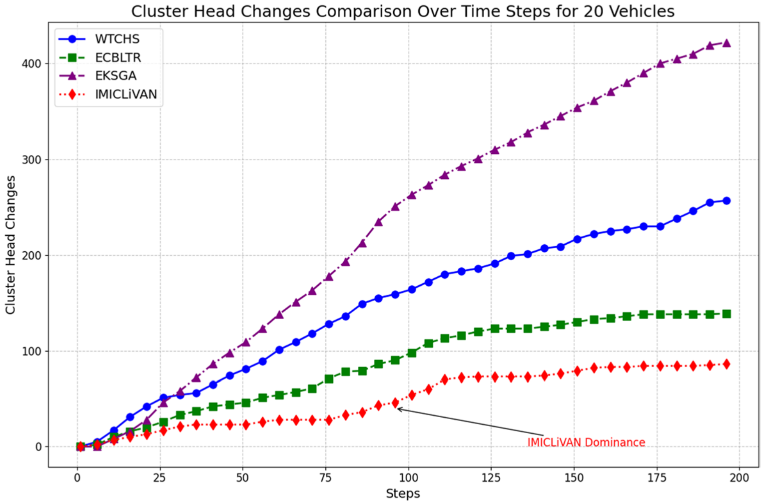


Fig. 6 Comparison of the cluster head changes of IMICLiVAN with WTCHS, ECBLTR, and EKSGA with 20 vehicles

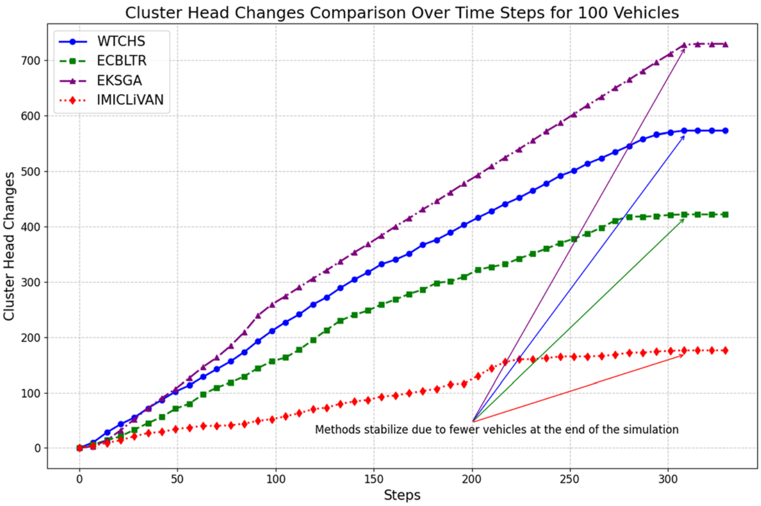


Fig. 7 Comparison of the cluster head changes of IMICLiVAN with WTCHS, ECBLTR, and EKSGA with 100 vehicles

that the most stable and suitable nodes are elected as cluster heads, reducing the frequency of reassignment. In contrast, methods like WTCHS and EKSGA rely on simpler or static selection criteria, making them prone to frequent reassignments as vehicular mobility increases. Moreover, by leveraging K-means clustering with silhouette scoring, IMICLiVAN ensures that clusters are optimally sized and balanced, which reduces the stress on cluster heads and prolongs their tenure. As shown in Fig. 6 and 7, Initial similarity is observed at the start of the simulation when clustering begins with uniform conditions. However, IMICLiVAN transitions into stable trends as the simulation progresses, consistently maintaining fewer CH changes due to its adaptive cluster management and stability-driven selection processes. The method's dominance is evident in both density scenarios, as it achieves significantly fewer CH changes than competing approaches, minimizing overhead and improving overall network efficiency.

As shown in Fig. 8, enabling the trust component within IMICLiVAN reduces the frequency of cluster head (CH) re-elections in both density scenarios. Specifically, the number of CH changes decreases from 92 to 86 for 20 vehicles and from 180 to 176 for 100 vehicles when trust is enabled. This behavior is scientifically expected because trust acts as a reliability filter during CH selection: nodes that exhibit inconsistent forwarding behavior (or are evaluated as less dependable through direct/indirect observations) are less likely to be elected as CHs. Consequently, CH leadership becomes more consistent over time, which reduces unnecessary CH oscillations and stabilizes intra-cluster coordination. From a network perspective, fewer CH changes directly translate into fewer reconfiguration events (e.g., CH announcements, re-association/join messages, and cluster-table updates), thereby lowering control-plane overhead. Since control signaling and repeated re-associations consume transmission resources, this reduction in overhead also decreases the overall transmission

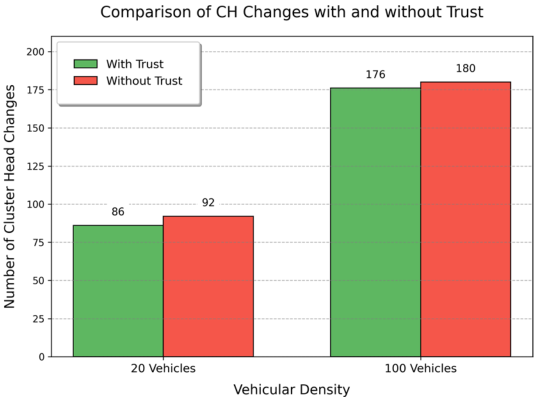


Fig. 8 cluster head changes with and without trust (20 and 100 vehicles)

effort and can improve energy efficiency (in terms of transmission-bytes/effort), while freeing additional channel capacity for data dissemination.

4.4 Cluster changes

Cluster changes measure how frequently a cluster needs to be reorganized or reformed due to shifts in vehicle positions.

Let $M_c(t_k)$ denote the set of member vehicles belonging to cluster c at time t_k. A cluster change is registered whenever the membership set changes between two consecutive instants. The number of cluster changes for cluster c is defined as:

$$N_{\text{Clus},c} = \sum_{k=2}^{T} \mathbf{1}\big(M_c(t_k) \neq M_c(t_{k-1})\big), \tag{15}$$

where $\mathbf{1}(\cdot)$ is the indicator function and T is the total number of simulation time instants. The total number of cluster changes in the network is then:

$$N_{\text{Clus,total}} = \sum_{c=1}^{N_{\text{cl}}} N_{\text{Clus},c}, \tag{16}$$

with N_{cl} denoting the total number of clusters.

