

Shailender Kumar

ASHISH KUMARI_FINAL_THESIS 21JULY

 paper

Document Details

Submission ID**trn:oid:::27535:146386898****Submission Date****Jul 24, 2026, 5:18 PM GMT+5:30****Download Date****Jul 24, 2026, 5:27 PM GMT+5:30****File Name****ASHISH KUMARI_FINAL_THESIS 21JULY.docx****File Size****11.4 MB****162 Pages****41,847 Words****251,946 Characters**

5% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- Bibliography
- Quoted Text
- Small Matches (less than 10 words)

Exclusions

- 4 Excluded Sources

Match Groups

-  **143 Not Cited or Quoted 5%**
Matches with neither in-text citation nor quotation marks
-  **20 Missing Quotations 1%**
Matches that are still very similar to source material
-  **0 Missing Citation 0%**
Matches that have quotation marks, but no in-text citation
-  **0 Cited and Quoted 0%**
Matches with in-text citation present, but no quotation marks

Top Sources

- 3%  Internet sources
- 3%  Publications
- 3%  Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

Match Groups

- 143 Not Cited or Quoted 5%**
Matches with neither in-text citation nor quotation marks
- 20 Missing Quotations 1%**
Matches that are still very similar to source material
- 0 Missing Citation 0%**
Matches that have quotation marks, but no in-text citation
- 0 Cited and Quoted 0%**
Matches with in-text citation present, but no quotation marks

Top Sources

- 3% Internet sources
- 3% Publications
- 3% Submitted works (Student Papers)

Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	Internet	arxiv.org	<1%
2	Student papers	Delhi Technological University on 2026-05-20	<1%
3	Student papers	Delhi Technological University on 2026-06-09	<1%
4	Student papers	Asia Pacific Institute of Information Technology on 2026-02-13	<1%
5	Publication	"Data Science and Applications", Springer Science and Business Media LLC, 2026	<1%
6	Publication	Parveen Kumar, Pawan Kumar Dahiya, Bijay Kumar Singh. " Modularized hypergr...	<1%
7	Student papers	Delhi Technological University on 2026-04-30	<1%
8	Internet	repository.ju.edu.et	<1%
9	Internet	systems.uomisan.edu.iq	<1%
10	Internet	dspace.dtu.ac.in:8080	<1%

11	Publication	Kajol Mittal, Payal Khurana Batra. "Graph-ensemble fusion for enhanced IoT intru...	<1%
12	Internet	ebin.pub	<1%
13	Student papers	Centre for Technical Support on 2025-07-25	<1%
14	Internet	www.theses.fr	<1%
15	Student papers	Foreign Trade University on 2025-06-09	<1%
16	Internet	eprints.utm.my	<1%
17	Student papers	Afe Babalola University on 2026-06-16	<1%
18	Student papers	National Institute of Technology, Patna on 2025-02-17	<1%
19	Student papers	University of Surrey on 2017-05-23	<1%
20	Internet	eprints.lancs.ac.uk	<1%
21	Student papers	Delhi Technological University on 2026-05-06	<1%
22	Publication	Liam Daly Manocchio, Siamak Layeghy, Marius Portmann. "Network Intrusion De...	<1%
23	Publication	Yanqing Yang, Kangfeng Zheng, Chunhua Wu, Xinxin Niu, Yixian Yang. "Building ...	<1%
24	Internet	etheses.whiterose.ac.uk	<1%

25	Student papers	University of Pretoria on 2013-10-19	<1%
26	Internet	www.researchgate.net	<1%
27	Publication	Divya Ramachandran, Mubarak Albathan, Ayyaz Hussain, Qaisar Abbas. "Enhanci...	<1%
28	Student papers	The University of Manchester on 2026-04-26	<1%
29	Student papers	National Institute of Technology, Rourkela on 2026-07-24	<1%
30	Internet	mdpi-res.com	<1%
31	Publication	Anish Shandilya, Varuna Gupta, Bosco Paul Alapatt. "Comparative Performance A...	<1%
32	Publication	Mays Kareem Jabbar, Hafedh Trabelsi, Thaar. A. Kareem. "Improved Hypergraph ...	<1%
33	Publication	"Advance Concepts of Image Processing and Pattern Recognition", Springer Scien...	<1%
34	Student papers	Delhi Technological University on 2026-04-09	<1%
35	Student papers	FICT on 2026-05-03	<1%
36	Publication	Ritu Kumari, Kusum Dalal. "Adaptive Clustering and Cluster Head Selection in VA...	<1%
37	Internet	dataforsolution.github.io	<1%
38	Student papers	Delhi Technological University on 2026-05-25	<1%

39	Publication	Poonam Nandal, Tapas Kumar, Meeta Singh, Mamta Dahiya. "Progressive Compu...	<1%
40	Publication	Wei Zhou, Xiaorui Xue, Yizhen Xu. "Credit card fraud detection based on self-pace...	<1%
41	Internet	downloads.hindawi.com	<1%
42	Internet	www.ijcaonline.org	<1%
43	Publication	"Data Mining Approach for Detection of DDoS Attacks Utilizing SSL/TLS Protocol", ...	<1%
44	Publication	"International Conference on Smart Systems and Emerging Technologies", Spring...	<1%
45	Student papers	AUT University on 2025-10-26	<1%
46	Publication	Idris Shehu Musa, Muhammed Bashir Mu'azu, Yusuf Ibrahim. "An EfficientNet-Ba...	<1%
47	Student papers	University of Sunderland on 2026-07-03	<1%
48	Internet	ceur-ws.org	<1%
49	Internet	eiid.eu	<1%
50	Internet	hal.science	<1%
51	Student papers	AUT University on 2014-10-14	<1%
52	Publication	Ali Jalooli, Frankie Murcia. "Vehicular Edge Computing-Driven Optimized Multiho...	<1%

53	Publication	Geeth P. Wijesiri N.B.A, Jussi Haapola, Tharaka Samarasinghe. "A Discrete-Time M...	<1%
54	Student papers	The University of Manchester on 2011-05-06	<1%
55	Publication	Tirtha Majumder, Rajiv Kumar Mishra, Sudhansu Sekhar Singh, Prasanna Kumar ...	<1%
56	Internet	journal2.uad.ac.id	<1%
57	Internet	theses.hal.science	<1%
58	Internet	www.csroc.org.tw	<1%
59	Student papers	Gujarat Technological University on 2026-03-21	<1%
60	Publication	Irshad Ullah, Xiaoheng Deng, Xinjun Pei, Husnain Mushtaq, Muhammad Uzair, Sh...	<1%
61	Publication	Parveen Kumar, Bijay Kumar Singh. "Enhanced VANET Performance by Stable Clu...	<1%
62	Student papers	SASTRA University on 2022-09-24	<1%
63	Student papers	Sridevi Women's Engineering College on 2026-07-17	<1%
64	Student papers	University of Bedfordshire on 2019-09-25	<1%
65	Student papers	University of Limerick on 2023-08-24	<1%
66	Internet	ais.khpi.edu.ua	<1%

67	Internet	appliednetsci.springeropen.com	<1%
68	Internet	journal.scsa.ge	<1%
69	Internet	shura.shu.ac.uk	<1%
70	Internet	stars.library.ucf.edu	<1%
71	Internet	tudr.thapar.edu:8080	<1%
72	Publication	"Artificial Intelligence of Things", Springer Science and Business Media LLC, 2024	<1%
73	Publication	Akhilesh Bijalwan, Iqram Hussain, Kamlesh Chandra Purohit, M. Anand Kumar. "E...	<1%
74	Student papers	Anil Neerukonda Institute of Technology & Sciences on 2026-01-27	<1%
75	Student papers	Anna University on 2026-03-12	<1%
76	Student papers	Arts, Sciences & Technology University In Lebanon on 2024-08-16	<1%
77	Publication	Aya G. Ayad, Nehal A. Sakr, Noha A. Hikal. "A hybrid approach for efficient feature...	<1%
78	Publication	Boubiche Djallel Eddine, Sedat Akleylek. "Generative AI for Cybersecurity", CRC Pr...	<1%
79	Student papers	College of Public Health on 2026-03-09	<1%
80	Student papers	Leeds Beckett University on 2022-05-31	<1%

81	Publication	Mays Kareem Jabbar, Hafedh Trabelsi. "A Novelty of Hypergraph Clustering Mode...	<1%
82	Student papers	National University of Singapore on 2021-03-05	<1%
83	Publication	Shailendra Pushkin, Ranvijay. "A Novel Hypergraph Clustered Gray Relational An...	<1%
84	Student papers	The Scientific & Technological Research Council of Turkey (TUBITAK) on 2026-05-31	<1%
85	Student papers	University Der Es Salaam on 2025-05-02	<1%
86	Student papers	University of Ruhuna Matara on 2026-07-14	<1%
87	Publication	Yonghao Xu, Bo Du, Liangpei Zhang. "Beyond the Patchwise Classification: Spectr...	<1%
88	Internet	archives.christuniversity.in	<1%
89	Internet	escholarship.org	<1%
90	Internet	ijact.in	<1%
91	Internet	mafiadoc.com	<1%
92	Student papers	nith on 2022-05-12	<1%
93	Publication	"Innovations in Computational Intelligence and Computer Vision", Springer Scien...	<1%
94	Student papers	Anna University on 2025-01-06	<1%

95	Student papers	Anna University on 2025-12-13	<1%
96	Student papers	Anna University on 2026-03-02	<1%
97	Student papers	Chandigarh University on 2015-06-28	<1%
98	Student papers	Delhi Technological University on 2026-06-14	<1%
99	Publication	Gulista Khan, Sumit Kumar, Wajid Ali, Kamal Kumar Gola, Rohit Kanauzia. "Enhanc...	<1%
100	Publication	J.V.G. Ferreira, M.E.S. Freire, E.M. Cruz, C.V.S. Prazeres, G.B. Figueiredo, M.L.M. Pei...	<1%
101	Publication	Jie Cui, Hu Sun, Hong Zhong, Jing Zhang, Lu Wei, Irina Bolodurina, Debiao He. "Col...	<1%
102	Student papers	King Abdulaziz University - Scientific Research on 2026-01-28	<1%
103	Publication	M. Chitra, S. Siva Sathya. "Bidirectional Data Dissemination in Vehicular Ad Hoc N...	<1%
104	Publication	Mahmood A Jumaah, Yossra H Ali, Tarik A. Rashid. "Artificial Liver Classifier: A Ne...	<1%
105	Publication	Prashanta Kumar Patra, Santosh Kumar Maharana. "Semantic-aware, graph-base...	<1%
106	Publication	Rotich, Gilbert Kibet. "Geospatial Understanding of Foundation Models: Self-Supe...	<1%
107	Student papers	University Tun Hussein Onn Malaysia on 2016-08-11	<1%
108	Student papers	University of Auckland on 2012-06-13	<1%

109	Student papers	University of Glasgow on 2025-08-13	<1%
110	Student papers	University of Leeds on 2010-09-02	<1%
111	Student papers	University of Newcastle on 2023-07-14	<1%
112	Student papers	University of Ruhuna Matara on 2026-01-20	<1%
113	Student papers	University of Witwatersrand on 2025-09-13	<1%
114	Publication	Xinwei Yang, Tianqi Yu, Zhongyue Chen, Jianfeng Yang, Jianling Hu, Yingrui Wu. "...	<1%
115	Internet	assets-eu.researchsquare.com	<1%
116	Internet	assets.researchsquare.com	<1%
117	Internet	boa.unimib.it	<1%
118	Internet	cdn.techscience.cn	<1%
119	Internet	eprints.staffs.ac.uk	<1%
120	Internet	hdl.handle.net	<1%
121	Internet	prod-ms-be.lib.mcmaster.ca	<1%
122	Internet	ruor.uottawa.ca	<1%

123	Internet	sci-hub.st	<1%
124	Internet	theses.lib.polyu.edu.hk	<1%
125	Internet	trepo.tuni.fi	<1%
126	Internet	uhra.herts.ac.uk	<1%
127	Internet	www.dspace.dtu.ac.in:8080	<1%
128	Internet	www.mdpi.com	<1%
129	Internet	www2.eecs.berkeley.edu	<1%

PERFORMANCE ENHANCEMENT OF VEHICULAR ADHOC NETWORKS USING INTELLIGENT TECHNIQUES

A Thesis Submitted

in Partial Fulfillment of the Requirements for the Degree of

DOCTOR OF PHILOSOPHY

in

Computer Science and Engineering

by

ASHISH KUMARI
(2K21/PhDCO/505)

Under the supervision of

Prof. Shailender Kumar

Professor

Department of Computer Science and Engineering
Delhi Technological University, Delhi

&

Dr. Ram Shringar Raw

Department of Computer Science and Engineering
NSUT (East Campus), Delhi



Department of Computer Science and Engineering
DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Shahbad Daultpur, Main Bawana Road, Delhi-110042, India

July, 2026



DELHI TECHNOLOGICAL UNIVERSITY

Formerly Delhi College of Engineering

Shahbad Daulatpur, Main Bawana Road, Delhi –42

CANDIDATE DECLARATION

I, Ashish Kumari (2K21/PhDCO/505), hereby certify that the work which is being presented in the thesis entitled “PERFORMANCE ENHANCEMENT OF VEHICULAR AD-HOC NETWORKS USING INTELLIGENT TECHNIQUES” in partial fulfilment of the requirements for the award of the Degree of Doctor of Philosophy, submitted in the Department of Computer Science and Engineering, Delhi Technological University is an authentic record of my own work carried out during the period from January 2022 to June 2026 under the supervision of Prof. Shailender Kumar from the Department of Computer Science and Engineering at Delhi Technological University, and Dr. Ram Shringar Raw from the Department of Computer Science and Engineering at Netaji Subhas Institute of Technology (East Campus), New Delhi.

The matter presented in the thesis has not been submitted by me for the award of any other degree of this or any other Institute.

Candidate's Signature

This is to certify that the student has incorporated all the corrections suggested by the examiners in the thesis and the statement made by the candidate is correct to the best of our knowledge.

Signature of Supervisors(s)

Signature of External Examiner



DELHI TECHNOLOGICAL UNIVERSITY

Formerly Delhi College of Engineering

Shahbad Daulatpur, Main Bawana Road, Delhi –42

CERTIFICATE BY THE SUPERVISOR(S)

Certified that Ashish Kumari (2K21/PhDCO/505) has carried out their search work presented in this thesis entitle “**Performance Enhancement of Vehicular Ad-Hoc Networks using Intelligent Techniques**” for the award of Doctor of Philosophy from Department of Computer Science and Engineering, Delhi Technological University, Delhi, under our supervision. The thesis embodies results of original work, and studies are carried out by the student herself and the contents of the thesis do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

Prof. Shailender Kumar

Supervisor

Department of CSE,
Delhi Technological University,
Delhi –110042, India

Dr. Ram Shringar Raw

Co-Supervisor

Department of CSE,
NSUT (East Campus).
Delhi, India

Date:

ABSTRACT

4 Vehicular Ad Hoc Networks (VANETs) form the communication backbone of Intelligent Transportation Systems (ITS), enabling vehicle-to-vehicle and vehicle-to-infrastructure exchange for safety-critical and traffic-management services. However, the high mobility of vehicles, rapidly changing topology, fluctuating density, and intermittent connectivity make stable and reliable communication difficult, producing unstable links, frequent cluster reformation, control overhead, and degraded performance. Clustering mitigates these problems by organising vehicles into manageable groups, yet conventional distance-, mobility-, and trust-based techniques adapt poorly to dynamic conditions, rely on subjective or fixed cluster-head selection weights, and model the network as a simple pairwise graph that cannot capture the higher-order interactions of dense urban traffic. Reliable communication is further threatened by intrusions such as spoofing, packet dropping, and denial-of-service attacks that cryptographic mechanisms alone cannot detect. This thesis addresses these coupled challenges by developing an intelligent, stability-aware, and secure clustering framework built on hypergraph-based spectral clustering, objective multi-criteria cluster-head selection, and graph-based deep learning intrusion detection.

The research pursues four objectives: to analyse existing VANET clustering algorithms, to develop scalable spectral clustering algorithms, to implement an efficient stability-based model benchmarked against the state of the art, and to analyse the reliability of the selected cluster heads. Guided by a bibliometric study of the field and an investigation of V2V and V2I communication reliability, two spectral clustering frameworks are proposed. The first represents the vehicular network as a weighted hypergraph, fixes the optimal number of clusters through an eigen-gap heuristic, forms clusters using an Improved Tensor Trace Maximisation (iTMM) procedure over a modularity-based Laplacian, and elects cluster heads through an objective weighted Criteria Importance Through Intercriteria Correlation (CRITIC) decision over four criteria—relative speed, connectivity level, link lifetime, and consensus trust—achieving 84% cluster-head stability with only three clusters, together with higher throughput and reduced packet delay. The second framework

retains the hypergraph foundation but introduces a modified tensor-trace-maximisation pipeline with Gaussian Mixture Model assignment and an Adaptive Neighbouring Behaviour-Incentivization (ANB-I) mechanism that rewards steady vehicles and penalises erratic ones, achieving 75% stability with three clusters—a 4.17% improvement over the hypergraph baseline—and consistently low packet delay across sparse and dense networks. Both frameworks were evaluated on a realistic urban scenario derived from Connaught Place, Delhi, using SUMO-generated mobility streamed into MATLAB.

To secure this stable clustering, a Knowledge Graph-based Deep Convolutional Neural Network (KG-DeepCNN) Intrusion Detection System (IDS) is deployed at the roadside unit for intrusion-aware access control. The model combines knowledge-graph and graph convolutional network embeddings with a weighted ensemble of four heterogeneous CNN tunnels, and was validated on the BoT-IoT, ToN-IoT, UNSW-NB15, and NSL-KDD benchmarks, attaining accuracies of approximately 99%, 98%, 98%, and 99% respectively, with a 2.01% accuracy gain on UNSW-NB15 over comparable methods and composite cluster-head reliability between 98.43% and 99.38%.

Taken together, the results demonstrate that objective, behaviour-aware clustering on a scalable spectral foundation, protected by knowledge-driven intrusion detection, yields VANET communication that is simultaneously stable, reliable, and secure. By sustaining stable cluster heads, minimising packet delay, and screening malicious access requests, the proposed frameworks support the timely delivery of safety-critical messages, reduce signalling overhead, and contribute to safer and more efficient intelligent transportation systems.

Keywords: VANET; hypergraph spectral clustering; cluster-head stability; Improved Tensor Trace Maximization; weighted CRITIC; incentive-based cluster-head selection; knowledge graph; deep learning; intrusion detection; reliable and secure communication.

ACKNOWLEDGEMENT

Above all, I bow in gratitude to Almighty God, whose grace bestowed upon me the strength, perseverance, and clarity of purpose that carried me through this research journey.

I owe an immeasurable debt of gratitude to my supervisors, Prof. Shailender Kumar and Dr. Ram Shringar Raw, whose scholarly guidance, patient mentorship, and thoughtful criticism shaped this work at every stage. I am especially indebted to Prof. Shailender Kumar, whose technical insight illuminated the path whenever I encountered obstacles.

I extend my sincere appreciation to Prof. Vinod Kumar, Prof. Manoj Kumar, Prof. Rajni Jindal, Prof. Rahul Kataria, Prof. Aruna Bhat, Dr. Rajesh Kumar Yadav, and Dr. Devanand for their steady encouragement and inspiring vision. I am grateful to Prof. Anil Singh Parihar, Head of the Department (CSE), for his valuable feedback, and to the esteemed faculty of the Department of Computer Science and Engineering particularly Dr. Rajeev Kumar, Dr. Pawan Singh Mehra, and Dr. Rahul Kumar for their generous support. My thanks also go to Prof. Prateek Sharma, Vice-Chancellor, DTU, for cultivating a research environment in which this work could flourish.

With profound love and reverence, I dedicate this thesis to the cherished memory of my late father, Mr. Suresh Kumar Duddi, whose blessings, values, and sacrifices remain my guiding light. To my beloved mother, Mrs. Nirmla Devi, I owe endless gratitude for her boundless love and unwavering faith in me. I am equally thankful to my in-laws, Mrs. Sneh Lata and Mr. Jageram Bohra, whose affection and blessings have continually strengthened my resolve.

My deepest gratitude belongs to my husband, Dr. Navdeep Bohra, whose love, patience, and steadfast belief in me have been my greatest source of strength throughout this endeavour. Finally, my heart overflows with love for my dear sons, Vedant Bohra and Virat Bohra, whose bright smiles and joyful presence have been my constant inspiration and my truest reward.

To all these remarkable people, this acknowledgment stands as a humble token of my heartfelt appreciation.

Ashish Kumari
(2K21/PhDCO/505)

TABLE OF CONTENTS

1.1 Overview of VANET and ITS	2
1.1.1 VANET Architecture	2
1.1.2 Communication Technologies and Infrastructure	3
1.1.3 Role of VANET in Intelligent Transportation Systems	3
1.2 Performance and Stability Limitations in VANET	4
1.2.1 High Mobility and Dynamic Topology	4
1.2.2 Link Instability and Frequent Disconnections	5
1.2.3 Scalability Issues in Sparse and Dense Networks	5
1.3 Requirement of Clustering in VANET	5
1.3.1 Governing Components of a Cluster	6
1.3.2 Cluster Head Selection and Responsibilities	7
1.3.3 Impact of Clustering on Stability and Performance	7
1.4 Cluster Generation in VANET	8
1.4.1 Insufficiency of Traditional Cluster Generation Approaches	8
1.4.2 Spectral-Based Approach Requirement for Clustering in VANET	9
1.5 Stability Modeling Challenges in VANET Communication	9
1.5.1 Disruptive Impact of Driver Behaviour and Road Topologies	10
1.5.2 Incompetence of Traditional Connectivity Management	10
1.5.3 Need for Adaptive Stability-Based Modeling	10
1.6 Enhancing Link Reliability and Stability in Dynamic VANET Environments	11
1.6.1 Mobility-Aware and Adaptive Reliability Modeling	11
1.6.2 Intelligent and Reliability-Driven Secure Communication in VANETs ..	11
1.7 Problem Statement	12
1.8 Research Objectives	13
1.9 Motivation of the Research Work	13
1.10 Organization of the Thesis	14
2.1 Clustering Methodologies in VANET	17
2.2 Classification of Clustering Techniques	18
2.2.1 Mobility-Based Clustering	18
2.2.2 Distance-Based Clustering	20
2.2.3 Trust-Based Clustering	21
2.2.4 Spectral-Based Clustering	23
2.3 Cluster Formation and Maintenance Strategies	25

2.4 Cluster Head Selection and Reliability Studies	26
2.4.1 Cluster Head Selection Parameters and Decision Criteria.....	27
2.5 Intelligent and Learning-Based Reliability Enhancement Mechanisms	30
2.6 Comparative Analysis and Research Gaps	32
2.6.1 Critical Discussion of Comparative Tables.....	32
2.6.2 Research Gaps Identification	33
2.6.3 Alignment of Proposed Research with Identified Gaps.....	33
2.7 Chapter Summary.....	36
3.1 Introduction.....	38
3.1.1 Motivation for Stable Cluster Head Selection	39
3.1.2 Research Scope and Chapter Organisation	40
3.2 Network and System Modeling.....	41
3.2.1 VANET Communication and Mobility Model	41
3.2.2 Hypergraph-Based Network Representation.....	42
3.3 Mathematical Modeling of the Proposed Clustering Framework.....	44
3.3.1 Modularity Matrix Formulation	44
3.3.2 Laplacian Matrix Construction	45
3.3.3 Optimal Cluster Determination using Eigen Heuristics.....	46
3.4 Proposed iTTM-Based Cluster Formation	47
3.4.1 Improved Tensor Trace Maximization (iTTM)	48
3.4.2 Gaussian Mixture Model-Based Cluster Generation	49
3.4.3 Spectral Cluster Formation Process	50
3.5 Multi-Criteria Cluster Head Selection Framework.....	50
3.5.1 Relative Speed Modeling	51
3.5.2 Connectivity Level Modeling	52
3.5.3 Link Lifetime Modeling.....	53
3.5.4 Consensus Trust Score Modeling	53
3.5.5 Weighted CRITIC-Based Criteria Evaluation	55
3.6 Proposed iTTM–Weighted CRITIC Optimization Framework.....	55
3.6.1 Integrated Optimization Framework.....	55
3.6.2 Cluster Head Decision Model	57
3.7 Proposed Algorithm Development.....	58
3.7.1 iTTM–Weighted CRITIC Algorithm.....	58
3.8 Theoretical Analysis.....	59
3.8.1 Computational Complexity Analysis	60

3.8.2 Cluster Stability Analysis.....	61
3.8.3 Optimization Analysis.....	62
3.9 Performance Evaluation and Discussion.....	63
3.9.1 Simulation Environment and Experimental Setup.....	63
3.9.2 Comparative Performance Analysis.....	65
3.9.3 Discussion of Key Findings	71
3.10 Chapter Summary.....	72
4.1 Introduction	74
4.1.1 Motivation and Objectives	75
4.1.2 Chapter Organization	76
4.2 Modified Incentive-Based Clustering Framework.....	76
4.2.1 Framework Architecture	76
4.2.2 Incentive-Based Cluster Head Selection Mechanism	79
4.3 Experimental Methodology.....	82
4.3.1 Simulation Environment and Network Configuration	83
4.3.2 Performance Evaluation Metrics.....	84
4.4 Cluster Formation and Stability Analysis	85
4.4.1 Cluster Formation Efficiency	85
4.4.2 Cluster Stability Assessment.....	86
4.5 Cluster Head Selection Performance Evaluation	88
4.5.1 Incentive-Based CH Selection Analysis	88
4.5.2 CH Lifetime and Leadership Performance	90
4.6 Communication Performance Analysis.....	91
4.6.1 Packet Delivery and Throughput Analysis	91
4.6.2 Delay and Communication Reliability Analysis.....	92
4.7 Comparative Performance Evaluation	94
4.7.1 Comparison with Existing Clustering Approaches	94
4.7.2 Performance Improvement Assessment.....	96
4.8 Scalability and Overhead Analysis	96
4.8.1 Scalability under Different Traffic Conditions	96
4.8.2 Communication and Computational Overhead.....	97
4.9 Discussion	98
4.9.1 Key Findings	98
4.9.2 Practical Implications for VANET Deployment.....	98
4.10 Chapter Summary.....	99