Fewer cluster changes mean greater stability and less overhead in managing the dynamic nature of the network. As illustrated in Fig. 9, for the 20-vehicle scenario, IMICLiVAN recorded 80 cluster changes, significantly fewer than WTCHS (285 cluster changes, a 72% reduction), ECBLTR (120 cluster changes, a 33% reduction), and EKSGA (470 cluster changes, an 83% reduction). Similarly, as shown in Fig. 10, for the 100-vehicle scenario, IMICLiVAN recorded 270 cluster changes, significantly outperforming WTCHS (649 cluster changes, a 58% reduction), ECBLTR (392 cluster changes, a 31% reduction), and EKSGA (843 cluster changes, a 68% reduction). While cluster changes increase with higher vehicular density,

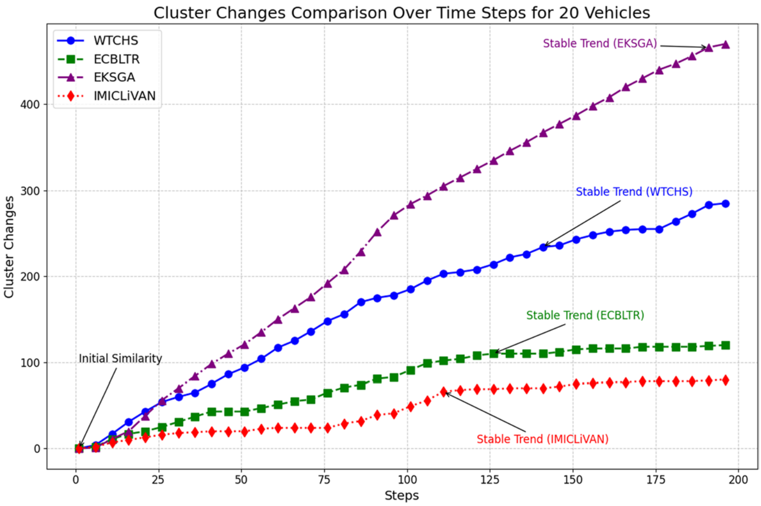
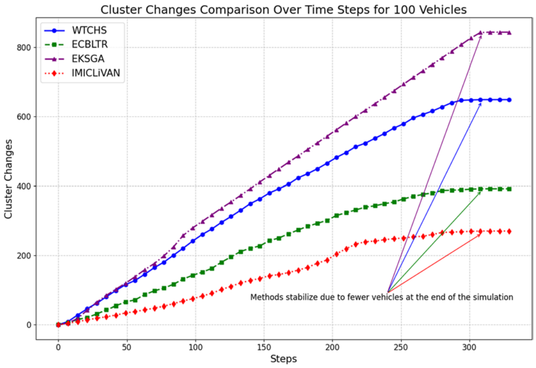


Fig. 9 Comparison of the cluster changes of IMICLiVAN with WTCHS, ECBLTR, and EKSGA with 20 vehicles

Fig. 10 Comparison of the cluster changes of IMICLiVAN with WTCHS, ECBLTR, and EKSGA with 100 vehicles



IMICLiVAN ensures significantly fewer changes than competing methods, demonstrating its scalability and robustness.

The significant reduction in cluster changes achieved by IMICLiVAN stems from its K-means-based clustering mechanism and the use of the silhouette score to determine the optimal number of clusters dynamically. IMICLiVAN minimizes the need for frequent reorganization, even under dynamic vehicular movement, by ensuring that clusters are compact and well-separated. Unlike competing methods, such as WTCHS and EKSGA, which rely on static or suboptimal clustering parameters, IMICLiVAN's adaptive approach results in fewer disruptions and higher clustering stability. Additionally, the weighted cluster head selection formula evaluates potential cluster heads based on mobility and connectivity, ensuring that stable and

Fig. 11 Overall comparison of simulation results with 20 vehicles

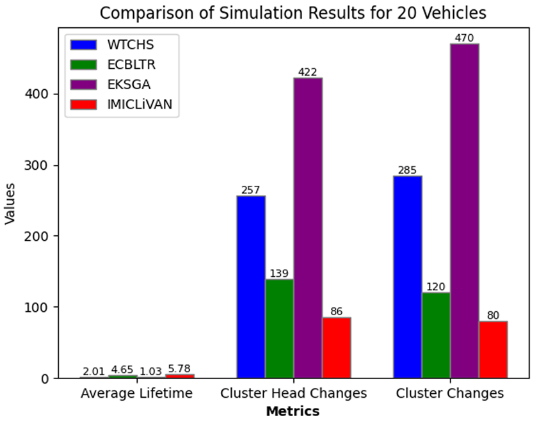
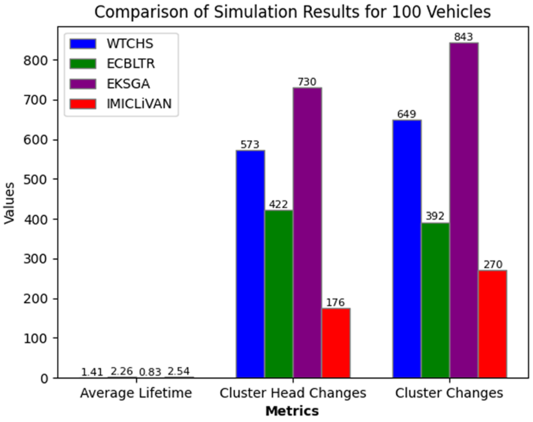


Fig. 12 Overall comparison of simulation results with 100 vehicles



well-suited cluster heads are selected. This reduces the ripple effects of cluster head changes on overall cluster stability, contributing to fewer cluster changes.

In conclusion, as illustrated in Figs. 11 and 12, the IMICLiVAN clustering method offers significant advantages over existing approaches by holistically and balancedly addressing the key challenges of VANETs. By integrating multiple metrics, 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 WTCHS, ECBLTR, and EKSGA algorithms, which tend to over-rely on specific metrics and face challenges in dynamic environments.

Tables 5 and 6 summarize the key performance metrics for IMICLiVAN, WTCHS, ECBLTR, and EKSGA under both moderate-density (20 vehicles) and high-density (100 vehicles) scenarios. The table highlights IMICLiVAN's superior performance across all metrics. For example, in the 20-vehicle scenario, IMICLiVAN achieves a network lifetime improvement of 187% over WTCHS, 24% over ECBLTR, and 461% over EKSGA, while reducing cluster head changes by up to

Table 5 Clustering-level performance metrics under 20 and 100 vehicles

Method	Nodes	Network Lifetime	CH Changes	Cluster Changes
WTCHS	20	2.01	257	285
	100	1.41	573	649
ECBLTR	20	4.65	139	120
	100	2.26	422	392
EKSGA	20	1.03	422	470
	100	0.83	730	843
IMICLiVAN	20	5.78	86	80
	100	2.54	176	270

Table 6 Percentage change of IMICLiVAN relative to each baseline for clustering-level metrics (20 and 100 vehicles)

Baseline	Nodes	Network Lifetime	CH Changes	Cluster Changes
WTCHS	20	+187%	-66%	-72%
	100	+80%	-69%	-58%
ECBLTR	20	+24%	-38%	-33%
	100	+12%	-58%	-31%
EKSGA	20	+461%	-80%	-83%
	100	+206%	-76%	-68%

80% and cluster changes by up to 83%. Similar trends are observed in the 100-vehicle scenario, where IMICLiVAN maintains stability and efficiency even in high-density traffic conditions. These results validate IMICLiVAN's scalability and robustness, making it an effective solution for dynamic VANET environments.

The practical deployment of IMICLiVAN in real-world vehicular networks faces challenges such as limited computational resources, communication delays, and scalability in dense traffic conditions. To address computational resource limitations, IMICLiVAN employs an efficient clustering mechanism that minimizes the computational overhead of clustering processes. The clustering process is designed to operate within the constrained resources of vehicular hardware, leveraging predefined parameters to optimize performance. Communication delays, which can arise due to message propagation in dynamic environments, are minimized by ensuring that clustering decisions are made using lightweight and time-sensitive algorithms. Furthermore, IMICLiVAN's ability to adaptively adjust cluster size and re-clustering intervals enhances scalability, ensuring stable and efficient performance in both moderate- and high-density traffic scenarios. These solutions, supported by the results presented in this paper, demonstrate the feasibility of deploying IMICLiVAN in real-world vehicular networks.

IMICLiVAN addresses ethical considerations in VANET communications by prioritizing the use of essential vehicular parameters, such as mobility patterns and local distance, and minimizing the collection of sensitive data. This design choice reduces potential privacy risks associated with vehicular communications.

To further ensure the privacy and security of vehicular data, future research will explore the integration of advanced security mechanisms into IMICLiVAN. These mechanisms may include encryption techniques to secure data transmissions and anonymization methods to protect user identities. By incorporating these measures, IMICLiVAN can provide a robust and ethically responsible clustering framework for VANETs, aligning with the broader goals of intelligent transportation systems.

Integrating IMICLiVAN into real-world vehicular hardware and software platforms presents several technical challenges. One significant challenge is the limited computational resources available in onboard vehicular units, which may constrain the execution of clustering algorithms. To address this, IMICLiVAN leverages lightweight clustering mechanisms such as K-means and the silhouette score for optimized cluster formation, ensuring that the computational demands remain within the capacity of typical vehicular processors. Additionally, using a weighted formula for cluster head selection minimizes computational overhead.

Another challenge is communication delays and packet losses in vehicular networks, affecting the timely execution of clustering operations. IMICLiVAN mitigates this by optimizing the clustering process to handle inherent delays and ensure robustness in message exchange under high-mobility scenarios.

By addressing these challenges, IMICLiVAN is positioned to provide a scalable, efficient, and adaptable clustering solution for real-world vehicular environments.

4.5 Network-level communication performance metrics

We complement clustering-level metrics with network-level communication performance indicators. All metrics are collected over the full simulation interval and are computed consistently for all compared methods under an identical offered load.

4.5.1 Packet delivery ratio and packet loss ratio

Packet delivery ratio (PDR) quantifies the reliability of data dissemination by measuring the fraction of application-layer packets that are successfully delivered to the destination during the simulation. It directly reflects the robustness of the communication process under mobility, contention, and clustering dynamics.

$$\mathrm{PDR} = \frac{N_{\mathrm{recv}}}{N_{\mathrm{sent}}}. \tag{17}$$

Here, N_{sent} denotes the total number of application packets transmitted by vehicles during the simulation, and N_{recv} denotes the number of those packets that are successfully received at the destination. The packet loss ratio (PLR) complements PDR and represents the fraction of packets that fail to reach the destination due to channel loss, congestion, or topology/clustering reconfigurations:

$$\mathrm{PLR} = 1 - \mathrm{PDR}. \tag{18}$$

Fig. 13 PDR versus number of vehicles for WTCHS, ECBLTR, EKSGA, and IMICLiVAN

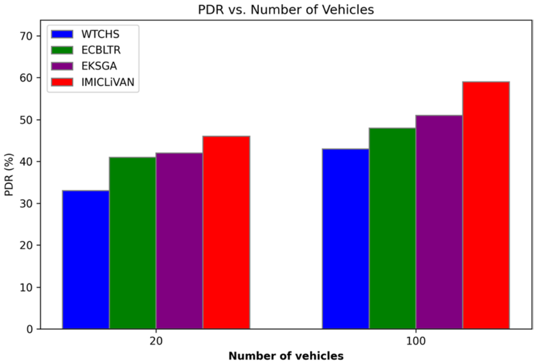


Fig. 14 PLR versus number of vehicles for WTCHS, ECBLTR, EKSGA, and IMICLiVAN

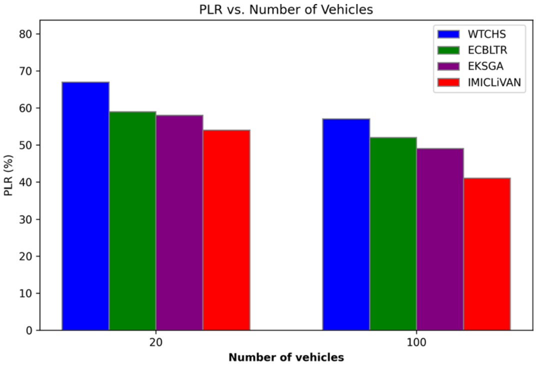


Figure 13 shows the PDR as a function of the number of vehicles for WTCHS, ECBLTR, EKSGA, and the proposed IMICLiVAN scheme, while Fig. 14 reports the corresponding PLR. For all schemes, increasing the number of vehicles from 20 to 100 improves PDR and reduces PLR. When the network is sparse (20 vehicles), packet losses are dominated by link outages and disconnected paths. As the density increases to 100 vehicles, connectivity improves and multihop routes become more stable; the additional contention is partially absorbed by the clustering layer, which limits channel access to cluster heads.

Across both densities, IMICLiVAN consistently attains the highest PDR and lowest PLR. At 20 vehicles, IMICLiVAN delivers roughly 8–10% more packets than the best baseline, and at 100 vehicles, the gain increases to around 15%. This improvement is consistent with the clustering metrics reported in Table 5, where IMICLiVAN exhibits substantially fewer cluster head changes and cluster membership changes (about 79% and 63% reductions, respectively) and a roughly threefold increase in average cluster lifetime compared to the baseline.

Moreover, IMICLiVAN selects cluster heads using a multimetric decision function that accounts for local distance, relative velocity, node degree, node center, answer ratio, trust, and distance to the base station. This favors relay nodes that are (i) centrally located within their cluster, (ii) have low relative speed with respect to their neighbors, and (iii) provide reliable responses and a good V2I link toward the base station.

Such cluster heads tend to remain valid for longer intervals and experience fewer route breaks, which further reduces packet losses along both V2V and V2I segments. The combination of more stable clusters and more reliable cluster head selection explains why IMICLiVAN achieves higher PDR and lower PLR than WTCHS, ECBLTR, and EKSGA over all tested vehicle densities.

4.5.2 Throughput

Throughput measures the delivered goodput at the receiver side, i.e., the rate at which useful payload information is successfully received over the entire simulation interval. This metric captures the combined effect of successful delivery (PDR), channel utilization, and contention caused by control overhead.

$$\text{Throughput} = \frac{8 \times B_{\text{recv}}}{T_{\text{sim}}} \text{ (bps)}, \tag{19}$$

where B_{recv} is the total number of successfully received payload bytes at the destination (base station) over the full simulation, and T_{sim} is the total simulation time in seconds. The factor 8 converts bytes into bits so that throughput is expressed in bits per second (bps); in the figures, it is reported in Kbps for readability. Figure 15 shows the throughput as a function of the number of vehicles for WTCHS, ECBLTR, EKSGA, and the proposed IMICLiVAN scheme. Increasing the number of vehicles from 20 to 100 increases the aggregate throughput for all schemes, since we have more vehicles in the road and overall connectivity improves in the denser scenario. However, IMICLiVAN consistently achieves the highest throughput at both densities. This behavior is consistent with the PDR results in Fig. 13 the proposed clustering produces fewer cluster head and membership reconfigurations, resulting in longer-lived clusters, which reduces the amount of control traffic and the duration of reconfiguration windows with elevated loss. In other word, when a vehicle remains higher amount of time as cluster head, and clusters are more stable, which means that vehicles in proposed method remain in the same clusters for longer time, it leads to a stronger communication link among the vehicles of the clusters. As a result, a larger fraction of the channel time is used for successful data delivery rather than for

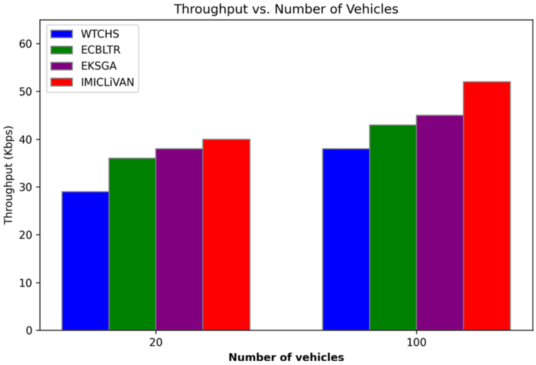


Fig. 15 Throughput versus number of vehicles for WTCHS, ECBLTR, EKSGA, and IMICLiVAN

control signaling or retransmissions, yielding higher goodput compared to WTCHS, ECBLTR, and EKSGA.