5.1	Introduction	101
5.1.1	Reliable and Secure Communication in VANETs.....	102
5.1.2	Security Challenges and Intrusion Detection in VANETs.....	103
5.2	Proposed KG-DeepCNN Framework	107
5.2.1	Framework Architecture	107
5.2.2	Knowledge Graph-Based Feature Representation	109
	<i>Knowledge graph construction</i>	110
	<i>Semantic feature generation with the GCN</i>	112
5.2.3	DeepCNN-Based Attack Detection Mechanism	113
5.3	Experimental Setup	116
5.3.1	Dataset and Simulation Environment.....	116
5.3.2	Performance Evaluation Metrics	118
5.4	Results and Discussion.....	118
5.4.1	Intrusion Detection Performance	118
	BoT-IoT.....	119
	ToN-IoT	119
	NSL-KDD	120
	UNSW-NB15	121
	Model Loss.....	121
	Per-class behaviour	122
5.4.2	Reliable Communication Performance	125
5.5	Comparative Analysis	127
5.5.1	Comparison with Existing Methods.....	127
5.5.2	Impact on Communication Reliability	129
5.6	Chapter Summary.....	130
6.1	Conclusion	132
6.2	Research Contributions	133
6.3	Research Limitations.....	135
6.4	Future Research Directions	135
6.5	Social Impact.....	136

LIST OF TABLES

Table 1.1	Key traits of VANETs and the resulting communication challenges
Table 1.2	Benefits of clustering in overcoming key VANET issues
Table 1.3	Need for spectral clustering in VANETs compared with traditional clustering
Table 2.1	Comparative Analysis of mobility-based clustering schemes
Table 2.2	Comparative Analysis of distance-based clustering schemes
Table 2.3	Comparative Analysis of trust-based clustering schemes
Table 2.4	Comparative Analysis of spectral-based clustering schemes
Table 2.5	Comparative Analysis of representative CH-selection studies
Table 2.6	Comparative Analysis of CH selection parameters and performance parameters
Table 2.7	Comparative Analysis of representative learning-based VANET-security methods
Table 2.8	Mapping of Research Gaps to Research Objectives
Table 2.9	Research Objectives mapping with the publications
Table 3.1	Computational complexity of the four proposed algorithms
Table 3.2	Specifications of the simulation environment
Table 3.3	Specifications of the network parameters (IEEE 802.11p)
Table 3.4	State-of-the-art comparison of the proposed framework against representative baselines
Table 3.5	Cluster-head change rate per second versus vehicle speed
Table 4.1	Simulation parameters
Table 4.2	Specifications of the IEEE 802.11p physical layer
Table 4.3	State-of-the-art comparison based on cluster-head stability
Table 4.4	State-of-the-art comparison based on network performance (packet delay)
Table 5.1	Overview of related work in VANET and related intrusion detection
Table 5.2	Parameter setting for the CNN used in tunnels 1 to 4
Table 5.3	Simulation settings for the VANET scenario
Table 5.4	Learning hyperparameters for the GCN and the DeepCNN
Table 5.5	Cluster head reliability indicators derived from the per-class confusion-matrix counts of Figures 5.10 to 5.13
Table 5.6	State-of-the-art comparison of the proposed KG-DeepCNN across the four datasets

LIST OF FIGURES

Figure 1.1	Basic diagram of the VANET architecture.....
Figure 1.2	Representation of clustering in a VANET.....
Figure 2.1	Classification of clustering techniques in VANET.....
Figure 3.1	VANET hypergraph representation: (a) the vehicular graph and (b) the equivalent hyperedge grouping of mutually communicating vehicles
Figure 3.2	Proposed scheme for stable cluster-head selection in VANET
Figure 3.3	Optimal number of clusters analysis.....
Figure 3.4	Cluster generation at the highest vehicle density.....
Figure 3.5	Overall stability and the effect of each individual vehicle attribute on the stability analysis
Figure 3.6	Evaluation on the sparse network of 100 vehicles.....
Figure 3.7	Evaluation on the dense network of 1000 vehicles
Figure 3.8	Analysis of varying vehicle density.....
Figure 4.1	Architecture of the modified incentive-based clustering framework
Figure 4.2	Incentivized cluster-head selection process
Figure 4.3	Map of Delhi (Connaught Place) and the three clusters formed at the instant of maximum vehicle density
Figure 4.4	Cluster-wise evaluation of cluster-head stability for 100 vehicles: the proposed model against the individual incentivized selection metrics
Figure 4.5	Cluster-wise evaluation of cluster-head stability for 1000 vehicles: the proposed model against the individual incentivized selection metrics
Figure 4.6	Evaluation of the proposed incentivized Adaptive-RAS parameter.....
Figure 4.7	Evaluation of the proposed incentivized Adaptive-ND parameter.....
Figure 4.8	Evaluation of the proposed incentivized Adaptive- LLT parameter
Figure 4.9	Evaluation of the proposed incentivized ANB-I model against the individual selection metrics over varying vehicle density
Figure 4.10	Cluster-wise evaluation of throughput for 100 vehicles.....
Figure 4.11	Cluster-wise evaluation of throughput for 1000 vehicles.....
Figure 4.12	Cluster-wise evaluation of packet delay for 100 vehicles
Figure 4.13	Cluster-wise evaluation of packet delay for 1000 vehicles
Figure 5.1	The proposed VANET system model, in which the RSU applies the IDS to grant or deny access requests
Figure 5.2	The proposed KG-DeepCNN model.....
Figure 5.3	Impact of the cosine-similarity threshold on classification accuracy across the four datasets.....
Figure 5.4	The overlapping clusters from the CNN (left) contrast with the well-separated, orthogonal clusters from the GCN embeddings (right)
Figure 5.5	Classification metrics of the proposed ensemble and tunnels 1 to 4 on the BoT-IoT dataset.....

Figure 5.6	Classification metrics of the proposed ensemble and tunnels 1 to 4 on the ToN-IoT dataset
Figure 5.7	Classification metrics of the proposed ensemble and tunnels 1 to 4 on the NSL-KDD dataset
Figure 5.8	Classification metrics of the proposed ensemble and tunnels 1 to 4 on the UNSW-NB15 dataset
Figure 5.9	Training loss curves for the ensemble and tunnels 1 to 4 on the BoT-IoT, ToN-IoT, UNSW-NB15, and NSL-KDD datasets.....
Figure 5.10	Confusion matrix for the NSL-KDD dataset
Figure 5.11	Confusion matrix for the BoT-IoT dataset
Figure 5.12	Confusion matrix for the ToN-IoT dataset
Figure 5.13	Confusion matrix for the UNSW-NB15 dataset
Figure 5.14	Cluster head reliability of the proposed KG-DeepCNN

LIST OF ABBREVIATIONS

Abbreviation	Full Form
ADAS	Advanced Driver Assistance Systems
ANB-I	Adaptive Neighbouring Behaviour-Incentivization
BoT-IoT	Botnet Internet of Things (benchmark intrusion-detection dataset)
CAM	Cooperative Awareness Message
CAN	Controller Area Network
CH	Cluster Head
CM	Cluster Member
CNN	Convolutional Neural Network
CRITIC	Criteria Importance Through Intercriteria Correlation
C-V2X	Cellular Vehicle-to-Everything
DBN	Deep Belief Network
DDoS	Distributed Denial of Service
DENM	Decentralized Environmental Notification Message
DL	Deep Learning
DNN	Deep Neural Network
DoS	Denial of Service
DSRC	Dedicated Short-Range Communication
EDCA	Enhanced Distributed Channel Access
GAN	Generative Adversarial Network
GCN	Graph Convolutional Network
GMM	Gaussian Mixture Model
IDS	Intrusion Detection System
IoV	Internet of Vehicles
ITS	Intelligent Transportation System
iTTM	improved Tensor Trace Maximization
KG-DeepCNN	Knowledge Graph-based Deep Convolutional Neural Network
LLT	Link Lifetime
MANET	Mobile Ad Hoc Network
MCDM	Multi-Criteria Decision Making
ML	Machine Learning
MTTM	Modified Tensor Trace Maximization

Abbreviation	Full Form
NSL-KDD	Network Security Laboratory–KDD (benchmark intrusion-detection dataset)
OBU	On-Board Unit
OFDM	Orthogonal Frequency Division Multiplexing
PD	Packet Delay
PDR	Packet Delivery Ratio
PU	Primary User
QoS	Quality of Service
RSU	Roadside Unit
SDN	Software-Defined Networking
SU	Secondary User
SUMO	Simulation of Urban Mobility
ToN-IoT	Telemetry of Things Internet of Things (benchmark intrusion-detection dataset)
TTM	Tensor Trace Maximization
UNSW-NB15	University of New South Wales Network-Based 2015 (benchmark intrusion-detection dataset)
V2I	Vehicle-to-Infrastructure
V2V	Vehicle-to-Vehicle
V2X	Vehicle-to-Everything
VANET	Vehicular Ad Hoc Network
WAVE	Wireless Access in Vehicular Environments

CHAPTER 1

INTRODUCTION

A Vehicular Ad Hoc Network (VANET) is a specialized variant of the Mobile Ad Hoc Network (MANET) that enables wireless communication among moving vehicles and between vehicles and fixed roadside infrastructure. VANETs have attracted significant attention for their potential to improve road safety, traffic efficiency, and advanced driver-assistance services [1], [2]. They differ fundamentally from traditional mobile networks because nodes move at high velocity, constrained by road geometry and traffic rules, giving rise to unique physical-layer and topological characteristics [3], [4]. As a building block of the Intelligent Transportation System (ITS), vehicles are equipped with interfaces supporting Vehicle-to-Vehicle (V2V), Vehicle-to-Infrastructure (V2I), and Vehicle-to-Everything (V2X) interactions, continuously exchanging speed, position, congestion alerts, and accident notifications [5], [6]. This exchange uses periodic Cooperative Awareness Messages (CAMs) and event-driven Decentralized Environmental Notification Messages (DENMs), which demand high reliability and low latency for safety-critical maneuvers [7], [8].

Communication is standardized through IEEE 802.11p for Wireless Access in Vehicular Environments, providing Dedicated Short-Range Communication (DSRC) in the 5.9 GHz band, whose 75 MHz spectrum is split into one control and six service channels [5], [6]. Because nodes are vehicles, high mobility causes frequent position changes that create severe instability and scalability problems, and node latency is higher than in a MANET [1], [9]. Doppler shift and multipath fading degrade signal integrity [10], [11], while rapid, unpredictable topology changes hinder reliable communication and data dissemination [12], [2]. Frequent reconfigurations produce unreliable V2V links [12], whose stability is quantified by the Link Lifetime (LLT), the duration two nodes stay within range given their relative velocity and trajectory [13], [14]. Maintaining communication quality while minimizing group

reorganizations is therefore essential, and clustering has emerged as one of the most promising techniques for this [5], [15].

1.1 Overview of VANET and ITS

VANETs form the communication backbone of the ITS paradigm, whose objective is safer, greener, and more efficient transportation through intelligent information exchange. This section introduces the VANET architecture, its enabling communication technologies, and its role in the broader ITS ecosystem.

1.1.1 VANET Architecture

A typical VANET comprises three principal entities: the On-Bord Unit (OBU) in every vehicle, the Roadside Unit (RSU) along the road, and a central Trusted Authority or back-end infrastructure. The OBU senses, processes, and transmits data, interfacing with internal sensors over the Controller Area Network (CAN) bus to gather metrics such as acceleration, braking, and engine diagnostics [16], [17], while the RSU acts as a gateway to the fixed network. Three communication patterns are distinguished: V2V (direct exchange of safety and traffic messages), V2I (between vehicles and RSUs or base stations), and inter-RSU communication connecting roadside infrastructure to the core network. The Trusted Authority acts as a root of trust, managing certificates and cryptographic keys for authentication and privacy and preventing false-information injection [18], [19]. Figure 1.1 illustrates a representative VANET architecture.

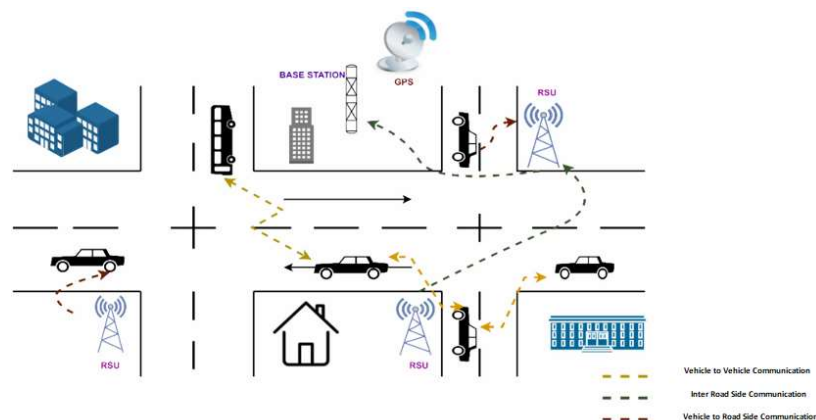


Figure 1.1. Basic diagram of the VANET architecture

As shown in Figure 1.1, vehicles communicate through V2V links and with RSUs through V2I links, while RSUs connect to base stations and GPS infrastructure for traffic management, information dissemination, and reliable communication. The architecture is decentralized and self-organizing: vehicles join and leave continuously and no fixed topology can be assumed, which is the root cause of many stability and reliability challenges addressed in this thesis. The absence of a central coordinator necessitates distributed resource-allocation algorithms, such as the Enhanced Distributed Channel Access (EDCA) mechanism in WAVE that prioritizes safety messages over infotainment [20], [21].

1.1.2 Communication Technologies and Infrastructure

1 VANET communication is primarily governed by the IEEE 802.11p / WAVE stack in the 5.9 GHz DSRC band, engineered for high relative speeds and short contact times, using an Orthogonal Frequency Division Multiplexing (OFDM) physical layer to mitigate frequency-selective fading from urban obstructions [8], [22]. More recently, Cellular V2X (C-V2X) based on LTE and 5G New Radio complements DSRC with wider coverage and higher throughput, using the PC5 interface for direct V2V and the Uu interface for vehicle-to-network communication, offering a stronger link budget in dense scenarios [23], [24]. The supporting infrastructure, comprising RSUs, base stations, and GPS, provides localization, time synchronization, and connectivity to cloud and edge resources, with RSUs often hosting Vehicular Edge Computing nodes that process latency-sensitive data closer to vehicles [25], [26]. The heterogeneity of these technologies, combined with node mobility, motivates intelligent network-organization techniques such as clustering that operate across diverse access technologies [6], [9].

1.1.3 Role of VANET in Intelligent Transportation Systems

Within ITS, VANET applications are broadly safety-related (collision avoidance, emergency brake warnings, intersection assistance), efficiency-related (traffic-flow optimization, congestion management, cooperative navigation), and comfort or infotainment services. Safety applications impose stringent low-latency and high-reliability requirements, since delayed or lost messages can compromise road safety. The network must ensure timely dissemination despite rapid topology changes, high

mobility, and dynamic traffic, demanding efficient clustering, stable Cluster Head (CH) selection, and reliable routing [9], [27], [28]. Reliable delivery in dense, fast-changing traffic depends on the stability of the communication structure, which depends on how vehicles are organized and how forwarding is assigned [15], [4]; clustering suppresses redundant rebroadcasts that saturate the channel, maintaining a critical Packet Delivery Ratio (PDR) at peak hours [29], [30].

1.2 Performance and Stability Limitations in VANET

Despite their advantages, VANETs face significant performance and stability limitations owing to high mobility and continuously changing topology [4], [18], [20]. Variations in speed, direction, and density cause unstable links, route failures, and control overhead [18], [21], [22], while the limited range of wireless transceivers causes frequent link breakages and reduced connectivity [6], [14], affecting cluster stability, reliability, and overall performance [4], [16], [29]. Table 1.1 summarizes the key VANET traits and corresponding challenges.

Table 1.1. Key traits of VANETs and the resulting communication challenges.

Key Trait of VANET	Resulting Challenge
High vehicle mobility	Frequent link disconnections and short-lived communication links
Rapidly changing topology	Difficulty maintaining stable routes and cluster structures
Variable vehicle density	Scalability problems in both sparse and dense traffic
Stringent latency needs	Hard to guarantee low-latency, reliable message delivery
Predictable road-constrained motion	Opportunity for mobility-aware route and stability estimation

1.2.1 High Mobility and Dynamic Topology

Vehicles move at high, frequently varying speeds, so relative positions change continuously, producing a highly dynamic topology [4], [18], [20]. Links form and break frequently, causing route reconfiguration, control overhead, packet loss, and reduced reliability [14], [18], [21], and reducing the effectiveness of conventional routing and clustering protocols [4], [16], [22]. Rapid variations shorten link lifetimes and trigger repeated maintenance, making stable cluster heads hard to sustain [9], [18],

[20]. In highway platooning the relative speed within a group is small but large between opposing lanes, causing rapid neighborhood changes [11], [14]; dense urban environments with frequent lane changes, signals, and intersections further complicate topology management, necessitating mobility-aware, predictive clustering [20], [21], [22].

1.2.2 Link Instability and Frequent Disconnections

Because link duration is governed by relative velocity and transmission range, links are inherently short-lived. Sudden acceleration, braking, lane changes, and overtaking further shorten link lifetimes and cause frequent disconnections. These disruptions are modeled through the probability of link failure, which increases exponentially as the distance between two nodes approaches the maximum transmission range R [12], [27]. Unreliable links directly degrade the PDR and increase end-to-end delay, undermining the safety applications VANETs are meant to support [12], [27].

1.2.3 Scalability Issues in Sparse and Dense Networks

VANET performance must remain acceptable across widely varying densities. In sparse networks, scarce relays cause partitioning and intermittent connectivity, requiring Carry-and-Forward strategies akin to Delay Tolerant Networks [9], [20], which raises delay, reduces PDR, and hinders stable CH formation [18], [21]. In dense networks, many neighbors competing for the shared channel create the broadcast-storm problem, with redundant transmissions, collisions, congestion, and overhead [8], [14], [21] that degrade throughput and increase delay [14], [21], while frequent topology changes force repeated maintenance and CH re-election [9], [18], [22]. A scalable framework must therefore adapt to sparse and dense environments without manual tuning, ensuring reliable connectivity and stable performance [9], [20], [21], [24].

1.3 Requirement of Clustering in VANET

Clustering groups vehicles into logical clusters to improve communication efficiency, scalability, and stability, localizing signaling to reduce global overhead [4], [7], [9]. Within each cluster, one vehicle is elected as the cluster head, acting as a local coordinator and virtual backbone, while the remaining Cluster Members (CMs)

communicate through it. The CH handles cluster management, intra-cluster coordination, routing support, and communication with neighboring CHs and RSUs, minimizing redundant transmissions [6], [14], [21]. Clustering reduces routing complexity, limits route discoveries, decreases control overhead, and mitigates the broadcast storm in dense traffic [8], [9], [14], while confining topology updates within clusters for scalability [9], [18], [21]. Selecting stable CHs by mobility, connectivity, and link stability increases cluster lifetime, improves PDR, and reduces delay for safety-critical dissemination [20], [24], [29], [30]. Figure 1.2 illustrates clustering in a VANET.

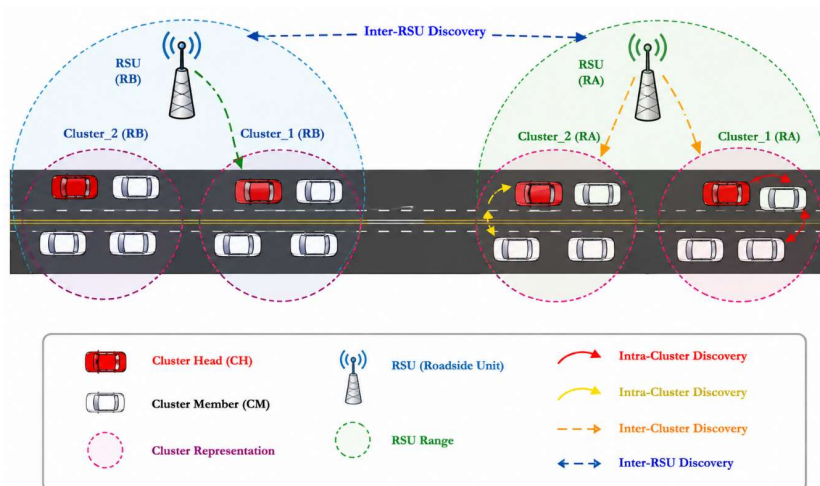


Figure 1.2. Representation of clustering in a VANET

As shown in Figure 1.2, vehicles are organized into clusters in which one vehicle acts as the CH and the rest are CMs. The CH manages intra-cluster communication, adjacent clusters communicate through inter-cluster links, and RSUs coordinate with nearby clusters and exchange information via inter-RSU communication, ensuring seamless connectivity and efficient dissemination.

1.3.1 Governing Components of a Cluster

A cluster is governed by three logical components. The cluster head is the elected coordinator managing membership, aggregating and forwarding data, scheduling communication, and relaying traffic among CMs, neighboring CHs, and RSUs [6], [14], [21]. The CMs are ordinary vehicles associated with a CH that exchange messages through the cluster with reduced routing complexity [9], [14]. Optionally, a

Gateway or Secondary Cluster Head acts as an intermediary between adjacent clusters or a backup coordinator, ensuring continuity when the primary CH leaves range or fails [22], [25]; such redundancy matters in dynamic environments where topology changes would otherwise trigger repeated reformation [18], [20], [22]. Intra-cluster communication is the exchange within one cluster under CH supervision [6], [14], whereas inter-cluster communication spans neighboring clusters through gateways or CHs and between CHs and RSUs, supporting end-to-end dissemination and cooperative applications [7], [21], [25], [30].

1.3.2 Cluster Head Selection and Responsibilities

The effectiveness of clustering depends overwhelmingly on selecting a stable, reliable CH, which serves as the network backbone by managing intra-cluster communication, aggregating and disseminating data, coordinating inter-cluster forwarding, and communicating with neighboring CHs and RSUs at low overhead [6], [14], [21], [9]. These duties require high connectivity, stable links, and a mobility pattern aligned with most CMs to maximize cluster lifetime [18], [20], [22]. Because the responsibilities are concentrated in one node, an unstable CH causes frequent re-clustering, excessive signaling, routing overhead, packet loss, and failures [9], [18], [21], and re-clustering is especially costly since repeated elections interrupt maintenance and delay safety-critical dissemination [4], [9], [18]. Advanced schemes therefore combine multiple parameters, relative speed, neighbor connectivity, link lifetime, mobility prediction, and trustworthiness, rather than a single criterion, to improve stability, PDR, and Quality of Service [9], [18], [20], [24], [29].

1.3.3 Impact of Clustering on Stability and Performance

By organizing vehicles into structured groups with stable CHs, clustering reduces routing complexity and overhead, mitigates the broadcast storm, and improves scalability [8], [9], [14], [21], confining topology updates and control messages within clusters for better channel utilization [9], [14], [21]. CHs enable coordinated intra-cluster scheduling such as TDMA, reducing collisions and giving predictable latency for delay-sensitive applications [6], [14]. Stable clustering minimizes reformation and re-election, improving cluster lifetime and maintaining reliable links despite mobility [18], [20], [22]. Selecting CHs by mobility, connectivity, link stability, and

trustworthiness enhances PDR, lowers delay, improves throughput, and raises the reliability of safety-message dissemination [9], [20], [24], [29], [30], forming the foundation for scalable, stable, reliable VANET communication under sparse and dense conditions [9], [21], [24]. Table 1.2 summarizes these benefits.

Table 1.2. Benefits of clustering in overcoming key VANET issues.

VANET Issue	Contribution of Clustering
Broadcast storm	Localizes traffic within clusters and aggregates messages at the CH
High control overhead	Reduces global signalling by handling coordination through CHs
Frequent handovers	CH-level handover reduces per-vehicle signalling with RSUs
Poor scalability	Hierarchical structure scales to dense traffic
Route instability	Stable CHs provide a durable virtual backbone for routing

1.4 Cluster Generation in VANET

Cluster generation determines the optimal number of clusters and assigns vehicles to them; its quality governs cluster lifetime, stability, and network performance, and must account for both the spatial distribution of vehicles and their temporal mobility trends [25]. An effective mechanism preserves connectivity while accommodating continuous changes in speed, direction, and density, producing compact, well-connected groups with high intra-cluster and low inter-cluster similarity. Choosing an appropriate number of clusters balances overhead against stability, providing a reliable foundation for CH selection and routing [5], [9], [16], [25].

1.4.1 Insufficiency of Traditional Cluster Generation Approaches

Traditional methods such as K-means and distance-threshold schemes are computationally attractive but less effective in the dynamic VANET setting. They rely on Euclidean distance or simple connectivity, which cannot predict link stability when vehicles move at different speeds and directions [9], [16], and K-means requires a predefined number of clusters, hindering adaptation to varying urban densities [5], [9]. Consequently, these methods produce unstable clusters and require frequent re-clustering [5], [9], [10].

1.4.2 Spectral-Based Approach Requirement for Clustering in VANET

Spectral clustering overcomes these limitations by grouping vehicles according to graph structure rather than distance alone. It uses the Laplacian matrix $L = D - A$, where D is the degree matrix and A the adjacency matrix, for dimensionality reduction before clustering in a lower-dimensional space [8], [10], [16]. By transforming the network into a similarity matrix and analysing its spectral properties, vehicles with similar mobility and connectivity are grouped, capturing global community structure that pairwise distance cannot express. The eigengap heuristic automatically determines the optimal number of clusters from the largest gap between consecutive eigenvalues. Spectral methods naturally handle complex, irregular topologies and extend to hypergraph representations that model multi-vehicle relationships, making them well suited to stable, scalable cluster formation [3], [13], [17], [20]. Table 1.3 contrasts traditional and spectral clustering.