4.5.3 End-to-end delay

The average end-to-end delay characterizes timeliness by measuring how long it takes for an application packet to travel from its source vehicle to the destination. In VANETs, this metric is critical for safety-related dissemination, where excessive latency can reduce the usefulness of messages. The average delay is computed over successfully delivered packets only:

$$\overline{D} = \frac{1}{N_{\text{recv}}} \sum_{i=1}^{N_{\text{recv}}} D_i, \tag{20}$$

where each D_i includes propagation, transmission time, contention and queuing, and processing delay and N_{recv} is the number of successfully delivered reports. Figure 16 depicts the average end-to-end delay as a function of the number of vehicles. For all schemes, the delay increases when the number of vehicles grows from 20 to 100, since higher density leads to more channel contention and longer queues at relay nodes. However, IMICLiVAN exhibits the lowest delay in both scenarios. This behavior is consistent with its clustering dynamics: IMICLiVAN generates significantly fewer cluster head and membership changes and thus fewer reconfiguration windows, during which our simulator imposes additional loss probability and extra queuing delay on data transmissions. Furthermore, the multimetric cluster head selection considering relative speed, local position, node degree, node center, answer ratio, trust, and distance to the base station tends to choose more stable and better-connected relays, which reduces route breaks and buffering. Besides, When clustering becomes unstable, inefficient cluster segregation leads to frequent inter-cluster hopping. These frequent hopping events disrupt communication paths and reduce the effective data transmission between vehicles. Moreover, repeated inter-cluster hopping increases the frequency of cluster head changes, which in turn raises packet delay and prolongs the end-to-end transmission time. To address this

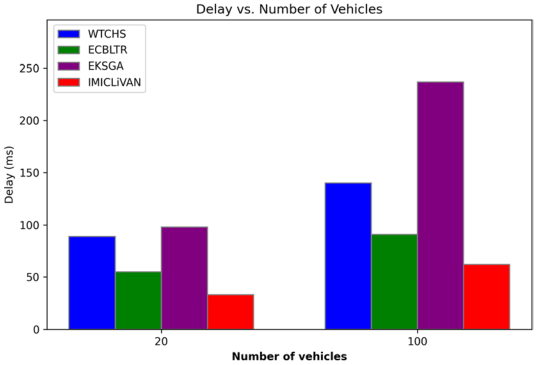


Fig. 16 Average delay versus number of vehicles for WTCHS, ECBLTR, EKSGA, and IMICLiVAN

issue, we employ the silhouette score to select clusters with higher segregation efficiency, thereby reducing inter-cluster hopping, stabilizing cluster head selection, and improving overall transmission performance. As a result, packets traverse fewer disrupted paths and experience shorter end-to-end delays compared to WTCHS, ECBLTR, and EKSGA.

4.5.4 Jitter

Jitter captures delay variability, which is important for assessing the stability of packet delivery timing. Even when the mean delay is moderate, high jitter indicates irregular delivery intervals, which can degrade application performance. We compute jitter as the mean absolute difference between the delays of consecutive successfully delivered packets:

$$\bar{J} = \frac{1}{N_{\text{recv}} - 1} \sum_{i=2}^{N_{\text{recv}}} \left| D_i - D_{i-1} \right|, \tag{21}$$

where D_i and D_{i-1} denote the delays of two consecutive successfully delivered packets, and N_{recv} is the number of successfully received packets. The resulting $\bar{J}$ is reported in seconds. Figure 17 illustrates the average jitter as a function of the number of vehicles for WTCHS, ECBLTR, EKSGA, and the proposed IMICLiVAN scheme. For all schemes, jitter tends to increase when the number of vehicles grows from 20 to 100 due to higher channel contention and more frequent queue build-up at relay nodes. However, IMICLiVAN consistently achieves the lowest jitter across both densities, indicating more regular packet delivery. This reduction in jitter is mainly due to the clustering and cluster head selection mechanisms. First, IMICLiVAN selects the number of clusters adaptively using the silhouette score, choosing at each simulation step the value of k that yields compact and well-separated clusters. Such clusters exhibit more stable memberships and shorter, more homogeneous intra-cluster paths, which reduce abrupt changes in queuing and transmission delays. Second, cluster heads are chosen using a weighted multimetric function that considers local distance, relative velocity, node degree, node centrality, answer ratio,

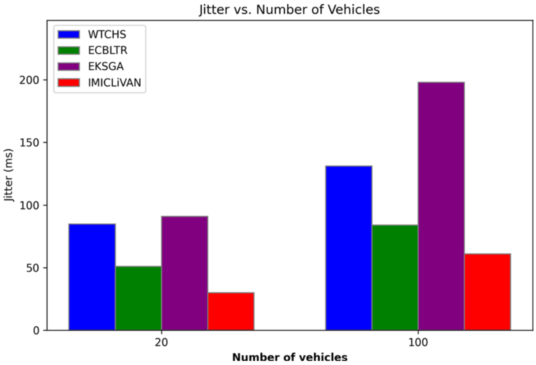


Fig. 17 Jitter versus number of vehicles for WTCHS, ECBLTR, EKSGA, and IMICLiVAN

distance to the base station, and trust, with trust assigned the largest weight. This favors stable and reliable relay vehicles that remain in the cluster for longer periods and maintain good connectivity to both neighbors and the base station. As a result, the routing paths remain consistent over time and packets experience fewer delay spikes caused by transient reconfigurations, leading to lower jitter for IMICLiVAN compared to WTCHS, ECBLTR, and EKSGA.

4.5.5 Overhead

We report the overhead in terms of transmitted control bytes per second:

$$\text{Overhead}_{\mathrm{B/s}} = \frac{B_{\mathrm{ctrl}}}{T_{\mathrm{sim}}}\ \text{(Bps)}. \tag{22}$$

where B_{ctrl} denotes the total number of transmitted control bytes accumulated over the whole simulation interval, and T_{sim} denotes the total simulation time in seconds. Figure 18 shows the control overhead as a function of the number of vehicles for WTCHS, ECBLTR, EKSGA, and the proposed IMICLiVAN scheme. Increasing the number of vehicles from 20 to 100 naturally increases the overhead for all schemes, since more vehicles participate in cluster formation, maintenance, and neighbor discovery. IMICLiVAN, however, consistently incurs the lowest overhead. This result is directly related to how clusters and cluster heads are formed and updated in our framework. First, IMICLiVAN selects the number of clusters adaptively using the silhouette score, choosing at each simulation step the value of k that yields compact and well-separated clusters. Such clusters exhibit more stable memberships and avoid both overly fragmented and excessively large clusters. In our simulator, every cluster head change or membership change triggers a reconfiguration window in which additional control messages (HELLO, JOIN/LEAVE, TABLE) are exchanged. By reducing the frequency of cluster head changes and cluster reassignments compared to the predefined schemes, IMICLiVAN naturally generates fewer reconfiguration events and therefore fewer bursts of control traffic. Second, cluster heads are chosen using a weighted multimetric decision function that combines local distance, relative velocity, node degree, node center, answer ratio, distance to the base station, and trust, with trust assigned the highest weight. This leads to the

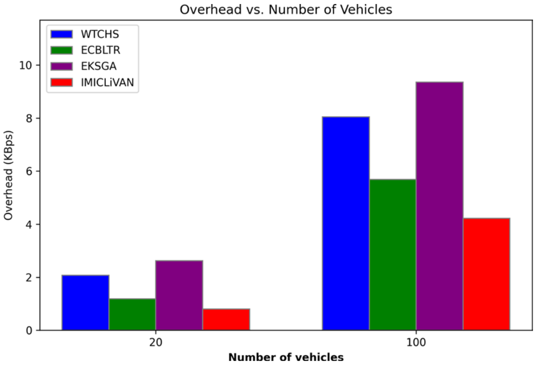


Fig. 18 Overhead versus number of vehicles for WTCHS, ECBLTR, EKSGA, and IMICLiVAN

selection of stable and reliable relay vehicles that tend to remain cluster heads for longer periods. As a consequence, the cluster head role does not oscillate frequently among vehicles, which further limits the amount of control signaling required for cluster maintenance. Overall, the combination of silhouette-based adaptive clustering and multimetric, trust-aware cluster head selection reduces unnecessary cluster reconfigurations and yields a lower control overhead for IMICLiVAN than for WTCHS, ECBLTR, and EKSGA.

4.5.6 Energy efficiency

Since SUMO does not natively model radio energy consumption, we implement an analytical state-based energy model on top of the communication layer. Each vehicle alternates between transmit (TX), receive (RX), and idle/listen states with constant power levels P_{tx}, P_{rx}, and P_{idle}, as summarized in Table 4. Let T_{tx}, T_{rx}, and T_{idle} denote the cumulative time spent in each state over the whole simulation. The total radio energy consumed by all vehicles is then:

$$E_{\mathrm{total}} = P_{\mathrm{tx}} T_{\mathrm{tx}} + P_{\mathrm{rx}} T_{\mathrm{rx}} + P_{\mathrm{idle}} T_{\mathrm{idle}}. \tag{23}$$

Here, T_{tx} and T_{rx} are obtained from the total transmitted and received bytes (data + control) using the PHY data rate R_{data}, while T_{idle} accounts for the remaining radio-on time when nodes are listening but not actively transmitting or receiving. Let $B_{\mathrm{payload,rx}}$ denote the total number of application payload bytes successfully delivered at the base station. We define the network-wide energy efficiency as the number of useful bits delivered per Joule of radio energy:

$$\eta_E = \frac{8\,B_{\mathrm{payload,rx}}}{E_{\mathrm{total}}} \quad [\mathrm{bits/J}]. \tag{24}$$

Larger values of η_E therefore indicate higher energy efficiency; in the results, η_E is reported in Kbits/J for readability. Figure 19 compares the energy efficiency of WTCHS, ECBLTR, EKSGA, and the proposed IMICLiVAN scheme for 20 and 100 vehicles. IMICLiVAN consistently achieves the highest η_E in both scenarios.

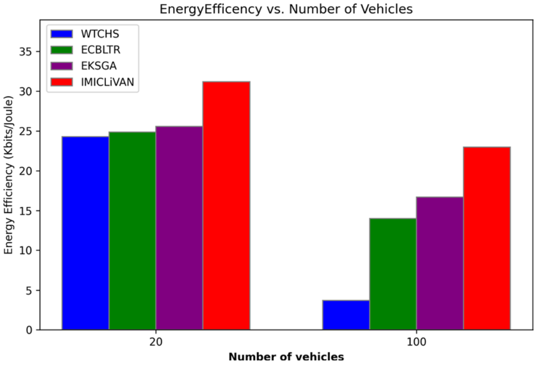


Fig. 19 Energy efficiency versus number of vehicles for WTCHS, ECBLTR, EKSGA, and IMICLiVAN

This improvement stems from two effects that are explicit in our simulator. First, IMICLiVAN produces substantially fewer cluster head and membership changes and longer cluster lifetimes, which reduces the number of reconfiguration windows. Each reconfiguration event triggers additional control messages and a period with increased drop probability and extra delay; consequently, schemes with higher churn waste more radio energy on control signaling and unsuccessful transmissions. Second, IMICLiVAN attains a higher PDR (cf. Figure 14), so a larger fraction of the consumed energy actually contributes to successful payload delivery at the base station. The combination of lower control/overhead energy and higher delivered payload directly translates into higher energy efficiency compared to WTCHS, ECBLTR, and EKSGA.

Using the above definitions and per-metric plots, we now provide a compact numerical comparison of all schemes. Table 7 reports, for both traffic densities (20 and 100 vehicles), the absolute values of the main network-level metrics obtained by WTCHS, ECBLTR, EKSGA, and the proposed IMICLiVAN under the same offered load and communication settings. The results confirm the trends observed in the figures: IMICLiVAN achieves higher throughput and PDR, lower PLR, and significantly reduced jitter and end-to-end delay, while at the same time improving energy efficiency and reducing control overhead.

5 Conclusions and future works

This paper introduced IMICLiVAN, a novel clustering mechanism designed to address the dynamic clustering challenges in VANETs. The core contributions include an adaptive clustering process using the K-means algorithm, guided by the silhouette score, to optimize cluster count in real time. The paper also proposed a robust cluster head selection protocol that leverages several metrics such as local distance, relative speed, and base station distance. These innovations enhance cluster stability and communication reliability in highly mobile vehicular environments.

IMICLiVAN was validated using SUMO integrated with Python's Traci library for real-time vehicle control, enabling dynamic clustering and cluster head selection.

Table 7 Network-level communication metrics under 20 and 100 vehicles

Method	Nodes	Throughput	PDR	PLR	Jitter	Delay	Energy Eff	Overhead
WTCHS	20	29 Kbps	33%	67%	85 ms	89 ms	24.3 Kb/J	2.076 KBps
	100	38 Kbps	43%	57%	131 ms	140 ms	3.7 Kb/J	8.049 KBps
ECBLTR	20	36 Kbps	41%	59%	51 ms	55 ms	24.9 Kb/J	1.187 KBps
	100	43 Kbps	48%	52%	84 ms	91 ms	14 Kb/J	5.682 KBps
EKSGA	20	37 Kbps	42%	58%	90 ms	98 ms	25.6 Kb/J	2.627 KBps
	100	45 Kbps	51%	49%	198 ms	237 ms	16.7 Kb/J	9.36 KBps
IMICLiVAN	20	40 Kbps	46%	54%	30 ms	33 ms	31.2 Kb/J	0.821 KBps
	100	52 Kbps	59%	41%	61 ms	62 ms	23 Kb/J	4.23 KBps

The simulation results demonstrate that IMICLiVAN outperforms the best existing methods (WTCHS, ECBLTR, and EKSGA) regarding network lifetime, cluster head changes, and cluster changes. Combining the K-means algorithm, silhouette score, and weighted formula for cluster head selection results in more stable clusters while exhibiting fewer re-clustering events and longer network lifetimes. This makes the IMICLiVAN method highly suitable for dynamic VANET environments, where minimizing overhead and maintaining stable communication links are critical.