Table 1.3. Need for spectral clustering in VANETs compared with traditional clustering

Aspect	Traditional Clustering	Spectral Clustering
Grouping basis	Geometric distance / connectivity	Graph structure and eigen-spectrum
Topology handling	Poor for irregular structures	Handles complex, irregular topologies
Multi-vehicle relations	Pairwise links only	Extensible to hypergraphs
Cluster stability	Brittle under high mobility	More coherent and durable clusters
Number of clusters	Often fixed a priori	Inferred from spectral properties

1.5 Stability Modeling Challenges in VANET Communication

Even with an effective clustering structure, modelling and maintaining stability is non-trivial because vehicle behaviour and the road environment introduce uncertainty that static models cannot capture. The complexity lies in the interaction between physical vehicle dynamics and the stochastic nature of wireless channels [19], [21].

1.5.1 Disruptive Impact of Driver Behaviour and Road Topologies

Driving styles vary widely: aggressive acceleration, abrupt braking, and frequent lane changes shorten link lifetimes and destabilize clusters. Rapid lane-weaving can make an OBU oscillate between two cluster memberships, causing the ping-pong effect and high signaling overhead [2], [7], [11]. Driver behaviour also affects relative speed and spacing, influencing neighborhood stability and link duration, both critical to CH selection [12], [13]. Road topology compounds this: intersections, ramps, and curves cause sudden changes in speed and direction, so a vehicle approaching a junction may rapidly switch neighbours, break distance- or mobility-only assumptions, and trigger premature CH re-selection [8], [21]. Traffic-flow variations from road geometry continuously alter density and connectivity [18], [22], increasing maintenance overhead and reducing lifetime, which highlights the need for mobility-aware, adaptive clustering incorporating both driving behaviour and road topology [9], [20], [21], [22].

1.5.2 Incompetence of Traditional Connectivity Management

Conventional connectivity-management schemes treat connectivity as a static or pairwise metric and assign fixed weights to CH-selection parameters. Such weighting depends on predefined assumptions or expert knowledge and cannot adapt to dynamic traffic, failing to capture the continuously changing, relative nature of vehicular interactions and yielding suboptimal CH selection and reduced stability [9], [24], [29]. This subjectivity is a significant limitation, since a weighting that suits high-speed highways may be ineffective in dense urban traffic [9], [21], [24].

1.5.3 Need for Adaptive Stability-Based Modeling

These observations motivate an adaptive, stability-oriented model in which each selection parameter's contribution is determined objectively from data and adjusted to the relative behaviour of vehicles in real time [9], [21], [24], [29]. One such objective approach is the weighted CRITIC method, which derives weights from each parameter's standard deviation and its correlation with others, removing human bias (proposed methodology; detailed in Chapters 3 and 4). Such a model should jointly consider mobility behaviour, link quality, connectivity, and trustworthiness, and

incentivize vehicles with regular driving patterns to assume CH responsibilities, maximizing cluster lifetime and CH stability [20], [21], [24], [29].

1.6 Enhancing Link Reliability and Stability in Dynamic VANET Environments

The ultimate goal of stability modelling is to enhance link reliability so that safety-critical messages arrive correctly and promptly under all traffic conditions. Reliability is the probability that a packet reaches its destination within a specified time threshold, and is challenging because rapid topology changes, intermittent links, varying densities, and diverse mobility disrupt communication paths [9], [18], [20]. Stability modelling therefore seeks CHs and links that sustain connectivity longer while minimizing route failures, loss, and delay [21], [22], [28]. A reliable framework considers mobility behaviour, link lifetime, connectivity, and trustworthiness [24], [29], and by reducing unnecessary re-clustering it improves PDR, lowers delay, reduces overhead, and provides dependable dissemination of delay-sensitive safety messages [9], [21], [24], [30].

1.6.1 Mobility-Aware and Adaptive Reliability Modeling

Mobility-aware reliability modelling exploits the constrained, road-following nature of vehicular motion to predict short-horizon link persistence and select CHs and routes likely to remain stable, using speed, direction, acceleration, and neighborhood information to estimate future connectivity [18], [20], [21]. It incorporates metrics such as neighbor degree and the variance in relative speed to assess long-term CH suitability [13], [14]. Relative speed, predicted link lifetime, connectivity, and mobility patterns help estimate link stability, letting the network anticipate breakages and proactively re-affiliate members before disconnection [9], [18], [20]. Such proactive modelling minimizes re-clustering, reduces loss and overhead, improves cluster lifetime, and enhances PDR, delay, and efficiency [21], [22], [24], [29], [30].

1.6.2 Intelligent and Reliability-Driven Secure Communication in VANETs

Reliability cannot be separated from security: continuous data exchange between vehicles and RSUs exposes a large attack surface to Denial-of-Service, flooding, and malicious packet injection. A topologically connected network can be functionally

unstable if unreliable or malicious nodes are elected as CHs, since they can drop packets or broadcast false alerts [18], [26]. To counter this, trust scores are computed for each node from both direct observations and indirect neighbor recommendations [17], [27]. Intelligent, reliability-driven mechanisms therefore combine trust evaluation with learning-based intrusion detection at the RSU. This thesis employs a Knowledge-Graph-based Deep Convolutional Neural Network (KG-DeepCNN) to map relationships between network entities and identify anomalous behavior in real time, preserving secure V2V and V2I communication [26], [28].

1.7 Problem Statement

45 Vehicular Ad Hoc Networks are a fundamental ITS component, enabling real-time communication among vehicles and roadside infrastructure to improve road safety and traffic efficiency. However, the highly dynamic environment, with high mobility, rapidly changing topology, fluctuating density, and intermittent connectivity, makes reliable, stable communication difficult [18], [22], [31], frequently causing unstable links, repeated reformation, control overhead, and degraded performance.

Clustering improves scalability and efficiency by organizing vehicles into clusters, yet existing techniques have several limitations. Distance-based approaches rely on geographical proximity, mobility-based methods depend on accurate mobility prediction, and trust-based techniques improve security at the cost of additional overhead [24], [32], [33], [34], [35]. Maintaining stable clusters and long-lasting CHs under highly dynamic traffic thus remains a significant challenge.

Cluster head selection is another critical issue. Most approaches combine mobility, connectivity, neighborhood degree, and link stability using predefined weights or heuristics that fail to adapt to changing conditions [25], [36], [37]. Moreover, conventional graph-based models represent only pairwise relationships and cannot capture the higher-order interactions in dense urban traffic, limiting stability and scalability [10], [16], [25].

Reliable communication is further challenged by security threats such as false information dissemination, packet dropping, spoofing, and denial-of-service attacks. Although cryptographic mechanisms ensure authentication and confidentiality, they

cannot detect intelligent or coordinated attacks in dynamic VANETs. Recent machine learning, deep learning, and graph-based intrusion detection techniques are promising, but their integration with stable clustering remains limited [38], [39], [40].

Therefore, there is a need for an intelligent, adaptive VANET framework that integrates stable hypergraph-based spectral clustering, objective multi-criteria cluster head selection, and graph-based deep learning security to simultaneously improve cluster stability, communication reliability, and network security. Addressing these challenges is the primary objective of this research.

1.8 Research Objectives

To bridge the gaps identified above, the following Research Objectives (ROs) are pursued in this thesis:

RO1: To study and analyse the different clustering algorithms in vehicular ad hoc networks.

RO2: To develop the intelligent algorithms based on scalable spectral clustering algorithms.

RO3: To implement an efficient stability-based model to compare with recent state-of-the-art works.

RO4: To analyse the reliability of the cluster head of the proposed models.

1.9 Motivation of the Research Work

The motivation arises from the observation that VANET stability and reliability hinge on two coupled problems: generating an optimal set of clusters and selecting stable cluster heads. Existing works typically model a VANET as a simple graph in which each vehicle connects to a few neighbours through distance-based edge weights. In reality, a vehicle is influenced simultaneously by many surrounding vehicles that affect its speed, mobility, and communication decisions. This influence field is better captured by hyperedges; a pairwise graph cannot express these higher-order interactions, and a distance-based adjacency matrix discards much network information [20], [17].

Furthermore, selecting the CH that yields the highest stability requires both the correct number of clusters and an objective assessment of each vehicle's dynamic behaviour, speed, reliability, connectivity, and local density, accounting for predicted link lifetime, mobility patterns, and neighborhood information [9], [18], [20], [21]. Fixed, operator-defined weights cannot capture the relativity of vehicular behaviour, which changes with driving style, density, and road context: dense urban settings favour neighbor degree and connectivity, whereas highways favour relative speed, mobility prediction, and link lifetime [9], [20], [22], [28]. These limitations motivate an intelligent, stability-aware clustering framework using hypergraph and spectral techniques for richer representation, objective multi-criteria CH selection, and reliability-driven secure communication to improve stability, reliability, and resilience against attacks [10], [16], [25], [26].

1.10 Organization of the Thesis

The remainder of this thesis is organized into five further chapters, as follows:

Chapter 2 - Literature Review presents a comprehensive review of clustering methodologies in VANETs, classifying them into mobility-based, distance-based, trust-based, and spectral-based techniques, analysing cluster formation and maintenance, CH selection and reliability studies, and learning-based reliability enhancement, and culminating in a consolidated statement of the research gaps.

Chapter 3 - Optimized Cluster Head Selection using the iTTM-Weighted CRITIC Framework presents the proposed hypergraph-based clustering framework, employing the improved Tensor Trace Maximization algorithm for stable cluster generation and the weighted CRITIC method for objective CH selection, evaluated against state-of-the-art methods using stability and performance metrics.

Chapter 4 - Performance Evaluation of the Modified Incentive-Based Clustering Framework presents an enhanced framework incorporating an ANB-I mechanism for reliable CH selection, improving cluster maintenance, reliability, and CH stability through dynamic vehicular behaviour, and evaluated using throughput, PDR, end-to-end delay, communication overhead, and CH lifetime against existing methods.

Chapter 5 - Reliable and Secure Communication in VANETs using Graph-Based Deep Learning presents a Knowledge Graph-based Deep Convolutional Neural Network deployed at the RSU for intelligent intrusion detection, integrating graph-based feature representation with deep learning to identify malicious activity while preserving reliable communication, validated on benchmark intrusion datasets using accuracy, precision, recall, F1-score, and reliability metrics.

Chapter 6 - Conclusion, Future Work and Social Impact summarizes the contributions, maps them to the research objectives, discusses limitations such as the computational overhead of tensor operations in dense networks, the social impact on road safety, and directions for future research in 6G-enabled vehicular networks.

CHAPTER 2

LITERATURE REVIEW

71 Clustering is one of the most effective network-organization techniques for improving scalability, communication efficiency, and stability in distributed systems. By partitioning nodes into logical groups, it transforms a flat topology into a hierarchy that reduces routing complexity, minimizes control overhead, and eases resource management, which is why it has been widely adopted in wireless sensor networks, mobile ad hoc networks, and Vehicular Ad Hoc Networks, where stable communication under dynamic conditions remains a fundamental challenge [15], [28], [32].

8 In VANETs, clustering supports Intelligent Transportation Systems by enabling efficient Vehicle-to-Vehicle and Vehicle-to-Infrastructure communication for safety-critical and traffic-management applications such as collision avoidance, emergency message dissemination, cooperative driving, and congestion mitigation. However, high mobility, rapidly changing topology, heterogeneous density, and diverse driving behaviours frequently disrupt links, causing unstable clusters, signalling overhead, and degraded reliability [18], [34], [24]. Cluster Head selection is widely recognized as the primary factor governing cluster stability and performance, since the CH coordinates intra-cluster communication, manages membership, and supports inter-cluster exchange; an unstable CH leads to frequent re-clustering, reduced lifetime, packet loss, and higher delay. Recent studies therefore incorporate mobility, connectivity, link lifetime, trust, and reliability into CH selection to improve long-term stability [10], [35], [41].

Existing methodologies have evolved from conventional distance- and mobility-based approaches to hypergraph-based, spectral, and learning-based techniques. Although these improve performance, challenges of dynamic topology adaptation, objective CH selection, communication reliability, and secure dissemination remain largely unresolved [16], [42], [43], [44]. This chapter therefore critically reviews state-of-the-art clustering methodologies, CH-selection mechanisms, reliability models, and graph-

based intelligent techniques, forming the foundation for the stability-aware, reliability-driven, and secure VANET frameworks proposed in this thesis [15], [16], [41].

2.1 Clustering Methodologies in VANET

Clustering organizes VANETs into logical groups to improve scalability, efficiency, and stability. Rather than a flat architecture in which every vehicle independently routes and communicates, clustering partitions vehicles by mobility, connectivity, or similarity, reducing routing complexity and control-message exchange while enhancing resource utilization in dynamic environments [28], [34], [36]. As density grows, flat routing suffers excessive overhead, frequent topology updates, and poor scalability from the continuous forming and breaking of links, making clustering a fundamental strategy for reliable large-scale VANET communication [18], [21].

A key advantage is mitigating the broadcast-storm problem in dense traffic, where many vehicles rebroadcast safety messages, increasing contention, collisions, delay, and bandwidth use. By restricting dissemination within cluster boundaries, clustering localizes communication, reduces retransmissions, and improves channel utilization [21], [22], [28]. Each cluster has ordinary Cluster Members and an elected CH acting as local coordinator for membership, topology maintenance, data aggregation, and routing [10], [35].

Clustering supports two complementary modes. Intra-cluster communication is the exchange among members under CH supervision, where the CH schedules channel access, maintains synchronization, and coordinates safety-message dissemination; selecting the CH by mobility, connectivity, and link lifetime keeps the cluster stable despite movement [18], [34], [35]. Inter-cluster communication lets neighbouring CHs exchange aggregated information with other clusters or nearby Roadside Units, which extend coverage beyond a single vehicle's range and support large-scale Internet of Vehicles (IoV) services [24], [30], [36]. The overall effectiveness of a framework therefore depends on efficient cluster generation and stable CH selection, which together ensure long-lived clusters, reliable communication, and low maintenance overhead [15], [41], [43], [44].

2.2 Classification of Clustering Techniques

VANET clustering techniques can be classified by mobility, connectivity, trust, learning models, and graph representations. This thesis organizes the literature into four representative categories, distance-based, mobility-based, trust-based, and spectral-based clustering, as they encompass the methodologies most closely related to the proposed research, enabling systematic comparison of cluster formation, CH selection, and stability before introducing the proposed hypergraph-based, reliability-aware frameworks.

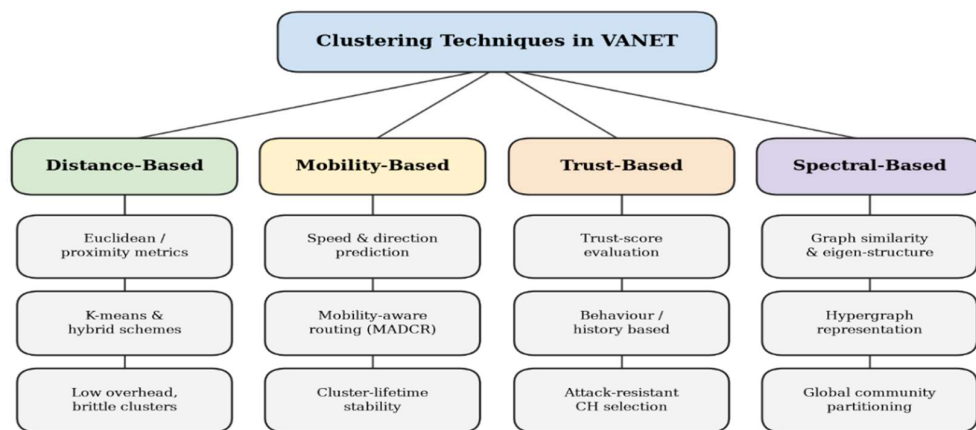


Figure 2.1. Classification of clustering techniques in VANET.

2.2.1 Mobility-Based Clustering

Mobility-based clustering recognizes that node position evolves over time, shifting proximity from static to predictive. By correlating cluster structure with expected link duration, these designs aim to extend cluster lifetime and reduce reconfigurations, incorporating relative velocity, direction, position variation, and link stability [18], [21], [22], [31]. Stability is often quantified from relative velocity and displacement to estimate link lifetime, enabling more reliable CH selection [18], [22]. However, accuracy depends strongly on the mobility model; prediction errors degrade stability estimation under heterogeneous traffic [31], [45]. These schemes work best over short horizons and reduce unnecessary reformations, but remain sensitive to measurement noise, sudden acceleration, and unpredictable driving [21], [22], [45]. Optimization-driven models such as IoVSSA further enhance mobility-aware clustering at the cost

of added computational complexity [46]. Table 2.1 compares representative mobility-based schemes.

Table 2.1. Comparative Analysis of mobility-based clustering schemes

Ref.	Year	Methodology	Mobility Metrics	Strengths	Limitations
Abdel-Halim et al. [31]	2019	Mobility prediction-based clustering (CAVs)	Future position, relative movement, mobility prediction, link stability	Extends cluster lifetime via prediction	Depends on prediction accuracy under dynamic traffic
Sennan et al. [21]	2021	Mobility-aware dynamic clustering routing (MADCR)	Mobility, cluster stability, PDR, delay	Improves routing reliability and efficiency	Focuses on routing rather than intelligent CH selection
Mukhtaruzza man et al. [22]	2021	Junction-based stable clustering	Relative velocity, displacement, LLT, junction mobility	Handles intersection mobility, improves stability	Evaluated mainly at urban junctions
Salim et al. [46]	2023	IoVSSA mobility-aware clustering	Mobility parameters, cluster quality, overhead	Optimizes formation, reduces overhead	Higher complexity from optimization
Mukhtaruzza man et al. [45]	2023	Feedback-based predictive clustering	Mobility feedback, prediction accuracy, link stability, lifetime	Improves maintenance via predictive feedback	Feedback increases processing overhead

2.2.2 Distance-Based Clustering

Distance-based clustering is one of the earliest approaches, valued for simplicity and low cost. It assumes vehicles within a geographical region share similar communication conditions and can be grouped together, forming clusters by Euclidean distance to the CH within a predefined range [28], [32], [47]. Its major limitation is that spatial proximity does not guarantee future stability: nearby vehicles may lose their link quickly due to differences in velocity and direction, and vehicles moving in opposite directions may satisfy distance constraints yet leave range rapidly, causing reconstruction and reduced lifetime [18], [47]. Researchers therefore added connectivity, density, and quality indicators; connectivity-based clustering improves robustness through neighbourhood relationships [28], though combining parameters increases complexity and tuning effort.

Spatial techniques such as K-means have been applied for efficiency [48], [49], but require a prior cluster count and assume spherical clusters, limiting use on irregular roads. Distance-oriented approaches remain sensitive to heterogeneous mobility, causing unstable formation and frequent CH changes [33], [43]. Thus, while efficient for initial grouping, their inability to capture mobility dynamics and complex relationships motivates advanced approaches based on mobility prediction, trust, and graph-based learning [33], [43]. Table 2.2 provides a comparative analysis.

Table 2.2. Comparative Analysis of distance-based clustering schemes

Ref.	Year	Methodology	Strengths	Limitations
Ucar et al. [32]	2015	Multi-hop cluster-based IEEE 802.11p/LTE hybrid, proximity grouping	Improves message dissemination by grouping nearby vehicles	Focuses on efficiency; mobility instability and CH selection not addressed
Khan et al. [28]	2021	Connectivity-based clustering via neighbourhood relationships	Enhances connectivity and reliability using neighbours	Unstable under high mobility and rapid topology change

Ref.	Year	Methodology	Strengths	Limitations
Elaraby and Abuelenin [47]	2021	Graph-theory connectivity analysis for directed highway networks	Analytical understanding of connectivity relationships	Analyzes connectivity only; no adaptive clustering or CH selection
Gbadoubissa et al. [48]	2020	K-means clustering via distance-based partitioning	Reduces complexity, improves efficiency	General mobile networks; ignores VANET-specific mobility
Kandali et al. [49]	2021	Modified K-means integrated with routing optimization	Improves routing via efficient clusters	Needs predefined parameters; degrades with high mobility
Pal et al. [50]	2018	Adaptive mobility- and range-based clustering MAC	Combines range and mobility to improve efficiency	Range decisions sensitive to sudden mobility/density change

2.2.3 Trust-Based Clustering

Trust-based clustering recognizes that stability depends not only on mobility but also on node trustworthiness. A VANET may stay topologically connected yet become functionally unreliable if malicious or selfish vehicles are elected as CHs, since they can launch denial-of-service attacks, drop safety messages, spread false information, or disrupt routing [24], [35], [51]. Trust-aware clustering therefore integrates security with formation by evaluating behavioural reliability before CH selection.

Most approaches estimate a trust score from indicators such as packet-forwarding ratio, message consistency, neighbour recommendations, historical interactions, and reliability [24], [29], [35], [52]. Higher-trust vehicles are preferred as CHs, reducing the chance of selecting malicious nodes, and recent work embeds trust into routing and secure-communication frameworks to improve stability and resilience to internal attacks [41], [52], [53], with combined trust, mobility, and connectivity further enhancing stability and security [54], [55]. However, trust management adds computational and communication overhead: centralized frameworks such as

StabTrust use RSUs to maintain consistent trust but create bottlenecks and single points of failure [24], whereas decentralized estimation scales better but is vulnerable to inaccurate recommendations and manipulation [29], [35]. Trust-based clustering is thus most effective when combined with mobility- and stability-aware mechanisms [29], [35]. Table 2.3 compares representative schemes.

Table 2.3. Comparative Analysis of trust-based clustering schemes.

Ref.	Year	Methodology	Trust Metrics	Strengths	Limitations
Awan et al. [24]	2020	StabTrust centralized trust-based clustering	Trust score, reputation , centralized trust	Stable clusters and reliable communication via RSUs	Centralized design adds overhead and single point of failure
Garg et al. [52]	2021	TruClu trust-based clustering for SDVN	Trust score, controller-assisted CH selection	Improves secure formation and attack resilience via SDN	Depends on SDN controller availability
Singh et al. [29]	2023	Trust-based clustering for stable CH selection	Trust score, mobility, reliability, lifetime	Improves lifetime and reliability via trustworthy vehicles	Limited validation at large scale
Kadam et al. [41]	2023	TACR trust-aware clustering-based routing	Trust value, routing reliability, PDR, delay	Enhances delivery, reliability, secure communication	Optimized for routing; long-term stability limited
Vengala et al. [54]	2025	TCESS trust-score clustering for security/stability	Consensus/behavioral trust, malicious detection	Resilience to malicious nodes with stable clusters	Accuracy depends on behavioural observations

Ref.	Year	Methodology	Trust Metrics	Strengths	Limitations
Awan et al. [55]	2025	Trust-based clustering with wireless energy sharing	Trust score, energy efficiency, reliability	Improves trustworthiness and energy-aware communication	Focuses on energy sharing; general applicability unproven

2.2.4 Spectral-Based Clustering

Spectral-based clustering is a powerful graph-theoretic framework that overcomes limitations of distance- and mobility-based methods. Unlike approaches relying on local information such as distance, range, or relative mobility, it exploits the global topological structure of the communication graph [5], [16], identifying coherent vehicle communities even when boundaries are irregular, overlapping, or dynamic, which suits highly dynamic environments where density, connectivity, and mobility evolve continuously [5], [56]-[60]. The network is represented as a weighted graph $G = (V, E, W)$, where V is the set of vehicles, E the links, and W the weighted adjacency (similarity) matrix whose entries quantify similarity from attributes such as distance, relative velocity, direction, connectivity duration, or quality, commonly via a Gaussian kernel. From W the diagonal degree matrix D is formed, and the graph Laplacian $L = D - W$ (or a normalized form for large networks) captures the structure while reducing the influence of heterogeneous node degrees.

Spectral clustering partitions the graph to maximize intra-cluster and minimize inter-cluster similarity by eigenvalue decomposition of the Laplacian, $Lv = \lambda v$. The smallest eigenvalues describe dominant connectivity patterns, and their eigenvectors give a low-dimensional embedding preserving global structure, so vehicles with similar coordinates are grouped even when not geographic neighbours, enabling detection of non-convex clusters and long-range relationships. The eigengap heuristic further estimates the optimal number of clusters from the largest gap between consecutive eigenvalues, removing the need to fix the count a priori as in K-means.

Practical VANETs exhibit many-to-many interactions that pairwise graphs cannot represent. Hypergraph-based spectral clustering, $H = (V, E_h, W_h)$, lets each hyperedge connect multiple vehicles, capturing group communication naturally; hypergraph Laplacians preserve higher-order neighbourhood information and improve partitioning under dense traffic. Modularity optimization additionally evaluates whether observed interactions exceed those expected in a random network, producing partitions with dense intra-cluster and sparse inter-cluster links to enhance stability. After embedding, clustering is done in the eigenspace, where Gaussian Mixture Models (GMMs) offer a flexible probabilistic alternative to K-means, fitting clusters of varying size, orientation, and density via Expectation-Maximization.

Despite superior quality, spectral clustering is computationally demanding: building the similarity matrix needs $O(n^2)$ memory and full eigen-decomposition is costly, so scalability is challenging at metropolitan scale. Research therefore pursues sparse representations, approximate eigensolvers, hypergraph optimization, and probabilistic clustering to cut overhead while preserving quality [5], [16], [56]-[60]. Overall, by combining weighted similarity matrices, Laplacians, eigenvalue decomposition, modularity, and hypergraph representations, spectral methods yield clusters that are more stable, coherent, and resilient than conventional techniques, though real-time large-scale deployment remains an open challenge. Table 2.4 compares representative spectral-based schemes.

Table 2.4. Comparative Analysis of spectral-based clustering schemes.