To better quantify these gains, Table 8 summarizes the percentage change of IMICLiVAN relative to each baseline. In the dense scenario (100 vehicles), IMICLiVAN provides up to roughly 15% to 37% higher throughput and PDR, more than 70% reduction in delay and jitter, substantial energy efficiency improvements (over a fivefold increase with respect to WTCHS), and overhead reductions of about 25% to 55%. These aggregated results highlight the consistent benefits of the proposed adaptive, trust-aware clustering over the considered baseline schemes across all evaluated operating points.

From a deployment perspective, the above results imply that IMICLiVAN can: (i) significantly reduce the frequency of cluster head and cluster reconfigurations, thereby lowering control-plane signaling and freeing channel resources for data traffic; (ii) improve the timeliness and reliability of safety and traffic information messages by reducing packet loss, delay, and delay variability; and (iii) lower the communication-related energy cost per delivered packet, which is particularly relevant for energy-constrained or battery-assisted vehicular platforms. In moderate-density scenarios, the algorithm is lightweight enough to be executed entirely on board. For dense or city-scale deployments under strict latency constraints, the method is also amenable to parallel and distributed execution (e.g., on RSU/MEC or cloud/HPC backends) thanks to the natural parallelism of candidate-K evaluation, distance/silhouette computations, and per-cluster CH scoring.

The present study has several limitations that should be acknowledged. First, the simulations are conducted on a single bidirectional multilane road segment with one base station, which does not capture all the complexities of large, heterogeneous urban road networks (intersections, multiple BSs/RSUs, mixed mobility patterns). Second, from a functional standpoint, the current design focuses on a hand-crafted, fixed-weight multimetric CH scoring rule and does not yet exploit data-driven or AI-based optimization to adapt the clustering parameters

Table 8 Percentage change of IMICLiVAN relative to each baseline for 20 and 100 vehicles

Baseline	Nodes	Throughput	PDR	PLR	Jitter	Delay	Energy Eff	Overhead
WTCHS	20	+37.93%	+39.39%	−19.4%	−64.71%	−62.92%	+28.4%	−60.46%
	100	+36.84%	+37.21%	−28.07%	−53.44%	−55.71%	+521.62%	−47.45%
ECBLTR	20	+11.11%	+12.2%	−8.47%	−41.18%	-40%	+25.3%	−30.83%
	100	+20.93%	+22.92%	−21.15%	−27.38%	−31.87%	+64.29%	−25.55%
EKSGA	20	+8.11%	+9.52%	−6.9%	−66.66%	−66.33%	+21.88%	−68.74%
	100	+15.56%	+15.69%	−16.33%	−69.19%	−73.84%	+37.72%	−54.81%

(e.g., weights, cluster size, or re-clustering intervals) to the observed traffic and network conditions. And third, it can include more detailed security or attack models.

Future research will also focus on enhancing IMICLiVAN by integrating AI-based optimization techniques and exploring its application in intelligent routing. Machine learning models can be employed to predict vehicle mobility patterns, enabling more efficient cluster head selection and proactive cluster reorganization. Reinforcement learning algorithms offer the potential to dynamically optimize clustering parameters, such as cluster size and re-clustering intervals, based on real-time traffic and network conditions, further improving cluster stability and scalability. Additionally, extending IMICLiVAN's framework to support intelligent routing decisions can leverage its clustering architecture to establish stable and efficient communication paths. By utilizing cluster heads as relay points and optimizing route selection through real-time traffic and network conditions, IMICLiVAN can further reduce latency and enhance communication reliability. These advancements will enable the integration of IMICLiVAN into next-generation VANET systems, offering robust solutions for dynamic and complex vehicular environments.

Acknowledgements The authors would like to acknowledge that no individuals or organizations contributed to this work.

Author contributions E.A. and E.M. contributed equally to the work presented in this manuscript. They were involved in all aspects of the research, including the design, methodology, data collection, analysis, and writing of the manuscript. S.A.A and M.B.M and Y.S were responsible for supervising, reviewing, and editing the manuscript.

Funding The authors declare that no funding was received for this work.

Data availability No datasets were generated or analyzed during the current study.

Declarations

Conflict of interest The authors declare no Conflict of interest.

References

1. Khan MF, Aadil F, Maqsood M, Bukhari SHR, Hussain M, Nam Y (2019) Moth flame clustering algorithm for internet of vehicle (mfca-iov). IEEE Access 7:11613–11629. https://doi.org/10.1109/ACCESS.2018.2886420
2. Alsarhan A, Kilani Y, Al-Dubai A, Zomaya AY, Hussain A (2020) Novel fuzzy and game theory based clustering and decision making for vanets. IEEE Trans Veh Technol 69(2):1568–1581. https://doi.org/10.1109/TVT.2019.2956228
3. Mahi MJN, Chaki S, Ahmed S, Biswas M, Kaiser MS, Islam MS (2022) A review on vanet research: perspective of recent emerging technologies. IEEE Access 10:65760–65783. https://doi.org/10.1109/ACCESS.2022.3183605
4. Kandali K, Bennis L, Bannay OE, Bennis H (2022) An intelligent machine learning based routing scheme for vanet. IEEE Access 10:74318–74333. https://doi.org/10.1109/ACCESS.2022.3190964
5. Badole MH, Thakare AD (2024) Cluster-based multicast optimized routing in vanets using elite knowledge-based genetic algorithm. Knowl Based Syst 294:111773. https://doi.org/10.1016/J.KNOSYS.2024.111773

6. Sahoo RR, Panda R, Behera DK, Naskar MK (2012) A trust based clustering with ant colony routing in vanet. In: 2012 3rd International Conference on Computing, Communication and Networking Technologies . https://doi.org/10.1109/ICCCNT.2012.6395939
7. Al-Shareeda MA, Manickam S (2023) A systematic literature review on security of vehicular adhoc network (vanet) based on veins framework. IEEE Access 11:46218–46228. https://doi.org/10.1109/ACCESS.2023.3274774
8. Ramlee SRB, Hasan S, Subramaniam SK (2023) Cluster optimization in vanet using mfo algorithm and k-means clustering. In: 2023 13th International Conference on Information Technology in Asia, CITA 2023, pp. 70–75 . https://doi.org/10.1109/CITA58204.2023.10262579
9. Nithyanandam G, Ambiyaram C, Prabathkumar S (2023) An intelligent hybrid prairie dog optimization algorithm-based stable cluster reliable routing scheme for vanets. Int J Commun Syst 36(14):5549. https://doi.org/10.1002/dac.5549
10. Amari H, Houda ZAE, Khoukhi L, Belguith LH (2023) Trust management in vehicular ad-hoc networks: extensive survey. IEEE Access 11:47659–47680. https://doi.org/10.1109/ACCESS.2023.3268991
11. Naeem A, Rizwan M, Alsubai S, Almadhor A, Akhtaruzzaman M, Islam S, Rahman H (2023) Enhanced clustering based routing protocol in vehicular ad-hoc networks. IET Electrical Systems in Transportation **13**(1) https://doi.org/10.1049/els2.12069
12. Jabbar M, Trabelsi H (2022) A betweenness centrality based clustering in vanets. In: 2022 15th International Conference on Security of Information And . https://doi.org/10.1109/SIN56466.2022.9970553
13. Amirshahi A, Romoozi M, Raayatpanah MA, Asghari SA (2020) Modeling geographical anycasting routing in vehicular networks. KSII Transactions Internet Information Syst (TIIS) 14(4):1624–1647. https://doi.org/10.3837/TIIS.2020.04.012
14. Badole M, Thakare A, Oliva D (2024) An improved multi-objective optimization framework with a hybrid model for vehicular adhoc network routing services. Wireless Pers Commun. https://doi.org/10.1007/s11277-024-11245-6
15. Bijalwan A, Hussain I, Purohit KC, Kumar MA (2023) Enhanced ant colony optimization for vehicular ad hoc networks using fittest node clustering. Sustainability **15**(22) https://doi.org/10.3390/SU152215903
16. Dua A, Sharma P, Ganju S, Gagangeet Singh Aujla AJ, Kumar N (2018) Rovan: A rough set-based scheme for cluster head selection in vehicular ad-hoc networks. In: 2018 IEEE Global Communications Conference (GLOBECOM) . https://doi.org/10.1109/GLOCOM.2018.8647576
17. Vergis S, Tsoumanis G, Oikonomou K (2021) A proposed multi-head clustering algorithm for vanet environments. In: 2021 6th South-East Europe Design Automation, Computer Engineering . https://doi.org/10.1109/SEEDA-CECNSM53056.2021.9566263
18. Mirsadeghi F, Rafsanjani MK, Gupta BB (2021) A trust infrastructure based authentication method for clustered vehicular ad hoc networks. Peer-to-Peer Networking Appl 14(4):2537–2553. https://doi.org/10.1007/S12083-020-01010-4
19. Hadded M, Zagrouba R, Laouiti A, Muhlethaler P, Saidane LA (2015) A multi-objective genetic algorithm-based adaptive weighted clustering protocol in vanet. In: 2015 IEEE Congress on Evolutionary Computation, CEC 2015 - Proceedings, pp. 994–1002 . https://doi.org/10.1109/CEC.2015.7256998
20. Alsuhli GH, Khattab A, Fahmy YA (2018) A mobility-based double-head clustering algorithm for dynamic vanet. In: 2018 Proceedings of the Japan-Africa Conference on Electronics, Communications, and Computations, JAC-ECC 2018, pp. 91–94 . https://doi.org/10.1109/JEC-ECC.2018.8679569
21. Amirshahi A, Romoozi M, Raayatpanah MA, Asghari SA (2020) Anycast routing in time-expanded vehicular networks. Computers & Electrical Engineering **82** https://doi.org/10.1016/J.COMPELECENG.2020.106560
22. Khayat G, Mavromoustakis CX, Mastorakis G, Batalla JM, Maalouf H, Pallis E (2020) Vanet clustering based on weighted trusted cluster head selection. International Wireless Communications and Mobile Computing (IWCMC) https://doi.org/10.1109/IWCMC48107.2020.9148339 . Accessed: Jul. 31, 2024
23. Alam I, Manjul M, Pathak V, Mala V, Mangal A, Thakur HK, Sharma DK (2024) Efficient and secure graph-based trust-enabled routing in vehicular ad-hoc networks. Mobile Networks and Applications, 1–21 https://doi.org/10.1007/s11036-023-02274-9

24. Blessy AMC, Brindha S (2024) Maximizing vanet performance in cluster head selection using intelligent fuzzy bald eagle optimization. Elsevier https://doi.org/10.1016/j.vehcom.2023.100660 . Accessed: Aug. 22, 2024
25. Aissa M, Bouhdid B, Mnaouer AB (2021) Enhanced fuzzy logic-based cluster stability in vehicular ad hoc network. In: 2021 International Symposium on Networks, Computers and Communications, ISNCC 2021 . https://doi.org/10.1109/ISNCC52172.2021.9615872
26. Singh GD, Prateek M, Kumar S, Verma M, Singh D, Lee HN (2022) Hybrid genetic firefly algorithm-based routing protocol for vanets. IEEE Access 10:9142–9151. https://doi.org/10.1109/ACCESS.2022.3142811
27. Ruban CB, Paramasivan B (2020) Vtd protocol: A cluster based trust model for secure communication in vanet. International Journal of Intelligent Engineering and Systems **13**(1) https://doi.org/10.22266/ijies2020.0229.04
28. Abadi AFE, Asghari SE, Sharifani S, Asghari SA, Marvasti MB (2022) A survey on utilizing reinforcement learning in wireless sensor networks routing protocols. In: Proceedings of the 13th International Conference on Information and Knowledge Technology, IKT 2022 . https://doi.org/10.1109/IKT57960.2022.10039013
29. Abadi AFE, Asghari SA, Marvasti MB, Abaei G, Nabavi M, Savaria Y (2022) Rlbeep: reinforcement-learning-based energy efficient control and routing protocol for wireless sensor networks. IEEE Access 10:44123–44135. https://doi.org/10.1109/ACCESS.2022.3167058
30. Awan KA, Din IU, Almogren A, Guizani M, Khan S (2020) Stabtrust-a stable and centralized trust-based clustering mechanism for iot enabled vehicular ad-hoc networks. IEEE Access 8:21159–21177. https://doi.org/10.1109/ACCESS.2020.2968948
31. Gayathri M, Gomathy C (2022) Fuzzy based trusted communication in vehicular ad hoc network. In: 2022 2nd International Conference on Intelligent Technologies, CONIT 2022 . https://doi.org/10.1109/CONIT55038.2022.9847823
32. Saleem MA, Shijie Z, Sarwar MU, Ahmad T, Maqbool A, Shivachi CS, Tariq M (2021). Deep Learning-Based Dynamic Stable Cluster Head Selection Vanet. https://doi.org/10.1155/2021/9936299
33. Babu D, Rao TS (2024) A secure routing protocol using trust-based clustering and bionic intelligence algorithm for uav-assisted vehicular ad hoc networks. Transactions Emerging Telecommunications Technol 35(5):4977. https://doi.org/10.1002/ett.4977
34. Malakreddy AB, Rajesh IS, G, MH, N, RU, Krishnamurthy MS, C M (2024) Flql-vanet: a hybrid of fuzzy logic and q-learning schemes for qos aware routing in vanet. International Journal of Intelligent Engineering and Systems 17(6):1268–1280. https://doi.org/10.22266/ijies2024.1231.92
35. Balaji TS, Srinivasan S, Bharathi SP, Ramesh B (2022) Fuzzy-based secure clustering with routing technique for vanets. Comput Syst Sci Eng 43(1):291–304. https://doi.org/10.32604/csse.2022.023269
36. Abdulrazzak HN, Hock GC, Radzi NAM, Tan NML, Kwong CF (2022) Modeling and analysis of new hybrid clustering technique for vehicular ad hoc network. Math 10(24):4720. https://doi.org/10.3390/math10244720
37. Alsuhli GH, Khattab A, Fahmy YA (2019) An evolutionary approach for optimized vanet clustering. In: Proceedings of the International Conference on Microelectronics, ICM, pp. 70–73 . https://doi.org/10.1109/ICM48031.2019.9021941
38. Khayat G, Mavromoustakis CX, Mastorakis G, Batalla JM, Maalouf H, Mumtaz S, Pallis E (2020) Successful delivery in vanets with damaged infrastructures based on double cluster head selection, 1–5 https://doi.org/10.1109/CAMAD50429.2020.9209259
39. Siddiqa A, Diyan M, Khan MTR, Saad MM, Kim D (2021) Mitigating broadcasting storm using multihead nomination clustering in vehicular content centric networks. Electronics 10(18):2270. https://doi.org/10.3390/electronics10182270
40. Aditya VMVS, Dhuli S, Sashrika PL, Shivani KK, Jayanth T (2020) Closeness centrality based cluster head selection algorithm for large scale wsns. In: 2020 12th International Conference on Computational Intelligence and Communication Networks, CICN 2020, pp. 107–111 . https://doi.org/10.1109/CICN49253.2020.9242639
41. Choudhary D, Yadav D, Joshi J, Deka M, Jha S (2016) A model based connectivity and localization strategy for vehicular ad hoc networks. In: International Conference on Intelligent Systems, Modelling and Simulation, ISMS, pp. 331–336 . https://doi.org/10.1109/ISMS.2016.17
42. Jabbar MK, Trabelsi H (2022) A novelty of hypergraph clustering model (hgcm) for urban scenario in vanet. IEEE Access **10** https://doi.org/10.1109/ACCESS.2022.3185075