Ref.	Year	Methodology	Stability Metrics	Strengths	Limitations
Liu et al. [16]	2020	Spectral clustering via graph Laplacian and similarity matrix	Connectivity, stability, link quality	Stable, well-separated clusters from global topology	Eigen-decomposition costly for large VANETs

Ref.	Year	Methodology	Stability Metrics	Strengths	Limitations
Radhika and Bhuvaneswari [58]	2021	DL-assisted spectral clustering (Ruzicka + Dragonfly)	Spectral similarity, routing reliability, PDR	Improves PDR and routing reliability	Multi-stage optimization adds overhead
Mohamed et al. [56]	2023	Spectral clustering with Deep Reinforcement Learning	Spectral similarity, resource use, throughput	Adaptive resource allocation, higher throughput	Focuses on resource allocation, not stable formation
Feng et al. [57]	2024	Spectral clustering-based RSU deployment	Connectivity, coverage, deployment efficiency	Optimizes RSU placement via graph connectivity	Addresses deployment, not vehicle clustering
Hu et al. [5]	2019	Spectral clustering with Dirichlet Process GMM	Graph similarity, community structure	Auto-determine cluster count without fixing k	General networks; needs mobility-aware similarity for VANETs

2.3 Cluster Formation and Maintenance Strategies

Cluster formation and maintenance are the two phases determining the efficiency, stability, and scalability of clustering-based VANETs. Formation determines the optimal number of clusters (k) and assigns vehicles by similarity or stability criteria [32], [36], [61], directly influencing lifetime, overhead, and routing [18], [22]. Too many clusters increase CH elections and handoff signalling, while too few create oversized clusters with intra-cluster contention and delay [32], [33], [43]. Conventional algorithms like K-means need k a priori [48], [49], unsuitable for

changing densities, so spectral methods use the eigengap heuristic to estimate k from Laplacian eigenvalues [5], [16], producing adaptive, balanced clusters; hypergraph approaches further improve formation by modelling higher-order relationships under dense traffic [10], [25], [36].

Maintenance then manages topology changes from mobility, including node joining, leaving, merging, and splitting [18], [21], [31], continuously monitoring stability via mobility, connectivity, and neighbourhood information and exchanging beacons to update membership [22], [45], [46]. Several studies add secondary or backup CHs to replace a departing primary immediately [32], [62], while mobility prediction and feedback-based strategies improve continuity by choosing CHs with longer expected link lifetimes [31], [45]; junction-aware and mobility-aware routing protocols improve lifetime, PDR, and reliability in urban traffic [21], [22]. Recent work integrates trust management, optimization, and intelligent CH selection into maintenance to jointly improve reliability and stability [24], [29], [35], [52], with trust-aware mechanisms excluding malicious CHs [35], [41], [52] and metaheuristic/multi-criteria methods selecting more stable CHs [46], [54], [55]. Nonetheless, maintaining long-term stable clusters in large urban VANETs remains challenging given the trade-off among accuracy, complexity, overhead, and rapidly changing traffic [16], [25], [61].

2.4 Cluster Head Selection and Reliability Studies

CH selection is among the most critical clustering phases, determining cluster management effectiveness and communication continuity. A reliable CH sustains stable links, coordinates members efficiently, and minimizes reorganization under dynamic traffic. Since the CH coordinates forwarding, management, and inter-cluster communication, its selection directly affects reliability, lifetime, and stability [10], [16], [22].

Studies evaluate diverse parameters including mobility, neighbourhood connectivity, graph structural properties, trustworthiness, and quality [16], [25], [29], [32]. Rather than a single metric, recent approaches use multi-criteria decision models to balance attributes and improve robustness [10], [25], [58]; hypergraph and spectral methods exploit higher-order connectivity, while trust-aware approaches incorporate behaviour

into the decision [16], [25], [29]. Yet most mechanisms depend on predefined weighting, heuristic rules, or application-specific criteria that adapt poorly to changing environments. Khan et al. (2026) [62] improved stability with a multi-criteria framework using Connectivity Lifespan and Past Cluster Head Lifetime, but still relies on weighted decision-making. Reliable CH selection with minimum re-election overhead thus remains open. Table 2.5 compares representative CH-selection studies.

Table 2.5. Comparative Analysis of representative CH-selection studies

Ref.	Year	Methodology	Strengths	Limitations
Liu et al. [16]	2020	Spectral clustering based on graph Laplacian	Stable partitions, less frequent reformation	Eigenvalue computation raises complexity
Khayat et al. [2]	2020	Weighted trust- and mobility-based CH selection	Enhances reliability via trust and mobility	Continuous trust evaluation adds overhead
Saleem et al. [23]	2021	Deep learning-based dynamic CH selection	Improves stability and lifetime	High computational cost and training complexity
Jabbar et al. [10]	2022	Hypergraph Clustering Model (HGCM)	Captures higher-order relationships, improves stability	High computation and memory for large hypergraphs
Singh et al. [29]	2023	Trust-based clustering approach	Improves lifetime via trustworthy CHs	Limited validation at large scale

2.4.1 Cluster Head Selection Parameters and Decision Criteria

CH selection is critical because communication stability depends on the selected CH, which coordinates members, maintains connectivity, manages forwarding, and supports inter-cluster communication; an unsuitable choice causes frequent leadership changes, signalling overhead, packet loss, and reduced performance [32], [33], [36]. Studies therefore develop selection strategies spanning mobility, connectivity, reliability, security, and resource availability.

Initial approaches used mobility and geographical characteristics, adopting relative velocity, position, inter-vehicle distance, and link duration because they directly affect lifetime and continuity [18], [22], [31]; however, these depend on accurate prediction and degrade under sudden acceleration, lane changes, and heterogeneous traffic [31], [45]. Later methods added connectivity parameters, including neighbourhood degree, density, and quality, to improve stability and reduce re-election [25], [34], [36].

With growing complexity, multi-criteria CH selection gained attention, evaluating mobility, connectivity, eccentricity, link reliability, and trust together. Spectral methods analyse global network structure rather than only local characteristics; Liu et al. used connectivity patterns to improve stability, and other spectral studies added structural parameters such as eccentricity [5], [16], [58], while hypergraph approaches model higher-order relationships for dense interactions [10], [15], [36]. Security-aware selection incorporates trust, considering direct/indirect trust, forwarding behaviour, and history to exclude unreliable vehicles [24], [29], [35], [52]; StabTrust used centralized trust to improve stability but adds overhead and infrastructure dependence [24], and trust-aware routing improves security at higher computational cost [41].

Reliability of CH selection is evaluated via CH stability duration, cluster lifetime, CH change rate, PDR, throughput, end-to-end delay, and routing reliability, indicating how well a CH maintains continuity under dynamic conditions [33], [42], [63]. Table 2.6 compares representative CH-selection parameters and performance metrics.

Table 2.6. Comparative Analysis of CH selection parameters and performance parameters.

Ref.	Year	Methodology	CH Selection Parameters	Performance Parameters
Mukhtaruzzaman and Atiquzzaman [22]	2021	Fast randomized clustering	Vehicle ID, CH ID, relative speed, position	CM lifetime, CH-leaving events, re-election, reaffiliation
Liu et al. [16]	2020	Spectral clustering with	Positive net force for CH leadership	Average cluster lifetime and size

Ref.	Year	Methodology	CH Selection Parameters	Performance Parameters
		force-directed algorithm		
Jabbar et al. [10]	2022	Hypergraph-based clustering model	Speed, location, neighbourhood degree, eccentricity, trust	CH stability, cluster number, CM lifetime, change rate, delay, throughput
Dutta et al. [42]	2021	Deep-learned spectral clustering routing	Node density, direction, distance, velocity, residual energy, bandwidth	PDR, drop ratio, delay, throughput, link stability
Radhika and Bhuvaneswari [58]	2021	Spectral clustering-based routing	Similarity, mobility, distance, velocity	Energy, PDR, delay, throughput
Singh et al. [29]	2023	Trust-based clustering	Trust value, node degree, position, travel time, speed	CH duration, state changes
Kaur et al. [64]	2025	CH selection metrics (speed, stability, centroid, hybrid)	Speed, stability, centroid position, hybrid metric	Throughput, PDR, delay, CH stability across conditions
Chiang et al. [65]	2026	Candidate Cluster Head Nomination Algorithm (CCHNA)	Mobility, number of neighbours, distance, CH nomination	Number of CHs, lifetime, overhead, maintenance efficiency
Khan et al. [62]	2026	Multi-Criteria Stable Clustering (MCSC)	Relative speed, connectivity, re-lifespan, node degree, PCL, CCF	CH/CM lifetime, stability, re-clustering rate, control overhead

The analysis shows CH selection has evolved from single-parameter strategies toward intelligent multi-criteria frameworks. Yet limitations remain: many methods use predefined weights or heuristic rules that misrepresent the changing importance of stability parameters across traffic conditions [25], [36], [52], and security-oriented approaches often treat trust separately rather than integrating reliability assessment with adaptive selection. An objective, adaptive CH-selection mechanism that evaluates multiple stability attributes while reducing parameter bias thus remains an open challenge.

2.5 Intelligent and Learning-Based Reliability Enhancement Mechanisms

Cryptographic authentication and encryption are often insufficient for highly dynamic VANETs because they protect confidentiality and authenticity but cannot detect insider attacks, compromised nodes, or evolving intrusion patterns [38], [39], [66]. Recent research therefore adopts Machine Learning (ML) and Deep Learning (DL) to identify malicious activity through traffic analysis and behavioural learning [57], [58]. Early ML-based intrusion detection used Decision Trees, Random Forests, and Bayesian learning, while advanced DL models such as Deep Belief Networks and hybrid architectures achieved higher accuracy by learning complex nonlinear feature relationships [40], [67], [68].

More recently, graph-based deep learning models the relational structure of vehicular communication. Knowledge graphs represent vehicles, RSUs, packets, and events as interconnected nodes for semantic reasoning [26], [38]; building on this, the Knowledge Graph-based Deep Convolutional Neural Network integrates graph representations with convolutional learning to capture structural dependencies and attack characteristics, improving detection of sophisticated coordinated attacks [26]. Multi-channel CNN architectures and ensemble learning further boost accuracy and robustness across benchmarks such as ToN-IoT and UNSW-NB15 [39], [40], [68]. Table 2.7 summarizes representative learning-based security approaches.

Table 2.7. Comparative Analysis of representative learning-based VANET-security methods

Ref.	Year	Methodology	Strengths	Limitations / Research Gap
Prabhakaran and Kulandasamy [69]	2021	Hybrid semantic deep learning architecture	Learns nonlinear relationships, improves detection	Proposed for cloud; needs adaptation for dynamic VANETs
Lo et al. [70]	2022	E-GraphSAGE GNN-based IDS	Exploits graph relationships for accuracy	Built for IoT; needs mobility-aware graph modelling
El-Dalahmeh and Adeel [66]	2023	Deep Belief Network with Decision Tree	Hybrid DL/DT improves detection	SDN-focused; overhead grows with network size
Park et al. [71]	2023	Graph-based IDCS (G-IDCS)	Graph representations for accurate classification	Designed for CAN, not inter-vehicle communication
Wang et al. [72]	2025	BS-GAT graph attention network IDS	Captures complex graph dependencies	High complexity and scalability challenges
Wang et al. [73]	2024	Attribute-based access control with IDS	Integrates detection with secure sharing and privacy	Focuses on access control, not clustering reliability
Song et al. [74]	2024	Dynamic graph-based IDS (DGIDS)	Models' dynamic entity relationships for detection	Evaluated on in-vehicle CAN; limited large-scale validation
Karthiga et al. [75]	2022	ML and DL-based intrusion detection for VANETs.	High attack detection accuracy and improved network security.	Lacks graph-based feature learning, semantic modeling, and communication reliability analysis.

2.6 Comparative Analysis and Research Gaps

2.6.1 Critical Discussion of Comparative Tables

The analysis indicates that no single clustering technique simultaneously satisfies stability, scalability, computational efficiency, and security in highly dynamic VANETs; each paradigm optimizes a specific objective with inherent trade-offs [61], [16]. As shown in Table 2.1, mobility-based clustering extends lifetime using relative velocity, trajectory, and prediction [18], [22], [31] and reduces reformation [21], [45], but depends on prediction accuracy that deteriorates under sudden lane changes [31], [46]. Per Table 2.2, distance-based clustering offers low complexity via distance and range [48], [49], yet proximity alone cannot ensure stable links, and K-means needs a predefined count with simple shapes [25], [33], [34].

Per Table 2.3, trust-based clustering improves reliability by selecting trustworthy CHs and mitigating malicious participation [24], [35], [52], though continuous trust evaluation adds overhead, especially in centralized designs [39], [51], [55]. Spectral-based clustering (Table 2.4) provides superior stability using graph similarity and eigenvalue decomposition to capture global topology [5], [16], with hypergraph extensions modelling higher-order relationships [10], [25], but incurs higher complexity from similarity-matrix construction and eigenvalue computation [56], [58]. Tables 2.5, 2.6, and 2.7 show the evolution of CH selection and learning-based security: recent CH schemes improve stability with mobility, connectivity, trust, and spectral information but mostly use heuristic or predefined weighting [16], [33], [35]; existing methods rarely evaluate parameter importance objectively, causing frequent CH changes [10], [25], [52]; and learning-based techniques detect intrusions effectively but are developed independently of clustering and CH reliability [38], [39], [66]. Overall, existing approaches address mobility, trust, topology, or security individually with limited integration [16], [25], [61], motivating an intelligent hypergraph-based framework unifying stable clustering, objective CH selection, and secure communication.

2.6.2 Research Gaps Identification

A critical analysis of the literature reviewed in this chapter reveals the following research gaps that this thesis addresses:

- ❖ Ineffective management of cluster transitions. Most schemes focus on formation but pay limited attention to vehicle transitions between clusters (handovers), which are critical for stability; frequent changes during transitions raise re-clustering frequency and overhead and reduce reliability [18], [20], [21], [22].
- ❖ Difficulty in handling multi-vehicle relationships. Traditional graph-based clustering represents only pairwise connections, limiting its ability to capture the many-to-many interactions in dense traffic and to model group-wise relationships for stable, scalable formation [10], [15], [17], [25].
- ❖ Determination of the optimal number of clusters. Existing methods do not guarantee the optimal count across varying scenarios, relying on predefined or heuristic numbers that cause unbalanced formation, reduced stability, and higher overhead [5], [9], [16].
- ❖ Need for a stable cluster head selection approach. Multi-attribute CH selection commonly uses weighted-sum methods whose weights are predefined or expert-dependent, so the relative importance of mobility, connectivity, link stability, and reliability cannot be evaluated objectively under changing conditions [2], [9], [22], [24], [29].
- ❖ Mismanagement of vehicle density in dynamic environments. Existing approaches adapt poorly to continuously varying densities and mobility, causing unstable formation, frequent maintenance, and reduced efficiency, so maintaining stable clustering with reliable and secure communication remains a significant challenge [9], [16], [24], [26].

2.6.3 Alignment of Proposed Research with Identified Gaps

Motivated by these challenges, this thesis proposes an intelligent, adaptive, reliability-oriented framework for enhancing communication in highly dynamic VANETs. It uses

hypergraph-based spectral clustering to capture complex multi-vehicle interactions beyond traditional graphs; the integration of iTTM with GMM enables stable, efficient formation, while the eigengap heuristic adaptively determines the optimal number of clusters. The weighted CRITIC-based CH selection objectively evaluates multiple stability parameters, improving reliability and continuity, and the KG-DeepCNN framework enhances security by detecting malicious patterns at the RSU. Collectively, these contributions address stable clustering, reliable CH selection, and secure communication, providing a scalable framework for next-generation intelligent vehicular networks.

The review identifies gaps in cluster formation, CH selection, reliability, and security in dynamic VANETs, and the research defines objectives aligned with them. Table 2.8 maps the identified gaps to the research objectives.

Table 2.8. Mapping of Research Gaps to Research Objectives

Research Gap	Research Objective	How the Objective Addresses the Gap?
Difficulty in handling multi-vehicle relationships	RO1: Study and analyse clustering algorithms	Study of existing approaches identifies limitations and motivates spectral clustering.
	RO2: Develop scalable spectral clustering algorithms	Hypergraph representation captures multi-vehicle interactions for realistic topology modelling.
Ineffective management of cluster transitions	RO2: Develop intelligent scalable spectral clustering	Hypergraph-based modified spectral clustering models multi-vehicle relations and reduces transitions via more stable clusters.
	RO3: Implement stability-based model for comparison	Stability metrics (relative speed, neighbourhood degree, link lifetime) maintain stable membership and minimize switching.
Determination of the optimal number of clusters	RO2: Develop intelligent spectral clustering	Similarity, modularity, and Laplacian matrices automatically determine the optimal cluster count.

Research Gap	Research Objective	How the Objective Addresses the Gap?
	RO3: Implement stability-based model	Stability-driven clustering keeps structures optimal under varying mobility and density.
Need for stable cluster head selection	RO3: Implement efficient stability-based model	Adaptive stability metrics select stable CHs, reducing frequent switching.
	RO4: Analyze reliability of cluster head	Reliability-oriented evaluation ensures selected CHs sustain stable performance under dynamic conditions.
Mismanagement of vehicle density in dynamic environments	RO3: Implement stability-based model	Framework evaluates performance under sparse and dense densities for scalability.
	RO4: Analyze reliability of proposed models	Reliability analysis under multiple scenarios confirms stability and efficient communication.

The mapping demonstrates that each objective is formulated to address a specific gap identified through the review. These objectives are realized through a series of published articles that collectively support the development, implementation, and validation of the proposed methodologies. Table 2.9 maps the research objectives to the corresponding publications.

Table 2.9. Research Objectives mapping with the publications

Research Objective	Publications
RO1: To study and analyze the different clustering algorithms in VANETs.	A. Kumari, S. Kumar, and R. S. Raw, "Investigating the Reliability of Vehicle-to-Vehicle and Vehicle-to-Infrastructure Communication," in Proc. Int. Conf. Innovations in Computational Intelligence and Computer Vision, Singapore: Springer, 2024, pp. 483-500. https://doi.org/10.1007/978-981-97-6992-6_36 (SCOPUS INDEXED). A. Kumari, S. Kumar, and R. S. Raw, "Clustering in Vehicular Ad Hoc Network: A Bibliometric Analysis," SN Computer Science, vol. 6, no. 2, 2025. https://doi.org/10.1007/s42979-024-03638-y (SCOPUS)

Research Objective	Publications
RO2: To develop intelligent algorithms based on scalable spectral clustering algorithms.	A. Kumari, S. Kumar, and R. S. Raw, "Advancing VANET stability: enhanced cluster head selection with iTTM and weighted CRITIC," <i>Journal of Supercomputing</i> , vol. 80, pp. 16133-16172, 2024. https://doi.org/10.1007/s11227-024-06049-1 (SCIE, IF-2.7)
RO3: To implement an efficient stability-based model to compare with recent state-of-the-art works.	A. Kumari, S. Kumar, and R. S. Raw, "Advancing VANET stability: enhanced cluster head selection with iTTM and weighted CRITIC," <i>Journal of Supercomputing</i> , vol. 80, pp. 16133-16172, 2024. https://doi.org/10.1007/s11227-024-06049-1 (SCIE, IF-2.7)
RO4: To analyze the reliability of the cluster head of the proposed models.	A. Kumari, S. Kumar, and R. S. Raw, "Modified clustering and incentivized stable CH selection for reliable VANET communication," <i>Cluster Computing</i>, vol. 27, pp. 11983-12005, 2024. https://doi.org/10.1007/s10586-024-04559-4 (SCIE, IF-3.6) A. Kumari, S. Kumar, "Mitigating intrusion attacks in VANETs through reliable communication with graph-based deep learning approaches," <i>Journal of Supercomputing</i>, vol. 81, 1467, 2025. https://doi.org/10.1007/s11227-025-07945-w (SCIE, IF-2.7) A. Kumari, S. Kumar, and R. S. Raw, "Incentivized-based Adaptive Neighbour Behaviour Approach for Vehicular Adhoc Networks," in 6th Int. Conf. <i>Data Science and Applications (ICDSA 2025)</i>, July 16-18, 2025. https://doi.org/10.1007/978-3-032-10756-5_16 (SCOPUS INDEXED)

The mapping demonstrates that each publication addresses one or more research objectives, ensuring a systematic and comprehensive realization of the proposed research framework.

2.7 Chapter Summary

This chapter provided a comprehensive review of state-of-the-art clustering methodologies, CH selection strategies, and intelligent reliability-enhancement techniques for VANETs. The literature was systematically classified into mobility-based, distance-based, trust-based, and spectral-based clustering, highlighting their

contributions to stability, efficiency, and scalability, and existing CH selection and security frameworks were critically examined for their effectiveness under dynamic conditions.

The comparative analysis showed that, despite considerable advances, existing studies mostly address cluster formation, CH selection, reliability, and security as separate problems. Consequently, adaptive cluster stability, objective CH selection, higher-order interaction modelling, and the integration of reliability with intelligent security remain insufficiently addressed. These findings identified the key research gaps and their alignment with the thesis objectives. The insights from this chapter provide the conceptual and technical foundation for the proposed research; the next chapter presents a hypergraph-based clustering framework with an objective CH-selection mechanism to enhance stability, reliability, and scalability in highly dynamic VANET environments.

CHAPTER 3

OPTIMIZED CLUSTER HEAD SELECTION USING THE iTTM-WEIGHTED CRITIC FRAMEWORK

3.1 Introduction

Vehicular Ad hoc Networks have emerged as the communication backbone of intelligent transportation systems, enabling vehicles to exchange safety, traffic, and infotainment information with one another and with the roadside infrastructure. By allowing timely dissemination of hazard warnings, congestion alerts, and cooperative-awareness messages, VANETs directly contribute to road safety and to the realisation of Advanced Driver Assistance Systems (ADAS). However, the very property that makes VANETs attractive, namely the mobility of the nodes, also makes the network topology exceptionally volatile. Vehicles join and leave communication range within seconds, link qualities fluctuate rapidly, and the density of nodes varies dramatically between sparse suburban roads and congested urban intersections. Managing communication in such an environment is a persistent research challenge, and clustering has become one of the most widely adopted strategies to impose a manageable, scalable structure on the network [76], [77].

In a clustered VANET, vehicles are grouped into logical partitions, and within each partition a single vehicle is elected as the cluster head. The CH coordinates intra-cluster communication, aggregates data from its cluster members, and acts as the relay towards the roadside unit or towards neighbouring clusters. The quality of the entire clustered architecture therefore hinges on two coupled decisions: how the vehicles are partitioned into clusters, and which vehicle within each cluster is chosen as the CH. If either decision is made poorly, the network suffers from frequent re-clustering, high control overhead, elevated packet delay, and reduced throughput. Consequently, the stability of the cluster head, the ability of an elected CH to retain its role over an extended period without frequent re-election, is the single most important indicator of the health of a VANET clustering scheme, and it is the central concern of this chapter.

3.1.1 Motivation for Stable Cluster Head Selection

The motivation for pursuing a more stable CH selection mechanism arises from three interrelated shortcomings observed in the existing body of work [9]. First, most conventional clustering schemes model the vehicular network as an ordinary graph, in which every edge connects exactly two vehicles. Such a pairwise abstraction is a poor fit for real traffic, where a single vehicle simultaneously interacts with a group of neighbours and where multiple vehicles share overlapping communication regions. Representing these many-to-many relationships through pairwise edges inflates the number of edges, obscures group structure, and ultimately degrades the quality of the resulting clusters.

Second, even among schemes that adopt richer representations, the determination of the number of clusters is frequently left to heuristics that are either fixed arbitrarily or re-computed at every instant. An arbitrarily large number of clusters lowers vehicle density per cluster and inflates control overhead, whereas too few clusters overload each CH and increase transmission delay. A principled, data-driven determination of the optimal cluster count is therefore essential, yet it is often absent. Moreover, the widely used k-means algorithm, when applied to the spectral embedding, implicitly assumes spherical, equally sized clusters and ignores the true distribution of the projected points, an assumption that is routinely violated in road networks whose geometry is elongated and irregular.

Third, cluster head selection typically depends on several vehicle attributes, speed, connectivity, link duration, and trust, that must be combined into a single decision. The prevailing practice is to fuse these attributes through a weighted sum in which the weights are chosen manually by the designer. Such subjective weighting is difficult to justify, does not adapt to changing traffic conditions, and cannot account for the statistical relationships among the criteria. A vehicle that scores well on a highly correlated pair of criteria may be unfairly favoured, while genuinely informative but overlooked criteria are suppressed.

The framework developed in this chapter is motivated by the goal of resolving all three shortcomings within a single, coherent pipeline. The network is represented as a

weighted hypergraph so that group interactions are captured natively; the optimal number of clusters is fixed once, at the instant of maximum vehicle density, using an eigen-gap heuristic; the spectral embedding is partitioned by a Gaussian Mixture Model (GMM) rather than k-means so that non-spherical, probabilistically modelled clusters can be formed; and the multi-attribute CH decision is made objective by deriving the criteria weights automatically through the criteria importance through inter-criteria correlation method. The combined method is referred to throughout this chapter as the improved Tensor-Trace-Maximisation clustering algorithm coupled with weighted CRITIC decision modelling.

3.1.2 Research Scope and Chapter Organisation

The scope of this chapter is confined to the design, mathematical formulation, and simulation-based evaluation of a centralised, RSU-assisted clustering and cluster-head-selection framework for urban VANETs. A centralised architecture is deliberately adopted: because the RSU gathers information from the cluster heads, it maintains a global view of the network and can therefore make more consistent stability-oriented decisions, while also mitigating the synchronisation and coordination problems that afflict fully distributed schemes. The evaluation environment is an urban scenario modelled on Delhi's Connaught Place, reconstructed from OpenStreetMap data and simulated in Simulation of Urban Mobility (SUMO), with the clustering and decision logic implemented in MATLAB and driven through the SUMO TraCI (Traffic Control Interface) interface. Both sparse (100 vehicles) and dense (up to 1000 vehicles) conditions are studied so that scalability can be assessed.

The remainder of the chapter is organised as follows. Section 3.2 establishes the network and system model, including the communication and mobility assumptions and the hypergraph representation of the vehicular network. Section 3.3 develops the mathematical foundation of the clustering framework: the eigen-heuristic determination of the optimal cluster count, the modularity-matrix formulation, and the construction of the Laplacian. Section 3.4 presents the proposed iTTM cluster-formation procedure, detailing the tensor-trace-maximisation objective, its GMM-based partitioning, and the overall spectral clustering process. Section 3.5 formulates the four cluster-head selection criteria, relative cooperative speed, connectivity level,

link lifetime, and consensus trust score, and the weighted CRITIC evaluation used to combine them. Section 3.6 integrates the clustering and decision components into a single optimisation framework and states the cluster-head decision model. Section 3.7 assembles the complete algorithm. Section 3.8 reports the simulation environment, comparative results against state-of-the-art baselines, and a discussion of the principal findings. Section 3.9 summarises the chapter.

3.2 Network and System Modeling

Before the clustering framework can be formulated, the physical and logical characteristics of the vehicular network must be specified. This section describes the communication and mobility assumptions that govern the simulated environment and then introduces the hypergraph abstraction on which the entire clustering pipeline is built.

3.2.1 VANET Communication and Mobility Model

The network consists of a set of mobile vehicles moving over an urban road topology and a set of stationary Roadside Units that provide infrastructure support. Each vehicle is equipped with an on-board unit capable of dedicated short-range communication in accordance with the IEEE 802.11p standard operating in the 5.9 GHz band, where channels of 10 MHz bandwidth (5.850–5.925 GHz) are available for vehicle-to-vehicle and vehicle-to-infrastructure exchanges. Every vehicle carries a spectrum sensor and is aware of its own kinematic state, position, velocity, and heading, obtained from on-board positioning. The transmission range of a vehicle is fixed at 200 metres, while the RSU is assigned a larger transmission range so that it can supervise the vehicles within its coverage region.

A vehicle is considered a neighbour of another vehicle if the Euclidean distance between the two is smaller than the transmission range, so that a reliable one-hop link can be established. The routing model adopted in this work is deliberately restricted to one-hop, direct communication: a cluster member transmits directly to the cluster head, which in turn exchanges messages with the RSU. This choice isolates the effect of clustering and cluster-head stability on the network metrics from the confounding influence of multi-hop routing decisions. Because the network is centralised, all data

points collected from the vehicles are ultimately managed and processed at the RSU on the basis of the information relayed by the cluster heads, giving the RSU the global perspective required for consistent decision-making.

Vehicle mobility is generated using the Simulation of Urban Mobility toolkit. The road network of Delhi's Connaught Place is imported from OpenStreetMap and converted into a SUMO network comprising 293 lanes; random trips are then generated so that vehicles travel from source to destination along realistic routes. The average vehicle speed in the scenario is approximately 10.5 m/s and the maximum permissible speed is 25 m/s. Because signal attenuation in such an environment is a function of relative velocity, the Doppler effect is explicitly accounted for in the channel model; in urban settings, where low-mobility and high-mobility flows coexist, this effect is pronounced and directly influences the reliability of the exchanged information. The simulation proceeds over 1500-time stamps, and at each instant SUMO exports the instantaneous vehicle parameters into MATLAB through the TraCI application programming interface for clustering and cluster-head computation.

3.2.2 Hypergraph-Based Network Representation

An ordinary graph encodes only pairwise relationships: an edge joins precisely two vertices. When a network of cars is modelled in this way, each vehicle is linked to at most two neighbours per edge, and capturing the fact that a vehicle communicates simultaneously with an entire group of surrounding vehicles requires a proliferation of edges. The result is a dense, unwieldy structure whose edge count grows rapidly and which obscures the natural group structure of traffic. A hypergraph overcomes this limitation because a hyperedge may connect any number of vertices at once, and hence a single hyperedge can represent a whole cluster of mutually communicating vehicles [10], [15], [17].

Formally, the urban vehicular network is represented as a weighted, undirected hypergraph

$$\mathcal{H} = (\mathcal{E}, V, w) \quad (3.1)$$

in which $V = \{v_1, v_2, \dots, v_n\}$ is the set of vertices, one per vehicle; $E = \{e_1, e_2, \dots, e_n\}$ is the set of hyperedges, as seen in Figure 3.1, each hyperedge being a non-empty

subset of V that groups together vehicles sharing spatial affinity [36]; and w is a weight function that assigns to every hyperedge a non-negative weight. In the vehicular setting the weight of a hyperedge reflects the strength of the communication coupling among the vehicles it contains and may depend on inter-vehicular distance, signal strength, communication delay, or link reliability. Because the hyperedges have no orientation, the connections they represent are bidirectional, which matches the symmetric nature of one-hop links between vehicles in range.

The representation is undirected because vehicular links are reciprocal, and weighted because not all group interactions are equally significant for stability. The central advantage of the hypergraph model is that it preserves the homogeneity of vertices and hyperedges: relationships between groups of vehicles are expressed directly rather than being decomposed into pairwise fragments. This yields a compact description of overlapping communities, reduces representational complexity, and provides a firmer basis for the partitioning that follows. The specific benefits that motivate the use of a hypergraph in this work are summarised below.

- Vehicles in a dense network naturally form groups on the basis of shared attributes such as speed, direction, or destination; a hyperedge can encode such an overlapping community in a single object rather than through many pairwise edges.
- Because any number of hyperedges may be incident on any number of vertices, situations in which numerous vehicles interact concurrently are modelled faithfully, which in turn reduces congestion in the representation and improves the steadiness of the derived clusters.
- The uniform, flexible treatment of nodes and edges enables a nuanced expression of many-to-many relationships that is unavailable to ordinary graphs, and this is precisely the property required to model vehicular communication accurately.

Any hypergraph may be partitioned into k disjoint subsets that jointly cover the vertex set. If the subsets are denoted e_i , $\bigcup e_i = V$, and $e_i \neq \emptyset, i = 1 \dots k$; the partition satisfies the covering condition that the union of all e_i equals V with every e_i non-empty for $i = 1, \dots, k$, together with the disjointness condition

$\cap e_i k_i = \emptyset$ that the intersection of distinct subsets is empty, while a cost function $f: \mathcal{H} \rightarrow \mathbb{R}$ is minimised over admissible partitions. These conditions formalise the intuitive requirement that every vehicle belongs to exactly one cluster and that the partition is chosen to optimise a stability-oriented objective. The number of clusters k is not assumed in advance; its optimal value is derived in Section 3.3.1, and the cost-minimising partition is produced by the iTTM procedure of Section 3.4.

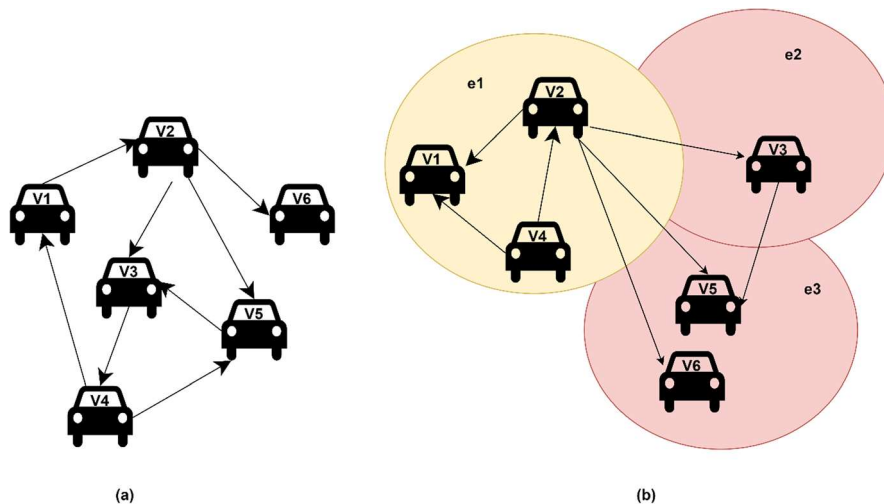


Figure 3.1 VANET hypergraph representation: (a) the vehicular graph and (b) the equivalent hyperedge grouping of mutually communicating vehicles.

3.3 Mathematical Modeling of the Proposed Clustering Framework

This section develops the mathematical machinery on which the proposed clustering procedure rests. Three components are formulated in turn: the modularity matrix that encodes the strength of association between neighbouring vertices, the Laplacian matrix that serves as the operator whose spectrum drives the spectral clustering, and the eigen-heuristic that fixes the optimal number of clusters.

3.3.1 Modularity Matrix Formulation

The partitioning of a network into groups is intimately connected with the notion of modularity, which quantifies the extent to which vertices cluster more densely than would be expected by chance. Whereas the adjacency matrix records which vertices are directly connected, its modularity counterpart is designed to express the strength of the bond between neighbouring vertices relative to a null model, and it is the

modularity matrix that is used to weight the hyperedges in this work. Following the Newman–Girvan formulation [78] adapted to the hypergraph, the entry of the modularity matrix B associated with vertices i and j is defined as:

$$B_{i,j} = -\frac{V_{k_i}V_{k_j}}{2m} \quad (3.2)$$

where V_{k_i} and V_{k_j} denote the degrees of vertices i and j , that is, the total weight of the hyperedges incident on each, and m is the overall edge weight of the hypergraph, given by $m = \frac{1}{2} \sum_i V_{k_i}$. The degree of a vertex therefore aggregates the influence of all the group interactions in which the corresponding vehicle participates.

Interpreted operationally, the modularity matrix assigns greater weight to the hyperedges connecting the most closely associated vertices, so that clustering efficiency is improved by emphasising strong local structure. A hypergraph can be progressively partitioned into smaller hypergraphs by removing the connections carrying the least modularity value, which peels away the weakest inter-group links while retaining the dense intra-group ones. Modularity has previously been used to detect mobility-aware communities in vehicular networks [8], and here it plays the analogous role of exposing the community structure that the spectral stage subsequently exploits. Because the objective that will later be optimised is a log-likelihood that may exhibit several local optima, the iterative procedure operating on this matrix is restarted from multiple initialisations to increase the probability of reaching a good solution. When the adjacency matrix and the modularity matrix are visualised side by side for real traffic data, they exhibit a high degree of overlap in their strongly connected regions, confirming that the modularity matrix captures the same geographic community structure that the adjacency matrix reflects while additionally exposing the spatial modularity that spectral clustering is designed to exploit.

3.3.2 Laplacian Matrix Construction

The Laplacian is the operator whose eigen-decomposition provides both the cluster count of Section 3.3.3 and the embedding that is partitioned in Section 3.4. Spectral clustering built on this operator has been shown to enhance clustering stability in vehicular networks [16], and its theoretical foundation rests on the normalised-cut

criterion [79] and the classical spectral clustering algorithm [80]. In its elementary form the Laplacian of the network is defined as:

$$L = D - Adj \quad (3.3)$$

where Adj is the adjacency matrix encoding the pairwise connections between vertices and D is the diagonal degree matrix whose i -th diagonal entry holds the degree of vertex i . To exploit the community structure captured by the modularity matrix rather than the raw adjacency, the proposed framework replaces the adjacency-based Laplacian with a symmetrically normalised, modularity-based Laplacian:

$$L = Dig^{-1/2} B Dig^{-1/2} \quad (3.4)$$

Here B is the modularity matrix of Equation 3.2 and Dig is the diagonal matrix obtained by traversing the hypergraph and summing, for each vertex, only the contributions of the vehicle nodes that lie one hop away. The diagonal entries of Dig are therefore given by:

$$Dig_{ii} = \sum_{j=1}^N B_{ij} \quad (3.5)$$

The symmetric normalisation ensures that vertices of widely differing degree are treated comparably, which stabilises the spectrum and improves the quality of the embedding. The eigenvalues of this normalised Laplacian are the common inputs to the eigen-gap heuristic of Section 3.3.3, and its leading eigenvectors form the low-dimensional representation in which the vehicles are subsequently clustered. In this way, the same operator simultaneously determines how many clusters to form and provides the geometry in which they are formed.

3.3.3 Optimal Cluster Determination using Eigen Heuristics

Selecting the number of clusters is a recurring difficulty in every clustering task, and in a mobility-driven network it is compounded by the requirement that the chosen value must serve the maximum area and the maximum number of vehicles with the least expenditure of resources. Partition-based methods such as k-means perform poorly here, because as the projected cluster shapes become non-convex and the data grow high-dimensional their accuracy deteriorates. For this reason, the eigen-gap

heuristic, which operates on the spectrum of the modularity-based Laplacian of Section 3.3.2, is adopted to determine the cluster count, following the eigen-heuristic approach applied to intelligent vehicular clustering [28].

Let the Laplacian of the vehicular network be a symmetric matrix of size $n \times n$, where n is the number of vehicles, and let its eigenvalues be arranged in non-decreasing order as $\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$. The eigen-gap heuristic computes the sequence of successive differences $\delta_i = \lambda_{i+1} - \lambda_i$ for $i = 1, 2, \dots, n-1$ and selects, as the optimal number of clusters, the index at which this difference attains its maximum:

$$k = \operatorname{argmax}(\delta_1, \delta_2, \dots, \delta_{n-1}) \quad (3.6)$$

The reasoning behind this rule is geometric. A large gap between two consecutive eigenvalues signals the greatest separation attainable between groups of vertices in the spectral embedding; the number of eigenvalues lying below this gap corresponds to the number of well-separated clusters that the data support. By locating the widest gap, the heuristic strikes a balance between too many and too few clusters and thereby preserves stability. In the proposed framework this computation is performed once, at the instant of maximum vehicle density, and the resulting value of k is then held fixed for the entire simulation, which reduces the computational burden and yields a consistent basis for evaluating the network parameters as the density varies.

3.4 Proposed iTTM-Based Cluster Formation

The clustering engine of the proposed framework is an iTTM procedure [81]. It retains the tensor-trace-maximisation objective that expresses hypergraph partitioning as the maximisation of an associativity measure, but it replaces the k-means step conventionally used to partition the spectral embedding with a Gaussian Mixture Model. The result is a spectral clustering pipeline that captures group interactions through the hypergraph, exploits community structure through the modularity-based Laplacian, and forms probabilistically modelled, non-spherical clusters through the GMM.

3.4.1 Improved Tensor Trace Maximization (iTMM)

The degree of a vertex v is the total weight of the hyperedges incident on it, $\deg(v) = \sum_{e \in \mathcal{E}: v \in e} W_e$. For a candidate partition $\{V_1, \dots, V_k\}$, the number of nodes incident on a subset,

$CM(V_1)$ and $CM(V_i) = \sum_{v \in V_1} \deg(v)$ and is referred as number of nodes incident on the node V_1 such that $V_1 \subseteq V$. The association between the edges contained within V_1 is given as $assoc(V_1) = \sum w_e$. The quality of the partition is then measured by its normalised associativity:

$$Par(V_1, \dots, V_k) = \sum_{i=1}^k \frac{assoc(V_i)}{CM(V_i)} \quad (3.7)$$

A partition that maximises this quantity concentrates hyperedge weight inside the subsets and starves the connections between them, which is precisely the behaviour required of good clusters. To render the objective amenable to a tensor formulation, the adjacency structure is first lifted to an order- z tensor whose entries record the weight of the hyperedge spanning a set of distinct vertices:

$$A_{i_1, i_2, i_3, \dots, i_z} = \begin{cases} W_{\{i_1, i_2, i_3, \dots, i_z\}} & \text{if } i_1, i_2, i_3, \dots, i_z \text{ are distinct} \\ 0 & \text{otherwise} \end{cases} \quad (3.8)$$

Introducing the modularity matrix B and the vertex-count-inverse quantity associated with each subset, the normalised associativity can be rewritten as a tensor trace. Denoting by $\times_\ell$ the mode- ℓ tensor product, the objective becomes:

$$Par_{i \in (1, \dots, k)} = \frac{1}{z!} \text{Trace}(B \times_1 Y^{(1)T} \times_2 Y^{(2)T} \times_3 Y^{(3)T} \dots \times_z Y^{(z)T}) \quad (3.9)$$

in which the factor matrices $Y^{(1)T}, Y^{(2)T}, Y^{(3)T}, \dots, Y^{(z)T} \in \mathbb{R}^{k \times z}$ encode, for each vertex, the count of cluster members with which it is associated. Each factor is normalised by the inverse of the summed member count of its subset,

$$Y^i = \frac{1}{\sum_k CM(V_i)} \quad (3.10)$$

Maximising the trace in Equation 3.9 over the factor matrices is the tensor-trace-maximisation (TTM) problem. Because the associated log-likelihood surface is non-convex, the optimisation is initialised from several starting points and the best solution

retained. Solving it yields the leading eigenvalues and eigenvectors of the modularity-based Laplacian, which together constitute the spectral embedding of the vehicular hypergraph. The distinctive feature of the improved variant, iTTM, is what happens to this embedding next: instead of committing the embedded points to a k-means assignment, they are handed to a Gaussian Mixture Model.

3.4.2 Gaussian Mixture Model-Based Cluster Generation

The k-means algorithm partitions the spectral embedding by assigning each point to the nearest centroid, which implicitly assumes that clusters are spherical and of comparable size [82]. In a road network, whose clusters are elongated and irregular, this assumption is unrealistic and leads to unstable assignments. A Gaussian Mixture Model relaxes it by describing the data as a weighted combination of Gaussian components, each with its own mean and covariance, so that clusters of arbitrary orientation and spread can be represented [5]. The mixture density is:

$$\sum_{i=1}^k \delta_i \mathcal{N}(\mu_i, \Sigma_i) \quad (3.11)$$

where k is the number of components, δ_i is the mixing probability that a point belongs to component i , and $\mathcal{N}(\mu_i, \Sigma_i)$ is the multivariate normal density with mean μ_i and covariance Σ_i . Model fitting proceeds by expectation–maximisation, an iterative scheme akin to k-means: the expectation step assigns responsibility for each point to the components, and the maximisation step re-estimates the component parameters; the two steps alternate until convergence. Applying Bayes' theorem, the posterior probability that sample x_j was generated by component i is:

$$\begin{aligned} P_M(z_j = i | x_j) &= \frac{P(z_j = i) \cdot p_M(z_j = i | x_j)}{p_M(x_j)} \\ &= \frac{\alpha_i \cdot p(x_j | \mu_i, \Sigma_i)}{\sum_{l=1}^k \alpha_l \cdot p(x_j | \mu_l, \Sigma_l)} \end{aligned} \quad (3.12)$$

The posterior responsibilities γ_{ji} obtained from Equation 3.12 assign each embedded vehicle to the component under which it is most probable, and the eigen matrix X is thereby partitioned into k clusters $C = \{C_1, C_2, \dots, C_k\}$. Because the assignment is probabilistic and covariance-aware, the resulting clusters adapt to the true distribution of the vehicles far more faithfully than a hard, distance-based partition, and this

improved fidelity is what raises the accuracy and the cluster-head stability of the hypergraph-based scheme.

3.4.3 Spectral Cluster Formation Process

The complete cluster-formation process weaves together the components developed above into a single sequence of operations. The procedure begins by identifying the time slot at which the number of active vehicles is greatest, because fixing the cluster count at maximum density guarantees that the chosen partition is adequate for every lighter load encountered during the simulation. A hypergraph is then constructed over the vehicles present in that slot, with hyperedges, vertices, and weights assigned as described in Section 3.2.2. From the inter-vehicular proximities a similarity matrix is derived, a square matrix whose order equals the number of vehicles and whose entries measure pairwise closeness, and from it the modularity matrix and the normalised Laplacian of Section 3.3 are computed.

Eigen-decomposition of the Laplacian supplies the eigenvalues that feed the eigen-gap heuristic, which fixes the optimal number of clusters. The leading eigenvectors are normalised to form the spectral embedding, and the rows of this embedding are clustered by the GMM of Section 3.4.2, producing the k vehicle clusters. The pseudo-code of the overall iTTM procedure and of the eigen-heuristic sub-routine that it invokes is presented in Section 3.7. By combining hypergraph representation, modularity-based spectral clustering, and GMM assignment, the process delivers an efficient and adaptive means of generating clusters that remain coherent under the dynamic conditions of a vehicular network.

3.5 Multi-Criteria Cluster Head Selection Framework

Once the clusters have been formed, exactly one vehicle in each cluster must be elevated to cluster head. A vehicle is a strong candidate for this role if it maintains a nearly constant velocity relative to its neighbours, is connected to many of them, sustains its links for a long time, and enjoys a high level of trust within the local community. These intuitions are made precise through four quantitative criteria, each computed for a vehicle relative to the neighbours to which it is joined through hyperedges, and each subsequently modularised so that the community structure of the

hypergraph informs the metric. When a vehicle enters a cluster, it scans for other vehicles able to receive its signal, and from these observations it derives its connectivity level, its relative speed, its anticipated link duration, and its trust value based on the ratings assigned by its immediate surroundings. The four criteria are formulated below and then fused, in Section 3.5.5, through the weighted CRITIC method.

3.5.1 Relative Speed Modeling

The relative cooperative speed of a vehicle is its velocity measured against those of the vehicles around it. Vehicles that travel much faster or much slower than their neighbours leave the cluster sooner, causing link breakages and shortening the time spent in mutual company; a vehicle that maintains a steady velocity relative to its context is therefore a more stable choice of cluster head. A running measure ψ_{veh_i} of this steadiness is updated at each step: it is reinforced while the deviation of the vehicle's speed v from the neighbourhood average v_{avg} remains within a threshold S_{thr} , and it is penalised once the deviation exceeds that threshold,

$$\begin{aligned} \psi_{veh_i}(t+1) &= \psi_{veh_i}(t) + \delta; |v - v_{avg}| \leq S_{thr} \\ \psi_{veh_i}(t+1) &= \psi_{veh_i}(t) + \delta; |v - v_{avg}| \geq S_{thr} \end{aligned} \quad (3.13)$$

The threshold S_{thr} is the cut-off, fixed at 2.77 in this work, above which the velocity deviation is deemed disruptive; the increment δ is initialised to 0.01 using the TraCI parameters, and the neighbourhood average is obtained by subtracting the surrounding vehicles' speeds from the individual vehicle's speed. To let the community structure influence the metric, it is modularised with the aid of the modularity matrix, so that the contribution of each neighbour is discounted by the null-model expectation:

$$\psi_{veh_i}^B = \sum_{D_{ij} \in NE} (\psi_{veh_i} \times T_{ij} - \frac{k_i k_j}{2m}) \quad (3.14)$$

Finally, the k dominating eigenvalues obtained from the modularity-based Laplacian are summarised into a single relative-cooperative-speed score by taking their arithmetic mean,

$$RCL_i = \frac{1}{|k|} \sum_{\lambda_i \in k} \lambda \quad (3.15)$$

A high score indicates a vehicle whose motion is well aligned with its neighbourhood and which is consequently likely to hold its cluster together.

3.5.2 Connectivity Level Modeling

The connectivity level counts the neighbouring vehicles with which a candidate can communicate; a well-connected vehicle improves performance and lowers the probability of packet loss, and keeping cluster nodes in continuous contact is central to cluster-head stability. The links available to a vehicle are recorded by a binary covering matrix T_{ij} , which is unity when two vehicles lie within transmission range and zero otherwise,

$$T_{ij} = \begin{cases} 1 & \text{if } D_{ij} < R_{vehi} \\ 0 & \text{otherwise} \end{cases} \quad (3.16)$$

where D_{ij} is the instantaneous separation of the two vehicles. Because the ability to exchange data falls off with distance, only vehicles satisfying $R_{vehi}(D_{ij}) > 0$ are treated as neighbours. The raw connectivity of a vehicle is the number of such neighbours,

$$CL = \sum_{D_{ij} \in NE} (T_{ij}) \quad (3.17)$$

As with relative speed, this count is reconstructed as a modularised, cumulative correlation measure on the weighted hypergraph, in which each link is discounted by the modularity null model:

$$CL^B = \sum_{D_{ij} \in NE} (T_{ij} - \frac{V_{k_i} V_{k_j}}{2m}) \quad (3.18)$$

The TTM Laplacian formulation then yields k dominating eigenvalues, and the cooperative connectivity of the vehicle is expressed as their arithmetic mean,

$$CCL_i^B = \frac{1}{|k|} \sum_{\lambda_i \in k} \lambda \quad (3.19)$$

A larger value denotes a vehicle embedded in a dense, cohesive neighbourhood, which experiences the least switching and therefore contributes most strongly to stability.

3.5.3 Link Lifetime Modeling

Sustainable links are indispensable in a dynamic urban environment because they raise data-transfer efficiency and lower maintenance cost. The classical Link Expiration Time was introduced to predict how long a connection will survive; the present work adopts a refined variant, the Link Lifetime, which measures the duration for which two vehicles remain connected. In a city where vehicles change direction frequently, a high LLT is a direct indicator of a durable link and hence of a dependable cluster head. Assuming that two nodes within transmission range can communicate, and accounting explicitly for the fluctuations in their coordinates, the link lifetime between vehicles i and j is:

$$LLT_{ij} = \frac{-\Delta v \cdot D_{ij} + |\Delta v| \cdot R_{veh}}{\Delta v^2} \quad (3.20)$$

where $\Delta v = v_{x_i} - v_{x_j}$ is the difference in the vehicles' velocities and the distance is constrained to lie within transmission range. Equation 3.20 has a singularity when two vehicles move at almost identical speeds: the denominator Δv^2 tends to zero and the expression returns an undefined value. To keep the metric well-behaved, whenever the relative speed is zero the link lifetime is set to the maximum admissible value determined by the driving environment; for the urban scenario studied here a maximum speed of 40 km/h is used for this purpose. This convention assigns the longest possible link lifetime to a vehicle moving in perfect concert with its neighbours and thereby raises its chance of being selected as a stable cluster head. The modularised treatment of link lifetime further enriches the connectivity analysis by weighing not merely the existence of a link but its position within the modular structure of the hypergraph, ensuring that the links which are retained are those that are sustained most efficiently.