43. Moura DLL, Ramos GS, Aquino ALL, Loureiro AAF (2023) A centrality approach to select offloading data aggregation points in vehicular sensor networks. IEEE Trans Intell Transp Syst 24:11639–11653. https://doi.org/10.1109/TITS.2023.3297886
44. Hajlaoui R, Alsolami E, Moulahi T, Guyennet H (2019) Construction of a stable vehicular ad hoc network based on hybrid genetic algorithm. Telecommunications Syst 71(3):433–445. https://doi.org/10.1007/s11235-018-0513-6
45. Charoenchai S, Siripongwutikorn P (2023) Genetic algorithm for multi-hop vanet clustering based on coalitional game. Journal of Network and Systems Management **32** https://doi.org/10.1007/s10922-023-09787-4
46. Madasamy B, Balasubramaniam P (2022) Enhanced load balanced clustering technique for vanet using location aware genetic algorithm. Promet - Traffic & Transportation 34(1):39–52. https://doi.org/10.7307/ptt.v34i1.3785
47. Begum S, Patil NB (2023). An effective cluster based routing using multi-objective prairie dog optimization in vehicular ad-hoc network. https://doi.org/10.1109/ICDSNS58469.2023.10245778
48. Hijjawi M, Alshinwan M, Khashan OA, Alomoush W, Karim NA, Shdefat AY (2023) A novel hybrid prairie dog algorithm and harris hawks algorithm for resource allocation of wireless networks. IEEE Access **11** https://doi.org/10.1109/ACCESS.2023.3335247
49. Husnain G, Anwar S (2021) An intelligent cluster optimization algorithm based on whale optimization algorithm for vanets (woacnet). PLoS ONE 16(4):0250271. https://doi.org/10.1371/journal.pone.0250271
50. Venkatasubramanian S (2024). Improved beluga whale optimizer-based cluster head selection in fanet. https://doi.org/10.1109/ICMNWC60182.2023.10435833
51. Shah YA, Habib HA, Aadil F, Khan MF, Maqsood M, Nawaz T (2018) Camonet: Moth-flame optimization (mfo) based clustering algorithm for vanets. IEEE Access 6:48611–48624. https://doi.org/10.1109/ACCESS.2018.2868118
52. Tariq R, Iqbal Z, Aadil F (2020) Imoc: optimization technique for drone-assisted vanet (dav) based on moth flame optimization. Wirel Commun Mob Comput 2020:1–29. https://doi.org/10.1155/2020/8860646
53. Sadrishojaei M, Navimipour NJ, Reshadi M, Hosseinzadeh M (2021) Clustered routing method in the internet of things using a moth-flame optimization algorithm. International Journal of Communication Systems **34**(16) https://doi.org/10.1002/dac.4964
54. Ramamoorthy R, Thangavelu M (2020) An improved distance-based ant colony optimization routing for vehicular ad hoc networks. International Journal of Communication Systems **33**(14) https://doi.org/10.1002/dac.4502
55. Khan Z, Fang S, Koubaa A, Fan P, Abbas F, Farman H (2020) Street-centric routing scheme using ant colony optimization-based clustering for bus-based vehicular ad-hoc network. Comput Electrical Eng 86:106736. https://doi.org/10.1016/j.compeleceng.2020.106736
56. Shivshankar S, Jamalipour A (2015) An evolutionary game theory-based approach to cooperation in vanets under different network conditions. IEEE Trans Veh Technol 64(5):2015–2022. https://doi.org/10.1109/TVT.2014.2334655
57. Meera VK, Balasubramanian C (2024) A hybrid fennec fox and sand cat optimization algorithm for clustering scheme in vanets. Sustainable Computing: Informatics and Systems **42** https://doi.org/10.1016/J.SUSCOM.2024.100983
58. Jahangeer N, Mustafa Z, Aldabbas H (2024) Utilizing a hybrid cat-swarm optimization algorithm for seamless vertical handoff in vanet. Wireless Pers Commun 137(1):495–521. https://doi.org/10.1007/s11277-024-11424-5
59. Badole MH, Thakare AD (2023) An optimized framework for vanet routing: a multi-objective hybrid model for data synchronization with digital twin. Inter J Intelligent Networks 4:272–282. https://doi.org/10.1016/j.ijin.2023.10.001
60. Lopez PA, Behrisch M, Bieker-Walz L, Erdmann J, Flötteröd Y-P, Hilbrich R, Lücken L, Rummel J, Wagner P, Wiessner E 2018) Microscopic traffic simulation using sumo, 2575–2582 (https://doi.org/10.1109/ITSC.2018.8569938
61. Erdmann J, Wagner P (2025) Sumo's interpretation of the krauß model. Simulation Communication Papers https://doi.org/10.52825/scp.v6i.2638
62. Al-Kharasani NM, Zulkarnain ZA, Subramaniam S, Hanapi ZM. An efficient framework model for optimizing routing performance in vanets **18** https://doi.org/10.3390/s18020597
63. Garcia N, Wymeersch H, Wen F, Do H, Boban M, Kakkavas A, Wang Q, Destino G, Mahmoodi T, Mandelli S, Saur S, Wesemann S, Phan-Huy D-T, Katranaras E, Abbas T. Report on channel

modelling and positioning for 5g v2x. Project Deliverable D3.2, 5GCAR (Fifth Generation Communication Automotive Research and Innovation). Horizon 2020 Grant Agreement No. 761510. https://5gcar.eu/wp-content/uploads/2019/02/5GCAR_D3.2_v1.0.pdf Accessed 2026-01-05

64. European Telecommunications Standards Institute (ETSI): Intelligent transport systems (its); radiocommunications equipment operating in the 5 855 mhz to 5 925 mhz frequency band; harmonized en covering the essential requirements of article 3.2 of the r&tte directive. Harmonized European Standard ETSI EN 302 571 V1.1.1, ETSI. Reference: DEN/ERM-TG37-007. https://www.etsi.org/deliver/etsi_en/302500_302599/302571/01.01.01_60/en_302571v010101p.pdf Accessed 2026-01-05

Authors and Affiliations

Erfan Moghadam[1] · Elham Asghari[2] · Seyed Amir Asghari[3] · Mohammadreza Binesh Marvasti[1] · Yvon Savaria[2]

✉ Seyed Amir Asghari
s.a.asghari@ut.ac.ir

[1] Department of Electrical and Computer Engineering, Kharazmi University, Tehran, Iran

[2] Department of Electrical and Computer Engineering, Polytechnique Montreal, Montreal, QC, Canada

[3] Data Science and Technology Department, University of Tehran, Tehran, Iran