3.5.4 Consensus Trust Score Modeling

The final criterion captures the reliability of a vehicle as judged by its community [2], [29]. The model draws on cognitive spectrum sensing: priority users such as VIP cars and emergency vehicles are treated as primary users (PUs), and the remaining vehicles act as secondary users (SUs) that opportunistically use the communication spectrum. Each vehicle carries a spectrum sensor, and when a primary user is detected

the secondary user relinquishes the spectrum to it, an act of cooperation that improves the secondary user's trustworthiness. A secondary user detects the presence of a primary user by comparing the received signal intensity against a probabilistic threshold; the energy test statistic collected during the sensing of a vehicle is:

$$F(Y) = \frac{1}{2N} \sum_{i=1}^N |Y_i^{re} + Y_i^{im}|^2 \quad (3.21)$$

where $F(Y)$ aggregates the real and imaginary components of the samples collected while testing a vehicle. This statistic is a random variable whose distribution may be approximated by a chi-square test, and the probability of correctly identifying a vehicle as a primary user is:

$$P_d(\varepsilon, t) = Q \left(\left(\frac{\varepsilon}{\sigma_u^2} - \gamma - 1 \right) \sqrt{\frac{t f_s}{2\gamma + 1}} \right) \quad (3.22)$$

If P_d exceeds the threshold ε , the vehicle is identified as a primary user. When a secondary vehicle finally decides to vacate the spectrum, its trust deficit is repaired, so that every vehicle carries a trust value reflecting its history of cooperation. Because this value is readily influenced by nearby vehicles, the local community must reach consensus before the trust index is accepted: each node revises its trust value for a vehicle only when the fraction of packets missed from its neighbours falls below two per cent, and the trust table is kept current accordingly. Every edge of the adjacency matrix is then weighted by the prevailing trust value, and after modularisation the trust criterion is expressed on the hypergraph as,

$$S^B = \sum_{D_{ij} \in NE} (t_{vehi} \times T_{ij} - \frac{k_i k_j}{2m}) \quad (3.23)$$

A high consensus trust score marks a vehicle that its neighbours have found consistently reliable, which lengthens link lifetime and improves cluster-head stability, whereas a low score signals susceptibility to malicious behaviour, link dropouts, and packet loss. Although the trust score is not used to compute network stability directly, it is decisive for the reliability of the elected cluster head.

3.5.5 Weighted CRITIC-Based Criteria Evaluation

The four criteria of Sections 3.5.1 to 3.5.4 must be reconciled into a single ranking of candidate vehicles. The conventional weighted-sum approach requires the designer to supply the weights manually, which is subjective and cannot respond to the statistical structure of the data. The proposed framework instead derives the weights objectively through the CRITIC method, which determines the importance of each criterion from two ingredients: the contrast intensity of the criterion, measured by its standard deviation, and the conflict it exhibits with the other criteria, measured through their linear correlation. A criterion that varies widely and that is weakly correlated with the others carries more independent information and is therefore assigned a larger weight.

The procedure begins by normalising each criterion to a common scale. Pearson correlation coefficients are computed among the normalised criteria, and the standard deviation of each criterion is recorded. For every criterion the measure of conflict it introduces into the decision is obtained by summing one minus its correlations with the remaining criteria. Multiplying this conflict measure by the standard deviation gives the information content of the criterion, and normalising these products across all criteria yields the objective weights. Denoting the four weights by w_1 , w_2 , w_3 , and w_4 , they are guaranteed to be non-negative and to sum to unity, so that they can be applied directly in a weighted-sum fusion. The pseudo-code that implements this computation is given as Algorithm 4 in Section 3.7. By making each criterion's contribution proportional to its relevance and to its independence from the others, the weighted CRITIC method renders the cluster-head decision both objective and reproducible.

3.6 Proposed iTTM–Weighted CRITIC Optimization Framework

3.6.1 Integrated Optimization Framework

The two halves of the design, cluster formation and cluster-head selection, are unified into a single optimisation framework that operates in a centralised, RSU-assisted manner. The framework accepts as input the instantaneous vehicle parameters exported by SUMO and produces as output, for every simulation instant, a set of stable

clusters each governed by a well-chosen cluster head. Its operation can be understood as a two-stage optimisation for stable CH selection as seen in Figure 3.2. In the first stage the vehicular hypergraph is partitioned so as to maximise the tensor-trace associativity of Equation 3.9 subject to the fixed cluster count determined by the eigen-gap heuristic; this stage yields clusters whose internal cohesion is as high as the modularity structure permits. In the second stage, within each cluster the four selection criteria are evaluated and combined through objectively derived CRITIC weights, and the vehicle that best satisfies the joint criterion is promoted to cluster head.

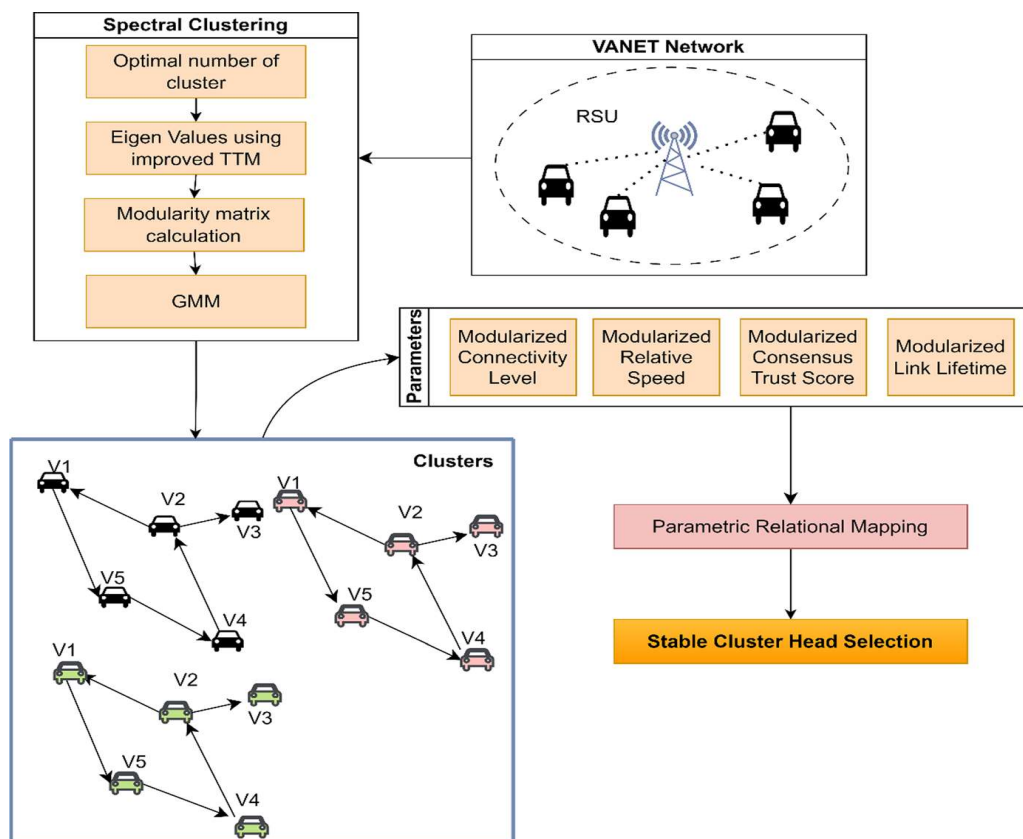


Figure 3.2 Proposed scheme for stable cluster-head selection in VANET

Fixing the cluster count once, at maximum density, is a deliberate optimisation choice. It removes the need to recompute the partition at every instant, which lowers the computational time to that required for a single maximum-density evaluation, and it allows the network parameters to be studied under a fixed number of cluster members irrespective of the fluctuating vehicle density. The two stages are coupled through the modularity-based Laplacian: the same spectral quantities that define the clusters also

enter, through their dominating eigenvalues, into the modularised connectivity and relative-speed criteria of Section 3.5. This coupling ensures that the notion of a good cluster and the notion of a good cluster head are grounded in the same underlying structure, which is the source of the stability improvements reported in Section 3.8.

3.6.2 Cluster Head Decision Model

The decision model formalises the second stage. Within a cluster, and at a given simulation instant, the connectivity, relative-speed, and link-lifetime criteria are computed for every candidate vehicle and normalised to the interval $[0, 1]$; the trust score contributes to reliability but, because it does not bear directly on stability, is excluded from the stability decision itself. Denoting by $nx_{c,j}$ the normalised value of criterion c for vehicle j , and by w_c the objective CRITIC weight of that criterion, the aggregate suitability of vehicle j is the weighted sum:

$$Score_j = \left(\sum_{c=1}^k w_c \right) * nx_{c,j} \quad (3.24)$$

The vehicle attaining the maximum aggregate suitability is elected cluster head for that instant,

$$CH = \operatorname{argmax}_{j \in C_k} (Score_j) \quad (3.24 \text{ a})$$

A concrete illustration clarifies the mechanism. For a simulation instant i and a candidate vehicle j , suppose the modularised connectivity, relative-speed, and link-lifetime values are 1, -0.0498 , and 5 respectively. The CRITIC procedure returns weights of 0.378, 0.333, and 0.289, which sum to unity and thus form a valid multi-criteria decision. After normalisation to $[0, 1]$ and multiplication by their weights, the vehicle's weighted attributes become 1, 0.12, and 0. The weighted attributes of every vehicle in the cluster are summed, and the vehicle carrying the maximum aggregate is designated the cluster head for that instant. Repeating this decision at each of the 1500 simulation stamps produces the sequence of cluster heads whose stability is analysed in Section 3.8. Because the weights are data-driven rather than hand-tuned, and because the criteria are grounded in the same spectral structure that defines the clusters, the elected heads are consistently those best able to hold their clusters together, which is the objective the framework was built to serve.

3.7 Proposed Algorithm Development

This section assembles the complete procedure as a set of four inter-locking algorithms. Algorithm 3.1 is the master iTTM clustering routine; it invokes Algorithm 3.2 to fix the optimal cluster count through the eigen-gap heuristic. Algorithm 3.3 performs stable cluster-head selection over the resulting clusters, and it invokes Algorithm 3.4, the weighted CRITIC routine, to obtain the objective criteria weights used in its decision step.

3.7.1 iTTM-Weighted CRITIC Algorithm

The master clustering routine constructs the hypergraph at the instant of maximum vehicle density, computes the modularity and Laplacian matrices, fixes the cluster count, and forms the clusters through GMM assignment of the normalised spectral embedding.

Algorithm 3.1 , Proposed iTTM clustering

Input: slot with maximum vehicles Vehi_max; location of each vehicle Vehi_Loc

1. Determine the time slot t at which the highest number of vehicles Vehi_max is present.
2. Construct a hypergraph $H = (E, V, w)$ from the vehicles present in that slot.
3. Compute the similarity / adjacency matrix from the inter-vehicular distances.
4. Calculate the modularity matrix B using Equation 3.2.
5. Compute the diagonal matrix Dig using Equation 3.5.
6. Construct the normalised Laplacian L using Equation 3.4.
7. Determine the optimal number of clusters K via Equation 3.6 (Algorithm 3.2).
8. Normalise the spectral-embedding matrix X .
9. for $i = 1 : K$
 10. generate the clusters $C = \{C1, C2, C3, \dots, CK\}$ by GMM assignment (Eq. 3.11, 3.12).
11. end for

Output: clustered vehicles

The optimal cluster count is fixed by the eigen-gap sub-routine, which decomposes the Laplacian and locates the widest gap between successive eigenvalues.

Algorithm 3.2 , Optimal cluster selection through eigen heuristics

Input: distance matrix Y

1. Determine the affinity matrix $S (n \times n)$ from Y .
2. Construct the Laplacian $L = D - Adj$, where $D_{ii} = \sum_j S_{ij}$ (i, j index the vertex labels).
3. Perform eigen-decomposition $[U, V] = eig(L)$; sort eigenvalues ascending as $(\lambda_1, \lambda_2, \dots, \lambda_n)$.
4. Compute successive differences $\delta_i = \lambda_{i+1} - \lambda_i$ for $i = 1, 2, \dots, n - 1$.
5. Compute consecutive ratios $\gamma_k = \delta_{k-1} / \delta_k$ for $k = 2, 3, \dots, k_{max}$ ($k_{max} = \text{maximum clusters}$).
6. Choose the number of clusters k that maximises γ_k , consistent with Equation 3.6.

Output: optimal number of clusters k

With the clusters fixed, cluster-head selection iterates over the simulation horizon and, within every cluster, evaluates the four criteria, obtains their CRITIC weights, and elects the vehicle of maximum aggregate suitability.

Algorithm 3.3 , Stable cluster-head selection

Input: optimal number of clusters k; cluster C_k

```

for i = 1 : sim_time
    for k = 1 : k
        for j = 1 : Ck
            1. Calculate modularised connectivity level (Eq. 3.18–3.19).
            2. Calculate modularised relative speed (Eq. 3.14–3.15).
            3. Calculate modularised consensus trust score (Eq. 3.23).
            4. Calculate modularised link lifetime (Eq. 3.20).
            5. Perform relational analysis using the CRITIC method.
            6. Generate the parameter weights from the weighted CRITIC method (Algorithm 3.4).
            7. Select the vehicle with the maximum weighted sum (Eq. 3.24, 3.24a) as CH.
        end for
    end for
end for

```

Output: stable cluster head per cluster per instant

The weighted CRITIC routine supplies the objective weights required in step 6 above.

Algorithm 3.4 , Weighted CRITIC method

Input: X, the stable-CH parameter matrix

```

1. Normalise the parameters, nx.
2. Determine the Pearson correlation p of the normalised values.
3. Find the standard deviation σ of each criterion.
4. for i = 1 : size(nx, 2) // conflict of each indicator against the others
    r(ii) = sum(1 - p(:, ii))
end for
5. for ii = 1 : size(nx, 2) // weight calculation
    w(ii) = ( σ(ii) · r(ii) ) / sum(σ · r)
end for
6. Repeat for the four parameters to obtain w1, w2, w3, w4.

```

Output: parameter weights w1, w2, w3, w4

3.8 Theoretical Analysis

Beyond its empirical performance, the framework rests on firm theoretical foundations. This section analyses the four algorithms of Section 3.7 along three axes: the computational cost they incur, the stability they confer on the resulting clusters,

and the soundness of their optimisation strategy. Together these properties explain why the framework scales gracefully, why its cluster heads persist under mobility, and why its decisions are principled and reproducible.

3.8.1 Computational Complexity Analysis

The cost of the framework is best understood at the level of its four constituent algorithms. Let n denote the number of vehicles at maximum density, k the number of clusters, d the dimensionality of the spectral embedding ($d = k$), m the number of selection criteria ($m = 4$), c the number of candidate vehicles in a cluster, t the number of EM iterations, and T the length of the simulation horizon. The dominant cost of each algorithm, together with how often it is executed, is summarised in Table 3.1.

Table 3.1 Computational complexity of the four proposed algorithms

Algorithm	Principal operations	Executions	Dominant complexity
3.1 iTTM clustering	Hypergraph and similarity matrix (Steps 2–3), modularity matrix (Step 4), diagonal and normalised Laplacian (Steps 5–6), spectral normalisation (Step 8), GMM cluster generation (Steps 9–11)	Once, at maximum density	$O(n^3)$ via Algorithm 3.2; matrix stages $O(n^2)$; GMM $O(t \cdot nkd)$
3.2 Eigen heuristics	Affinity matrix (Step 1), Laplacian (Step 2), eigen-decomposition and sorting (Steps 3–4), eigen-gap ratios (Steps 5–6)	Once, invoked by Algorithm 3.1	$O(n^3)$, reducible to $O(n^2k)$ [83]
3.3 Stable CH selection	Four modularised criteria per candidate (Eqs 3.14–3.23), CRITIC relational analysis and CH election (Eqs 3.24, 3.24a)	Every instant, over the horizon	$O(T \cdot n \cdot m^2)$
3.4 Weighted CRITIC	Normalisation, Pearson correlation, standard deviation, and weight computation over m criteria	Per cluster per instant, invoked by Algorithm 3.3	$O(m^2 \cdot c)$, effectively $O(c)$ since m is constant

The strength of this design becomes clear when the frequencies are taken into account. The only cubic cost, the eigen-decomposition, resides in Algorithm 3.2, which is invoked by Algorithm 3.1 a single time at maximum density; the resulting cluster count is then reused for the entire simulation (Sections 3.3.3 and 3.4). This one-time expense is therefore amortised across all T steps rather than repeated. The algorithms that run

continuously, Algorithm 3.3 and its weighted CRITIC sub-routine Algorithm 3.4, are only linear in the number of vehicles at each instant, because the criterion count m is a small constant. Concentrating the heavy computation in a single off-line pass while keeping the per-instant work lightweight is precisely what allows the framework to sustain real-time cluster-head selection in dense networks, with an overall memory requirement of $O(n^2)$.

3.8.2 Cluster Stability Analysis

The framework is engineered to produce clusters that endure, and this stability follows directly from the structure of its algorithms. Algorithm 3.2 fixes the cluster count at the widest gap $\delta = \lambda_{k+l} - \lambda_k$ in the Laplacian spectrum. A wide gap guarantees that the k leading eigenvectors span a strongly separated subspace, and the perturbation theory underlying spectral clustering shows that the larger this gap, the more resilient that subspace is to changes in the Laplacian [83]. Because vehicular mobility alters the topology only incrementally between consecutive instants, this resilience translates into a partition that holds its shape over time.

This effect is strengthened by the modularity-weighted Laplacian built in Algorithm 3.1 (Steps 4–6). Weighting the hyperedges by modularity [78] concentrates connection strength within communities and suppresses it between them, deepening the spectral separation and widening the eigen-gap still further, so that the two mechanisms reinforce one another. A further stabilising decision is that the cluster count, once obtained by Algorithm 3.2 at maximum density, is reused throughout the simulation; this removes the re-clustering churn that destabilises schemes which recompute their partitions as density fluctuates. Finally, because Algorithm 3.3 elects each cluster head from objective, data-driven weights rather than manually tuned ones, the same vehicle tends to be re-elected while its measured superiority persists, which stabilises the cluster head itself and not merely the membership.

These design choices are borne out by the results: the lowest partition variance occurs at three clusters, and the cluster-head change rate rises only marginally with speed, remaining well below that of the junction-based and multi-hop baselines. The

structural argument and the measured behaviour thus point to the same conclusion, giving strong, mutually confirming evidence of the framework's stability.

3.8.3 Optimization Analysis

The optimisation strategy is among the framework's most elegant features, since the four algorithms together decompose a problem that is intractable in its exact form into three stages, each solved by the most reliable means available.

The first stage is a spectral relaxation carried out by Algorithms 3.1 and 3.2. The exact, discrete maximisation of the normalised associativity of the partition (Equations 3.7 and 3.9) is an NP-hard combinatorial problem, yet relaxing the discrete cluster indicators to continuous values converts it into the eigen-problem of the modularity-based Laplacian, whose leading eigenvectors are the global optimum of the relaxed problem and are obtained directly, in closed form, by the eigen-decomposition of Algorithm 3.2 [83]. The framework therefore solves its core partitioning objective optimally rather than heuristically.

The second stage rounds this embedding into discrete clusters using the GMM assignment of Algorithm 3.1 (Step 10), fitted by expectation–maximisation, which increases the data log-likelihood at every iteration and converges reliably to a high-quality solution [84]. By fitting from several initialisations and retaining the best result, the framework consistently secures strong, mobility-aware assignments that adapt to the true, non-spherical shape of vehicular clusters, an advantage the conventional k-means step cannot offer.

The third stage, the weighting of the selection criteria in Algorithm 3.4, is fully deterministic. Each weight is derived in closed form from the standard deviation and the inter-criteria correlations of the data [85], as expressed in the conflict and weight computations of that algorithm. This yields weights that are objective, perfectly reproducible, and free of the subjectivity of manually tuned schemes, so that identical data always produce identical, defensible cluster-head decisions. Taken together, a globally solved relaxation, a robust and adaptive probabilistic assignment, and a deterministic weighting form an optimisation pipeline that is at once tractable,

principled, and precisely aligned with the stable, high-performing cluster heads demonstrated in the evaluation that follows.

3.9 Performance Evaluation and Discussion

The proposed iTTM-weighted CRITIC framework is evaluated through extensive simulation and compared against representative state-of-the-art schemes. This section first describes the experimental setup, then reports the comparative results across cluster-count selection, sparse and dense networks, and varying vehicle densities, and finally distills the principal findings.

3.9.1 Simulation Environment and Experimental Setup

The framework is implemented on the MATLAB platform, while vehicle movement is generated by SUMO [86] and streamed into MATLAB through the TraCI application programming interface. MATLAB performs the clustering and cluster-head selection, and SUMO parses the vehicular environment into a matrix of vehicle parameters sampled at every simulation instant. The urban scenario reproduces Delhi's Connaught Place, whose road network is imported from OpenStreetMap and comprises 293 lanes; the simulated region is centred at 28°39'39" N, 77°13'48" E. Scenario generation follows the standard SUMO workflow: the OpenStreetMap extract is converted into a network file, random trips are generated so that vehicles travel uniformly from source to destination, a configuration file binds the network and route files, and departure options are set so that the required number of vehicles enters the network over the simulation window. The optimum cluster count is computed for the maximum density of 1000 vehicles and then held fixed, and the evaluation is repeated across a range of densities. The full simulation specification is given in Table 3.2, and the physical-layer parameters of the IEEE 802.11p radio model appear in Table 3.2.

Table 3.2 Specifications of the simulation environment

Simulation parameter	Value
Scenario	Urban
Lanes	293
Simulation time	1500 s

Simulation parameter	Value
Placement of RSU	500 m
Average speed	10.51 m/s
Transmission range of RSU	350 m
Transmission range of vehicle	200 m
Length of vehicle	5 m and 10 m
Speed of vehicle	0–25 m/s
Mobility protocol	Random way point (SUMO)
Routing protocol	One-hop communication

The radio model, developed to reflect intra-vehicular communication over a Rayleigh fading channel, generates random bits at 14 Mbps, applies subcarrier modulation with a 32-point FFT, introduces a Doppler shift to account for velocity-dependent attenuation, applies QPSK modulation, adds white channel noise convolved with Rayleigh fading, and finally demodulates and deconvolves the signal so that the bit errors can be measured. The corresponding parameters are listed below.

Table 3.3 Specifications of the network parameters (IEEE 802.11p)

Attribute	Measure
Doppler shift	$V_{s,r} / \lambda$
Rate of samples	10 MHz
Number of pilot subcarriers	4
FFT point count	32
Subcarrier count	$52 + D$
Data subcarrier count	52
Coding scheme	$\frac{1}{2}$
Modulation scheme	OFDM
Rate of data	14 Mbps
Cyclic prefix	16

Two network performance indicators are used to gauge stability: packet delay and throughput. Packet delay (PD) is the time taken for a data packet to travel from one vehicle to another, and it is influenced by traffic, noise, and hop distance; it is expressed as,

$$PD = \frac{datarate * sizeofPckt}{t} \quad (3.25)$$

Throughput is the rate, in megabits per second, at which data is transferred between vehicles; fewer hops and greater stability raise it,

$$Throughput = \frac{Bits_{Received}}{t} \quad (3.26)$$

where t denotes time. Cluster-head stability itself is quantified by how often a single vehicle is retained as cluster head over the life of the optimal solution. Sorting the occurrences of each vehicle identifier in descending order, the maximum frequency with which a given identifier is chosen as cluster head is used to form the stability bins,

$$CHS = sort\left(\max\left(\sum_{i=1}^t Vehi_{ID}\right), -1\right) \quad (3.27)$$

A histogram of cluster-member occurrences is compiled for each of the 1500-time stamps and for the fixed set of three clusters, and the network parameters are generated by simulating packet transmission once the clusters are formed. Higher link lifetime, optimal relative speed, higher trust, and higher connectivity all raise throughput and lower packet delay, as do lower total distance, an optimal cluster-member frequency, and a stable cluster head; conversely, as inter-node distance grows the signal weakens and both connectivity and effective relative speed fall, degrading performance. Optimising these factors jointly is therefore what yields the best network behaviour.

3.9.2 Comparative Performance Analysis

Optimal cluster count. The eigen-heuristic was run for twenty trials at each candidate maximum-cluster value, and box plots of the resulting cluster counts were examined for the range from five to eleven maximum clusters in Figure 3.3. A cluster count of three produced a distinctly narrow inter-quartile spread with notably shorter whiskers than the neighbouring configurations, signalling low variability and therefore the greatest stability in the clustering outcome; the highest variability appeared around six clusters. Three clusters were accordingly adopted for all subsequent experiments, and a visualisation of the clustered vehicles at maximum density confirmed that the partition captures the anticipated spatial grouping of the traffic across the simulation area as shown in Figure 3.4.

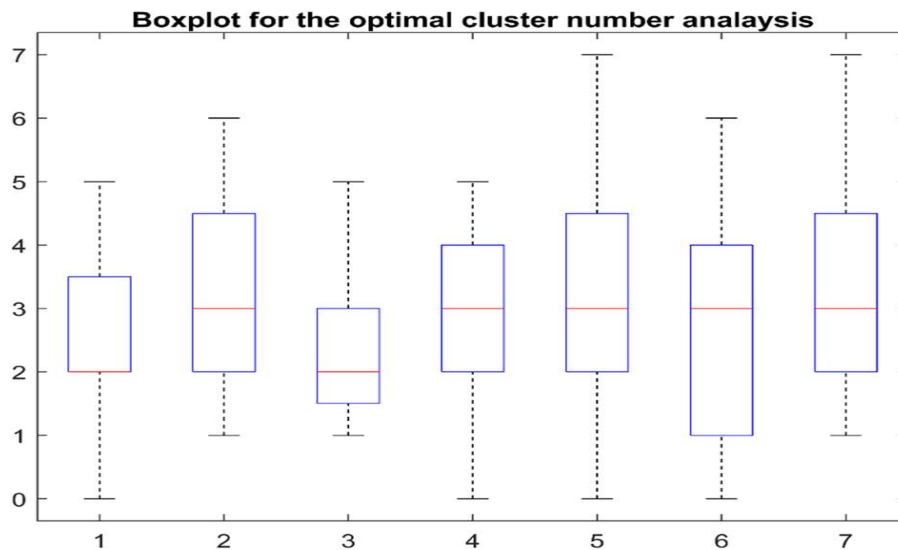


Figure 3.3 Optimal number of clusters analysis

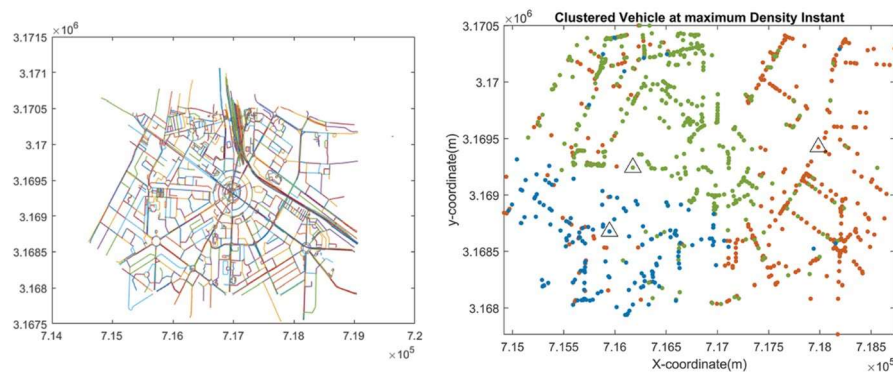


Figure 3.4 Cluster generation at the highest vehicle density

Individual criterion contribution. The contribution of each vehicle attribute to overall stability was isolated by comparing the full scheme against variants driven by a single attribute in Figure 3.5. The trust score was excluded from this stability comparison because it enhances reliability rather than acting directly on stability. Among the individual attributes, the connectivity level yielded the highest stability, because a high connectivity level arises when vehicles are in close proximity and the switching frequency is consequently at its lowest. The full multi-criteria scheme, however, dominated every single-attribute variant, confirming that the weighted combination of criteria selects better cluster heads than any criterion alone.

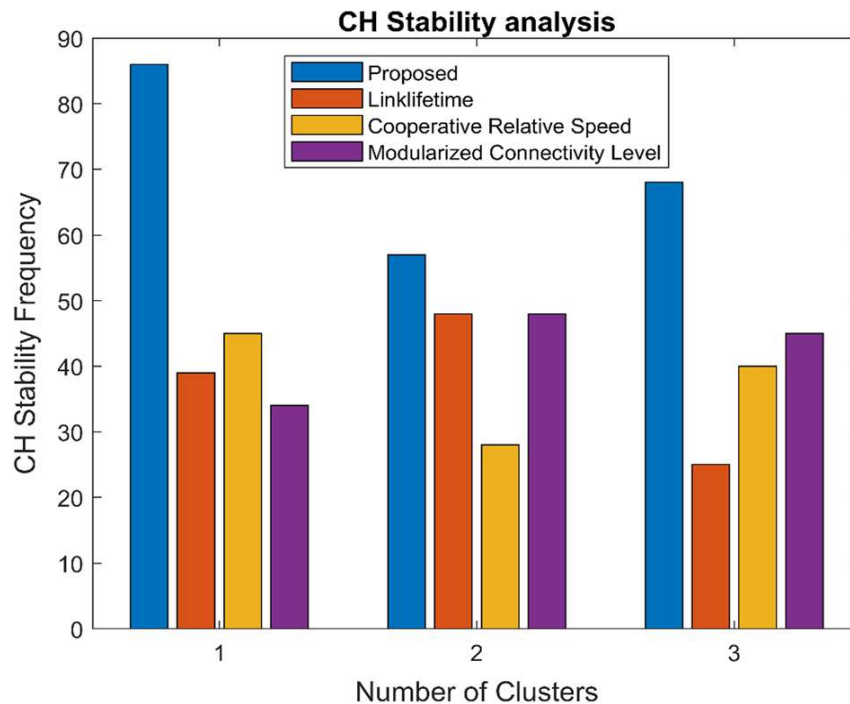


Figure 3.5 Overall stability and the effect of each individual vehicle attribute on the stability analysis

Sparse and dense networks. Stability was assessed under a sparse network of 100 vehicles and a dense network of 1000 vehicles in Figure 3.6 and Figure 3.7 respectively, examining packet delay and throughput jointly with cluster-member frequency and total distance. In the sparse case a high cluster-member frequency lowers stability because the cluster head switches more often; as stability improves, throughput rises and, with less inter-node distance and a lower cluster-member frequency, packet delay falls and throughput is maximised. Unlike schemes that suffer throughput degradation below one hundred vehicles, the proposed model sustains its highest throughput even in the sparse regime and thus satisfies the throughput requirement there.

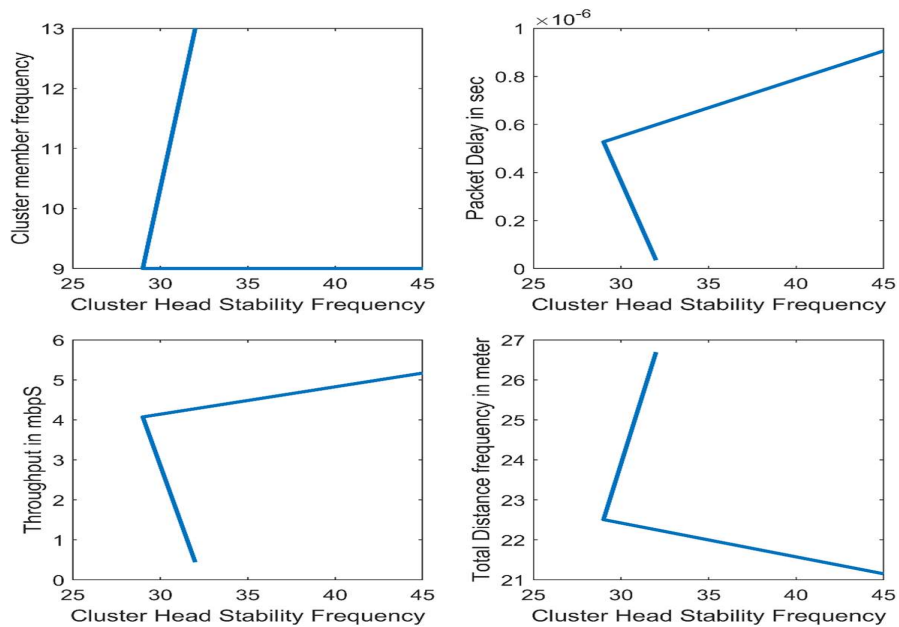


Figure 3.6 Evaluation on the sparse network of 100 vehicles.

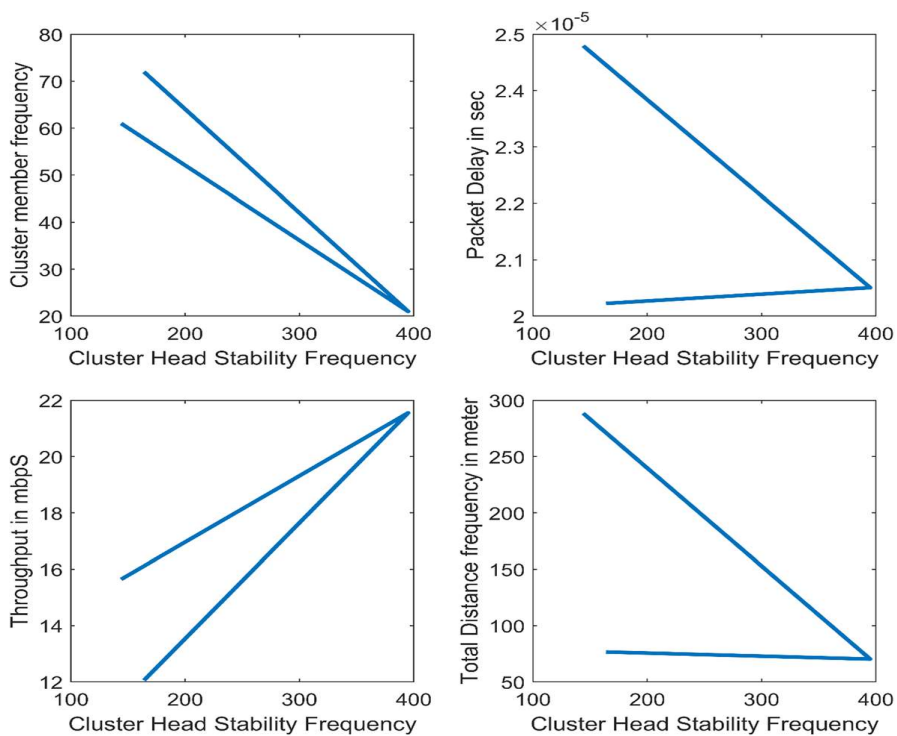


Figure 3.7 Evaluation on the dense network of 1000 vehicles

In the dense case, as the number of cluster members decreases the cluster-head stability improves, the hallmark of a scalable design, and the best stability coincides with a total inter-node distance below the 200 m transmission range, exactly the condition that keeps links alive. Both network types are therefore shown to be stable when their parameters fall within the ranges prescribed by the criteria of Section 3.5.

Varying vehicle density. Sweeping the vehicle count from zero to five hundred and up to the maximum of 1000, the average cluster-head stability was plotted against average cluster-member frequency, average total distance, average packet delay, and average throughput as shown in Figure 3.8. Cluster-head stability peaked at around 500 vehicles, where the average cluster-member frequency was lowest and the average distance travelled fell below 100 m, indicating that a high connectivity level and long link lifetime were being maintained. Between zero and five hundred vehicles the packet delay changed little because the average total distance remained essentially constant (the sparse regime), while increasing the count from 500 to 1000 raised the throughput (the dense regime). Across the whole range the cluster members exhibited little variation and kept the transmission links operational throughout their lifetime, which is the direct cause of the sustained cluster-head stability.

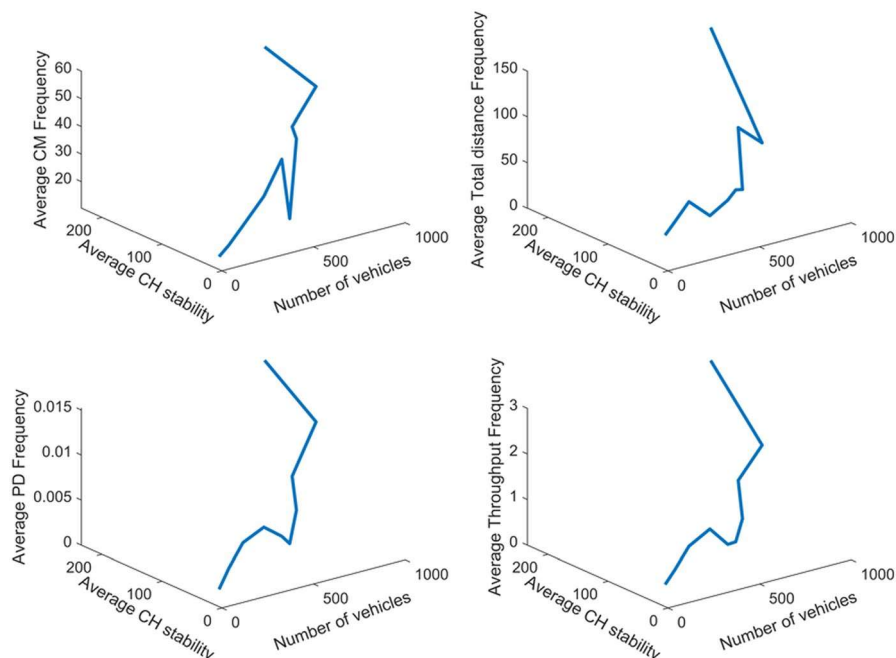


Figure 3.8 Analysis of varying vehicle density

State-of-the-art comparison. The framework was benchmarked against the hypergraph-based dynamic stable clustering scheme of Kumar et al. [36], which uses a fuzzy decision over link lifetime, cooperative connectivity, relative speed, and a vehicular hypergraph and is likewise piloted on an urban Delhi scenario, and against the connectivity-based clustering scheme of Khan et al. [28], which estimates cluster-head eccentricity and forms dynamic clusters by traffic volume over a two-kilometre road. Like both baselines the proposed work is simulated in SUMO; its 1500 s horizon matches that of Kumar et al. [36], while, unlike Khan et al. [28], it fixes the optimum cluster count for the whole simulation to reduce computation. Crucially, the proposed method replaces k-means with GMM in the spectral step and derives its criteria weights through CRITIC rather than a fuzzy or manual rule. The headline metrics are compared in Table 3.4.

Table 3.4 State-of-the-art comparison of the proposed framework against representative baselines

Scheme	No. of clusters	Avg. vehicles per cluster	CH stability	Throughput (Mbps)	PD (s)
Proposed model	3	51.3333	84%	3.2870	0.0148
Khan et al. [28]	140	9	NA	NA	4
Kumar et al. [36]	14	44.0588	82%	3.1840	0.0151

The critical analysis of these results can be organised around four observations.

- 1. Cluster density and stability.** Despite forming a higher density of vehicles per cluster than either baseline, the proposed model exhibits a lower switching frequency and higher cluster-head stability, because it manages cluster transitions more effectively.
- 2. Effect of speed on cluster formation.** Where Kumar et al. [36] use speed to denote hyperedges, obtaining low variance and a large number of clusters, the proposed model maintains a small, optimal, fixed set of clusters, which lowers switching frequency and reduces the likelihood of packet drops during transmission.

3. Effectiveness of fixed clusters. The use of optimal fixed clusters delivers lower packet delay and improved throughput regardless of density, indicating efficient management of network resources and stable communication channels.

4. Comparative metrics. The proposed model achieves 3.133% higher throughput than Kumar et al. [36], together with a 2% lower packet delay than Kumar et al. [36] and a reduction of 99.63% relative to Khan et al. [28] Its cluster-head stability of 84% exceeds the 82% of Kumar et al. [36], and it does so while using far fewer clusters.

Finally, cluster-head stability at road junctions, where vehicle direction is least predictable, was compared against a junction-based clustering scheme (JCV) [22] and a Vehicular Multi-hop algorithm for Stable Clustering (VMaSC) [32], using the rate of cluster-head change per second as a function of vehicle velocity; a lower rate indicates better stability. As Table 3.5 shows, the proposed scheme exhibited only about a 10% increase in change rate as velocity rose, against nearly 30% for JCV, confirming its robustness under the demanding conditions of an intersection.

Table 3.5 Cluster-head change rate per second versus vehicle speed

Vehicle speed (m/s) →	10	15	20	25	30	35
Proposed scheme	0.42	0.47	0.48	0.49	0.50	0.60
JCV [22]	0.68	0.68	0.82	0.87	0.98	0.99
VMaSC [32]	0.80	0.80	0.50	0.50	0.51	0.53

3.9.3 Discussion of Key Findings

Taken together, the experiments substantiate the three design hypotheses that motivated the chapter. First, fixing the number of clusters through the eigen-gap heuristic at maximum density and holding it at three produced the most stable clustering outcome while eliminating the recurring cost of re-estimating the cluster count; the low variability observed at three clusters is the empirical signature of this choice. Second, replacing k-means with a Gaussian Mixture Model in the spectral step allowed the clusters to follow the true, non-spherical distribution of vehicles on the road, which is reflected in the higher cluster-head stability and the sustained throughput across both sparse and dense regimes. Third, deriving the criteria weights

objectively through CRITIC removed the subjectivity of manual weighting and produced cluster heads that outperform every single-attribute alternative.

The practical consequences are consistent across all the experiments: the proposed framework keeps inter-node distance within the transmission range, sustains link lifetime, and thereby minimises cluster-head switching. This yields simultaneously higher throughput, lower packet delay, and a markedly lower rate of cluster-head change than the compared schemes, including under the adverse conditions of road junctions. The improvement is achieved with a smaller number of clusters, which is itself advantageous because it lowers control overhead. The framework is therefore both effective and scalable, and it advances cluster-head stability in urban VANETs beyond the state of the art examined here.

3.10 Chapter Summary

This chapter presented a centralised, RSU-assisted framework for stable cluster-head selection in urban VANETs, built on a hypergraph representation of the vehicular network. The framework combines an improved Tensor-Trace-Maximisation clustering algorithm, in which a Gaussian Mixture Model replaces k-means in the spectral partitioning step, with a weighted CRITIC decision model that fuses four cluster-head selection criteria: modularised relative cooperative speed, modularised connectivity level, modularised link lifetime, and a consensus trust score. The number of clusters is fixed once, at maximum vehicle density, using an eigen-gap heuristic, which both reduces computation and provides a consistent basis for evaluation.

The complete mathematical formulation was developed, from the hypergraph definition and the modularity-based normalised Laplacian, through the tensor-trace objective and its GMM-based partitioning, to the four criterion equations and the objective CRITIC weighting, and the whole procedure was assembled into four interlocking algorithms. Simulations conducted on a SUMO reconstruction of Delhi's Connaught Place, over a 1500 s horizon and across densities from 100 to 1000 vehicles, showed that three clusters is the optimal and most stable configuration, that connectivity level is the strongest individual contributor to stability, and that the full multi-criteria scheme dominates every single-attribute variant. Against state-of-the-art

baselines the framework delivered 84% cluster-head stability, 3.133% higher throughput and 2% lower packet delay than the closest hypergraph baseline, a 99.63% packet-delay reduction relative to a connectivity-based scheme, and a cluster-head change rate at junctions roughly a third that of a junction-based competitor.

Certain limitations frame the scope of these conclusions and point to future work. The hypergraph representation, while expressive, carries a computational cost that can become significant for very large networks, and the relationships encoded within hyperedges can be harder to interpret than those of ordinary graphs. The centralised architecture, which affords security and resolves synchronisation, remains subject to a single point of failure at the RSU. Future research could therefore pursue scalable and efficient algorithms tailored to hypergraphs, techniques to visualise and interpret hyperedge structure, and, most importantly for a safety-critical setting, the integration of stronger security mechanisms into the cluster-head selection process, which is taken up in the next chapter.

CHAPTER 4

PERFORMANCE EVALUATION OF THE MODIFIED INCENTIVE-BASED CLUSTERING FRAMEWORK

4.1 Introduction

The framework developed in Chapter 3 established that an objective, multi-criteria treatment of cluster-head selection can markedly improve the stability of a hypergraph-clustered vehicular network. That framework, however, rated candidate vehicles purely on their instantaneous kinematic and connectivity attributes and did not consider how the driving behaviour of a vehicle evolves over time. In dense urban traffic the conduct of individual drivers is far from uniform: some maintain steady speeds and predictable trajectories, while others accelerate and brake erratically, and it is precisely the erratic vehicles that most often trigger link breakage and force a cluster head to be re-elected. This chapter addresses that gap by evaluating a modified clustering framework in which cluster-head selection is made sensitive to the observed behaviour of vehicles through an incentive mechanism, so that steady, cooperative vehicles are preferred and volatile ones are discouraged from assuming leadership.

The framework studied here retains the hypergraph representation, the modularity-based spectral construction, and the Gaussian-mixture cluster assignment introduced in Chapter 3, and therefore those components are recalled only briefly rather than re-derived. Its two distinguishing contributions are a Modified Tensor-Trace-Maximisation (mTTM) clustering pipeline and an Adaptive Neighboring Behavior-Incentivization (ANB-I) mechanism for cluster-head selection. The ANB-I mechanism replaces the four static criteria of Chapter 3 with three adaptive behavioural metrics, the Adaptive Relative Average Speed, the Adaptive Neighborhood Degree, and the Adaptive Link Lifetime, each adjusted in real time by a reward and penalty rule that rewards usual, stable behaviour and penalises unusual, aggressive behaviour. The remainder of the chapter is devoted to a thorough evaluation of this framework across two vehicle densities and against representative state-of-the-art schemes.

4.1.1 Motivation and Objectives

8 Vehicular Ad-hoc Networks are a cornerstone of the intelligent transportation system, supporting safety warnings, collision avoidance, and cooperative awareness among vehicles over the IEEE 802.11p standard [1]. Their defining property, the high mobility of the nodes, is also their principal weakness, because it causes the network topology to change continually and forces the frequent reconstruction of clusters. Clustering imposes a manageable, scalable structure on this volatile network [4], yet the benefit of clustering is only realised if the elected cluster heads are stable, since every re-election generates signalling overhead and interrupts communication. Conventional partitioning methods such as k-means struggle in this setting because they assume a fixed number of spherical clusters, whereas spectral techniques, which favour data connectivity over geometric proximity, are far better matched to the evolving graph structure of a VANET [8,9,10].

The motivation for the present work stems from a specific limitation of existing stable-clustering schemes. Stability has traditionally been pursued through metrics such as relative speed, connectivity, link lifetime, and trust, all of which describe the state of a vehicle at a single instant. What these metrics omit is the dynamic behaviour of the driver, that is, whether the vehicle habitually behaves in a steady and cooperative manner or in an erratic and aggressive one. High variance in a vehicle speed is a reliable indicator of aggressive driving, and such vehicles are the most likely to disrupt intra-cluster communication [11]. A cluster head drawn from this population is short-lived by nature, so incorporating a behavioural judgement into the selection process is a direct route to greater stability. The idea of relativity, in which each metric is adjusted with reference to the surrounding vehicles rather than taken at face value, provides the means to encode this judgement [3, 13, 87].

Against this background, the objectives of this chapter are fourfold. The first objective is to describe the modified clustering framework and its incentive-based cluster-head selection mechanism in sufficient detail to make the evaluation reproducible. The second is to assess the quality of the clusters that the mTTM pipeline forms and the stability of the cluster heads that the ANB-I mechanism elects, examined cluster by cluster. The third is to quantify the communication performance of the

framework in terms of throughput and packet delay across a sparse network of one hundred vehicles and a dense network of one thousand vehicles. The fourth is to benchmark the framework against representative hypergraph-based and eigen-based schemes and to characterise its scalability and overhead, so that its suitability for real-world deployment can be judged.

4.1.2 Chapter Organization

The chapter is organised as follows. Section 4.2 presents the modified incentive-based clustering framework, covering both the mTTM architecture and the ANB-I cluster-head selection mechanism together with the adaptive behavioural metrics and their governing equations. Section 4.3 describes the experimental methodology, including the simulation environment, the network configuration, and the metrics used for evaluation. Section 4.4 analyses the efficiency of cluster formation and the stability of the resulting clusters. Section 4.5 evaluates the incentive-based cluster-head selection in detail, comparing the adaptive metrics against their conventional counterparts and examining cluster-head lifetime. Section 4.6 reports the communication performance in terms of throughput, packet delivery, and delay. Section 4.7 provides a comparative evaluation against existing schemes and quantifies the improvement achieved. Section 4.8 examines scalability and overhead, Section 4.9 discusses the key findings and their practical implications, and Section 4.10 summarises the chapter.

4.2 Modified Incentive-Based Clustering Framework

The framework operates in two sequential phases. The first phase forms the clusters using a modified spectral pipeline, and the second phase elects a stable cluster head within each cluster using the incentive-based mechanism. The clustering phase reuses the hypergraph model and the modularity-based spectral machinery established in Chapter 3, modifying only the assignment stage, whereas the selection phase is new and constitutes the primary contribution of this chapter.

4.2.1 Framework Architecture

The vehicular network is represented as a weighted hypergraph in which the vertices are the vehicles and the hyperedges capture the multi-vehicle communication

relationships within transmission range, exactly as defined in Chapter 3. Rather than describing this hypergraph by its raw adjacency matrix, the framework uses the modularity matrix, which measures how strongly a pair of vehicles is connected relative to what would be expected by chance and thereby exposes the community structure that stable clustering depends upon [25, 88]. The construction of the modularity matrix and its interpretation are as given in Chapter 3 and are not repeated here.

The overall architecture, summarised in Figure 4.1, begins by selecting the time slot with the maximum number of participating vehicles, so that the partition is sized for the densest condition the network will encounter. The locations of the vehicles in that slot are gathered, the hypergraph is constructed, and the modularity matrix is formed. The optimal number of clusters is then fixed by the eigen-gap heuristic, which examines the spectrum of the Laplacian and selects the count corresponding to the widest gap between successive eigenvalues,

$$k = \arg \max(\delta_1, \delta_2, \dots, \delta_{n-1}) \quad (4.1)$$

where the differences are taken over the eigenvalues sorted in ascending order. This is the same criterion used in Chapter 3, and it removes the need to specify the number of clusters in advance, a well-known weakness of k-means in a topology that changes continually [30].

The distinctive element of the clustering phase is the Modified Tensor-Trace-Maximisation procedure. As in Chapter 3, the spectral embedding is derived from a modularity-based Laplacian rather than an adjacency-based one, so that the community structure is preserved through the spectral stage. The diagonal matrix required for the normalisation is obtained by traversing the hypergraph and summing, for each vehicle, the modularity contributions of its one-hop neighbours,

$$Dig_{ij} = \sum_{j=1}^N Q_{ij} \quad (4.2)$$

and the symmetrically normalised, modularity-based Laplacian is then formed as

$$L = Dig^{-1/2} Q Dig^{-1/2} \quad (4.3)$$

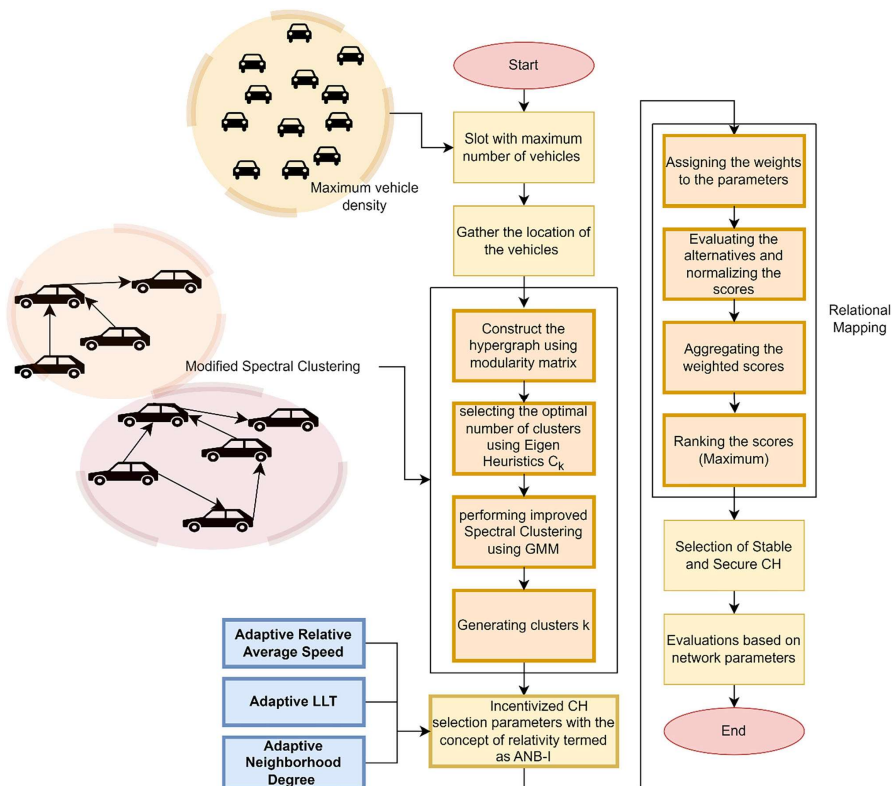


Figure 4.1 Architecture of the modified incentive-based clustering framework

The essential modification relative to a conventional spectral method lies in the assignment stage. Where k-means would partition the spectral embedding by nearest-centroid assignment, and would thereby impose spherical, equally sized clusters, the mTTM pipeline assigns vehicles with a Gaussian Mixture Model. The mixture describes the embedding as a weighted combination of Gaussian components, each with its own mean and covariance, so that clusters of arbitrary orientation and spread can be represented. This probabilistic, covariance-aware assignment models the true distribution of vehicles far more faithfully than a hard, distance-based partition, and it is the reason the modified pipeline adapts well to the irregular geometry of an urban road network [5, 81, 89, 90]. The complete clustering procedure is set out in Algorithm 4.1.

Algorithm 4.1 , Hypergraph-based modified TTM (mTTM) clustering

Input: vehicle information, communication relationships, mobility parameters, simulation time

1. Select the time slot with the maximum number of participating vehicles.
2. Gather vehicle locations and construct the weighted hypergraph $H = (E, V, w)$.
3. Generate the similarity / adjacency matrix from inter-vehicular distance proximity.

4. Construct the modularity matrix Q of the hypergraph (Chapter 3).
5. Compute the diagonal degree matrix Dig using Equation 4.2.
6. Construct the modularity-based Laplacian L using Equation 4.3.
7. Perform eigen-decomposition of L and sort the eigenvalues in ascending order.
8. Determine the optimal number of clusters k via the eigen-gap heuristic (Equation 4.1).
9. Normalise the spectral-feature matrix obtained from the eigenvectors.
10. Assign vehicles to the k clusters using GMM-based probabilistic clustering.

Output: optimally formed vehicle clusters

4.2.2 Incentive-Based Cluster Head Selection Mechanism

Once the clusters are formed, each cluster elects a cluster head through the Adaptive Neighboring Behavior-Incentivization mechanism. The mechanism rests on a single principle: the metrics used to rate a vehicle should not be taken at their raw values but should be adjusted according to whether the vehicle is behaving usually or unusually relative to its neighbours, and vehicles that behave steadily should be rewarded with a greater chance of becoming cluster head.

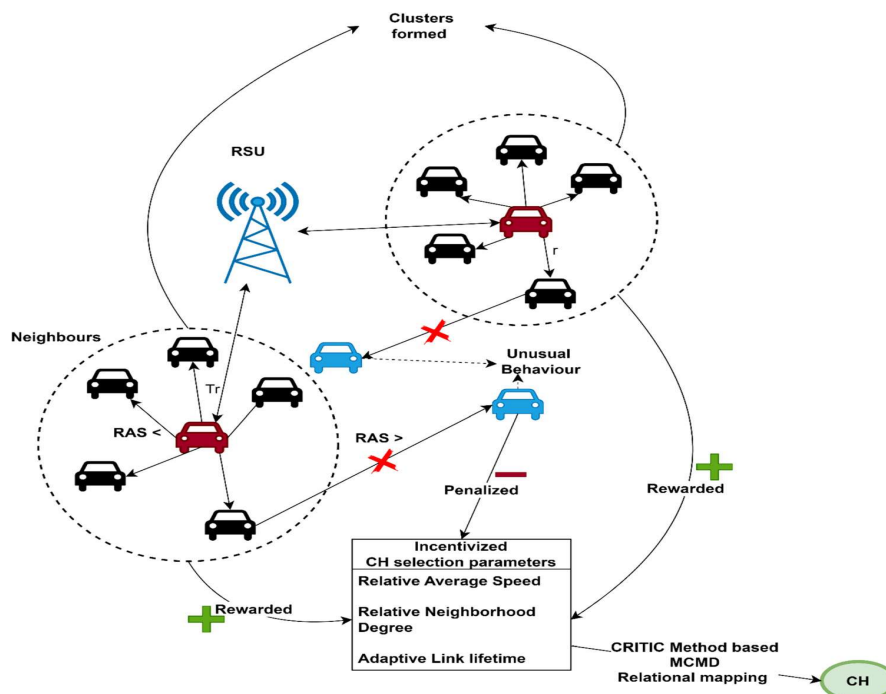


Figure 4.2 Incentivized cluster-head selection process

This behavioural adjustment is what distinguishes the present selection process from the purely instantaneous criteria of Chapter 3. The complete process is illustrated in Figure 4.2. At the heart of the mechanism is a relativity rule that rewards or penalises each candidate metric according to the difference between its expected and its observed value. For a metric with initial value denoted below, a small constant is added when the difference falls within a threshold, indicating steady behaviour, and subtracted when the difference exceeds the threshold, indicating erratic behaviour,

$$Adjus_{CHmetrics_i} = \begin{cases} Ini_{CHmetrics_i} + \delta, (if\ diff \leq Thr) \\ Ini_{CHmetrics_i} - \delta, (if\ diff > Thr) \end{cases} \quad (4.4)$$

where the threshold is set dynamically as a multiple of the standard deviation of the metric, so that the notion of usual behaviour adapts to the prevailing conditions of each cluster rather than being fixed a priori. Three adaptive metrics are computed for every candidate vehicle under this rule.

The first metric is the Adaptive Relative Average Speed. To promote non-aggressive driving and a steady speed, the difference between the current speed of a vehicle and the average speed of the cluster is fed to the relativity rule, so that vehicles holding a speed close to the cluster average are rewarded and become preferred candidates. Because it responds to the local average rather than to an absolute figure, the adaptive relative speed remains far more consistent across changing vehicle densities than the conventional relative-speed metric.

The second metric is the Adaptive Neighborhood Degree, defined as the number of genuine neighbours of a vehicle whose speed differences fall below the threshold. A neighbour that moves at a markedly different speed is unlikely to sustain a link and is therefore not counted, so the degree is updated by rewarding only the stable neighbours,

$$\begin{cases} RNd(t+1) = RNd(t) + \delta \times 1, & if\ diff < Thr \\ & else\ 0 \end{cases} \quad (4.5)$$

This yields a consistent and reliable set of neighbours for each vehicle and, because it too is governed by the relativity rule, it varies little as the vehicle density changes.

The third metric is the Adaptive Link Lifetime, which estimates how long a link between two vehicles will persist. It ensures a longer duration of link among the vehicles. Consider the vehicles A at position a_i and a_j with speeds v_i and v_j then to sustain communication link it must follow the condition, $-Tr \leq a_i - a_j \leq Tr$ (Tr is communication range). Hence, the LLT_{ij} is given as the ratio of differences in velocities ($D_v = v_i - v_j$) and difference in spatial coordinates ($D_a = a_i - a_j$). Considering the difference in the velocities and the difference in the spatial coordinates of a pair of vehicles constrained to lie within transmission range, the adaptive link lifetime is expressed as,

$$LLT_{adp} = \sqrt{(D_a)^2 + (D_v)^2} \cdot \sqrt{(D_v)^2 + \delta} + \frac{Tr}{D_v} \quad (4.6)$$

A small constant is added to the standard deviation when the threshold is adjusted, which prevents the number of neighbours from varying sharply. Where the relative speed is zero, and no explicit information is therefore available, a predefined speed of forty kilometres per hour is assumed so that the metric remains well defined. A longer adaptive link lifetime denotes a more durable connection and hence a more suitable cluster head.

The three adaptive metrics are collected into a decision matrix and reconciled into a single ranking through a Multi-Criteria Multi-Decision procedure based on the CRITIC method [91]. The metrics are first normalised, after which the Pearson correlation among them and the standard deviation of each are computed. A conflict measure for each metric is obtained by summing the complements of its correlations with the other metrics, capturing how much independent information the metric contributes. The objective weight of each metric then follows as,

$$w_i = \frac{\sigma_{CH_{metric_i}} \times r_{CH_{metric_i}}}{\sum_{j=1}^n \sigma_j \times r_j} \quad (4.7)$$

so that a metric receives more weight when it is both highly variable, and therefore discriminating, and weakly correlated with the others, and therefore non-redundant. Because the weights are derived from the data rather than assigned by hand, the decision is objective and fully reproducible, which is an advantage over the arbitrary

weighted-sum schemes common in the literature. The aggregate suitability of a candidate vehicle is the weighted sum of its normalised metrics,

$$W(V_{ID}) = \sum_{i=1}^{CH_{metric_i}} w_i \times Norm(V_{ID_i}) \quad (4.8)$$

and the vehicle attaining the maximum aggregate suitability within a cluster is elected as its cluster head,

$$CH = Max(W(V_{ID})) \quad (4.9)$$

The incentive mechanism enters the selection at every stage, because the metrics that feed the decision have already been rewarded or penalised according to the behaviour of each vehicle. In this way the elected cluster head is not merely the most connected or the fastest-linked vehicle at a single instant but the one that has demonstrated steady, cooperative behaviour, which is the property most conducive to a long cluster-head lifetime [90]. The full selection procedure is given in Algorithm 4.2.

Algorithm 4.2 , ANB-I based stable cluster-head selection

Input: optimally formed clusters, vehicle mobility and neighbour information

for each simulation instant t

for each cluster k

for each candidate vehicle j in cluster k

1. Compute Adaptive Relative Average Speed via the relativity rule (Equation 4.4).
2. Compute Adaptive Neighborhood Degree (Equations 4.4, 4.5).
3. Compute Adaptive Link Lifetime (Equations 4.4, 4.6).
4. Assemble the decision matrix from the three adaptive metrics.
5. Normalise the metrics and compute CRITIC objective weights (Equation 4.7).
6. Compute the aggregate suitability score (Equation 4.8).
7. Elect the vehicle of maximum suitability as cluster head (Equation 4.9).

end for end for end for

Output: stable cluster head per cluster per instant

4.3 Experimental Methodology

The framework was implemented and evaluated through a combination of network simulation and physical-layer modelling, using the same experimental basis established in Chapter 3 so that the two frameworks can be compared on equal terms.

This section describes the simulation environment, the network configuration, and the metrics adopted for the evaluation.

4.3.1 Simulation Environment and Network Configuration

The simulation of the proposed model is carried out using the MATLAB (R2023b), SUMO. The complete network model is designed and performance is evaluated through metrics like the throughput and Packet Delay in MATLAB. For the study, the area considered is at 28°39'39"N and 77°13'48"E location in Delhi. SUMO uses Openstreetmap roadways, which have 293 lanes and an average vehicle speed of 10.53 m/s and a maximum speed of 25 m/s, to simulate the traffic conditions at Connaught Place, Delhi. Delhi Capital of India is a city with huge traffic and is appropriate for an urban setting. The SUMO models traffic flow by utilizing the XML collection of roads and intersections from Openstreet map. The road-network extract of the city and the clusters formed at the instant of maximum vehicle density are shown in Figure 4.3 within Section 4.4.

Two vehicle densities were studied so that both sparse and dense conditions are represented: a sparse network of one hundred vehicles and a dense network of one thousand vehicles. The complete set of simulation parameters is listed in Table 4.1.

Table 4.1 Simulation parameters

Parameter	Value
Density of vehicles	100, 1000
Number of lanes	293
Speed of vehicles	25 m/s (max), 10.53 m/s (avg)
Simulation time	1500 s
Scenario	Urban (Connaught Place, Delhi)
Mobility / traffic generator	SUMO with OpenStreetMap

Communication between vehicles was modelled over a Rayleigh fading channel with Orthogonal Frequency Division Multiplexing, and the Doppler effect was included to represent the signal degradation caused by relative vehicle velocity. Random data bits were generated at fourteen megabits per second, subjected to subcarrier modulation and a thirty-two-point fast Fourier transform, shifted in frequency to account for relative velocity, encoded onto the subcarriers by phase-shift

keying, and finally corrupted by white and Rayleigh channel noise so that realistic channel conditions were reproduced. The wireless standard is IEEE 802.11p, operating in ten-megahertz channels within the 5.9 gigahertz band. The physical-layer parameters are summarised in Table 4.2.

Table 4.2 Specifications of the IEEE 802.11p physical layer

Attribute	Measure
Doppler shift	$V_{s,r} / \lambda$
Rate of samples	10 MHz
Number of pilot subcarriers	4
FFT point count	32
Subcarrier count	$52 + D$
Data subcarrier count	52
Coding scheme	1/2
Modulation scheme	OFDM
Rate of data	14 Mbps
Cyclic prefix	16

4.3.2 Performance Evaluation Metrics

Because energy and processing power are abundant in vehicular platforms, the evaluation concentrates on the quantities that govern communication quality under high mobility rather than on resource consumption. The assessment proceeds along two connected lines. The first is the stability of the clustering, expressed through the cluster-head stability, the frequency of change in cluster membership, and the number of clusters formed. Cluster-head stability is measured as the frequency with which a single vehicle is retained in the cluster-head role over the life of the simulation, so that a higher value denotes fewer re-elections and a steadier backbone. When the membership of a cluster changes more often, a robust algorithm should nevertheless preserve the cluster head, and this is exactly what the stability metric captures.

The second line is the communication performance, expressed through throughput and packet delay. Throughput is the rate at which packets are successfully delivered from source to destination vehicles, measured in kilobits per second, and a higher value indicates more effective use of the channel. Packet delay is the time taken for a packet

to reach its destination, which grows with hop distance, channel noise, and congestion, so a lower value is desirable and is especially important for the time-critical safety messages that VANETs are built to carry. These two metrics are reported for both vehicle densities and, where informative, cluster by cluster, so that the behaviour of the framework can be examined in detail rather than only in aggregate.

4.4 Cluster Formation and Stability Analysis

This section examines the first phase of the framework, assessing how efficiently the modified pipeline forms clusters and how stable those clusters and their heads prove to be under mobility.

4.4.1 Cluster Formation Efficiency

Applying the eigen-gap heuristic at the instant of maximum density consistently identified three as the optimal number of clusters for the Connaught Place scenario, and this value was then held fixed for the remainder of each simulation. Fixing the count in this way is deliberate: it prevents the continual re-clustering that would otherwise accompany the natural rise and fall of vehicle density, and re-clustering is one of the largest sources of instability in a mobile network. A count of three also strikes a sensible balance, since too many clusters raise the hand-off cost while too few lead to more frequent link disconnections.

Figure 4.3 shows the road-network extract of Connaught Place alongside the three clusters generated at the instant of maximum vehicle density, with the clusters distinguished by colour and the roadside units marked by triangles. The partition follows the spatial grouping of the traffic across the mapped area, which confirms that the modularity-based, GMM-assigned pipeline captures the community structure of the network rather than imposing an artificial geometry. Because the assignment is probabilistic and covariance-aware, the clusters conform to the elongated, irregular shapes that arise naturally along an urban road layout, an outcome a nearest-centroid method could not achieve.

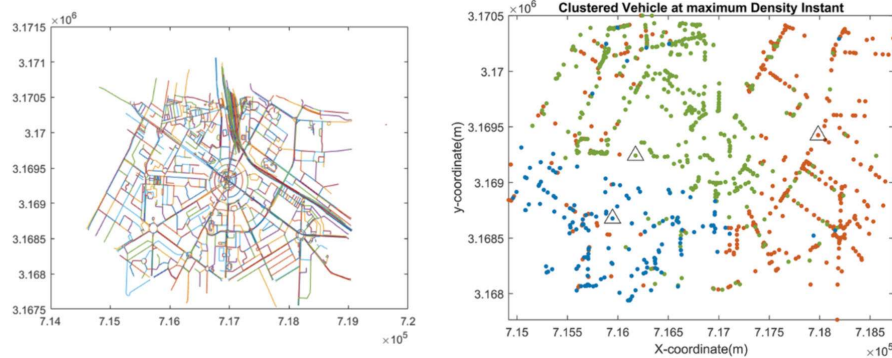


Figure 4.3 Map of Delhi (Connaught Place) and the three clusters formed at the instant of maximum vehicle density

4.4.2 Cluster Stability Assessment

Cluster-head stability was evaluated cluster by cluster, comparing the full incentive-based model against the individual metrics used in isolation, first for the sparse network of one hundred vehicles. Figure 4.4 reports the cluster-head stability frequency across the three clusters. In the third cluster the full model attains the highest stability of all the schemes, confirming that the incentive process succeeds in prioritising vehicles that exhibit steady behaviour.

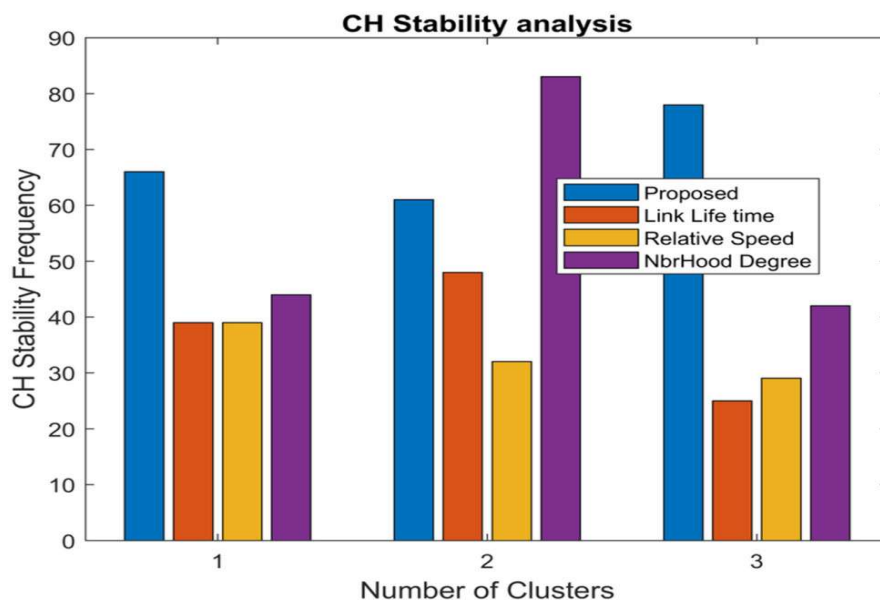


Figure 4.4 Cluster-wise evaluation of cluster-head stability for 100 vehicles: the proposed model against the individual incentivized selection metrics

The relative-speed metric on its own yields the lowest stability over the three clusters, while the neighbourhood-degree metric, once the relativity adjustment is applied, achieves high stability in the second cluster. The dynamic nature of the relativity rule, whose threshold is computed from the standard deviation within each cluster, allows the mechanism to adapt to clusters of differing character, and it is this adaptability that underlies the robustness of the model across diverse conditions.

The same assessment for the dense network of one thousand vehicles is shown in Figure 4.5. The proposed model again performs strongly, delivering high stability in the first and second clusters. In the third cluster the adaptive neighbourhood-degree metric slightly exceeds the full model, a consequence of the incentive mechanism prioritising vehicles with usual behaviour under a particular local configuration, but this single discrepancy does not detract from the overall advantage of the combined model. Across both densities the emphasis on incentivized, behaviour-aware selection yields a steadier cluster head, and the reward-and-penalty rule aligns with the goal of distinguishing usual from unusual driving so that stable vehicles are consistently favoured.

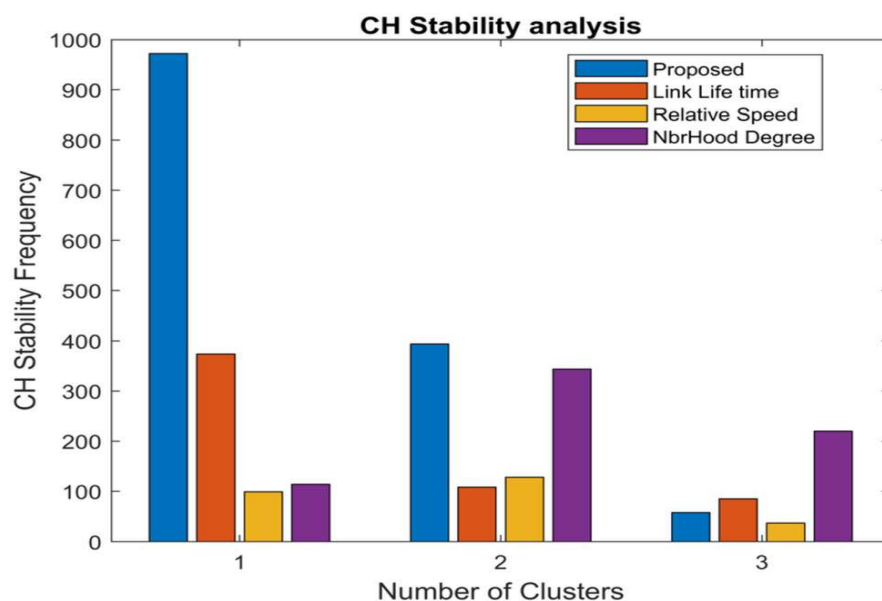


Figure 4.5 Cluster-wise evaluation of cluster-head stability for 1000 vehicles: the proposed model against the individual incentivized selection metrics

4.5 Cluster Head Selection Performance Evaluation

This section examines the behaviour of the three adaptive metrics that drive the incentive-based selection, and then considers the lifetime and leadership quality of the elected cluster heads.

4.5.1 Incentive-Based CH Selection Analysis

The value of the incentive mechanism is most clearly seen by comparing each adaptive metric against its conventional counterpart across varying vehicle density. The Adaptive Relative Average Speed is examined in Figure 4.6. Compared with the conventional relative-speed parameter, the adaptive version shows markedly less variation as the vehicle density changes, because it is measured relative to the cluster average and adjusted by the relativity rule. Since high variance in speed is the signature of aggressive driving and of the disruptions it causes, the steadiness of the adaptive metric translates directly into a more dependable basis for selecting a cluster head.

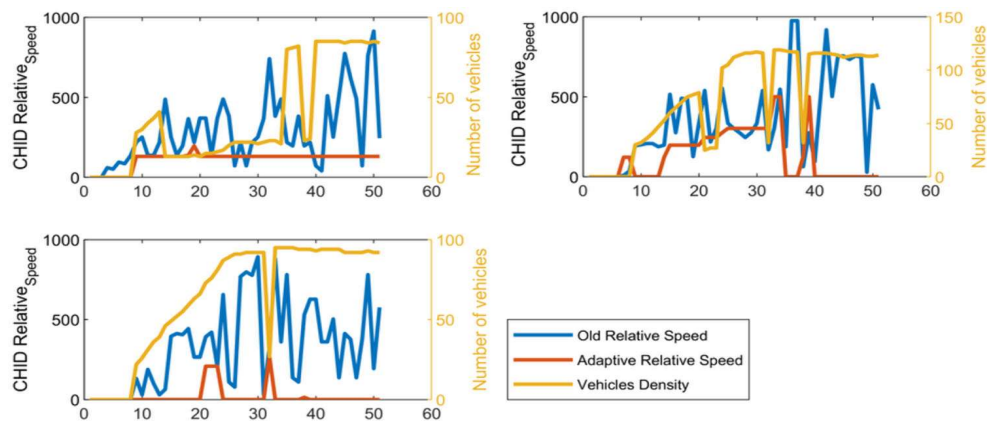


Figure 4.6 Evaluation of the proposed incentivized Adaptive-RAS parameter

The Adaptive Neighborhood Degree is examined in Figure 4.7. By counting only those neighbours whose speed differences fall below the threshold, and by rewarding vehicles that maintain such stable neighbours, the adaptive degree exhibits considerably less variance than the conventional neighbourhood degree over the range of densities. A vehicle with a consistent set of reliable neighbours is a natural candidate for a durable cluster head, so this steadiness again supports stability.

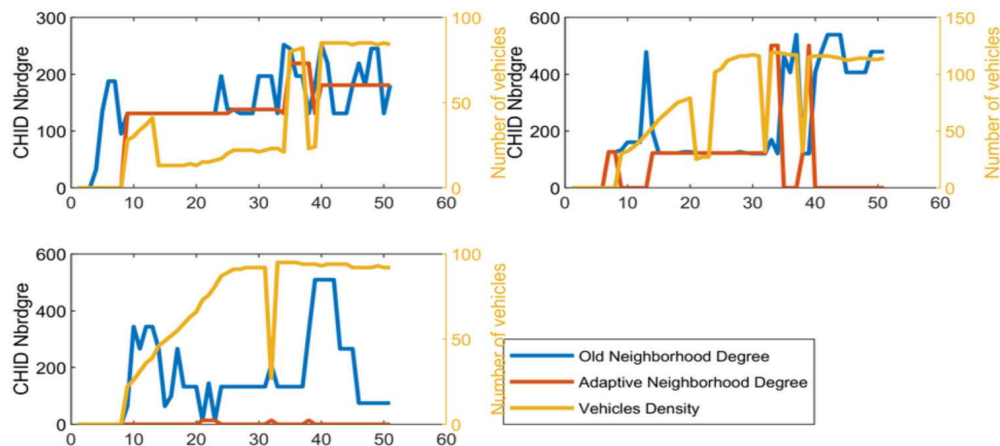


Figure 4.7 Evaluation of the proposed incentivized Adaptive-ND parameter

The Adaptive Link Lifetime is examined in Figure 4.8. The adaptive formulation, which augments the standard deviation with a small constant and assigns a defined value when the relative speed is zero, keeps the estimated link lifetime stable over time without sharp fluctuations. This indicates that the vehicles within a cluster maintain relatively durable links, and vehicles with such characteristics are precisely those that the incentive mechanism rewards and elevates towards leadership.

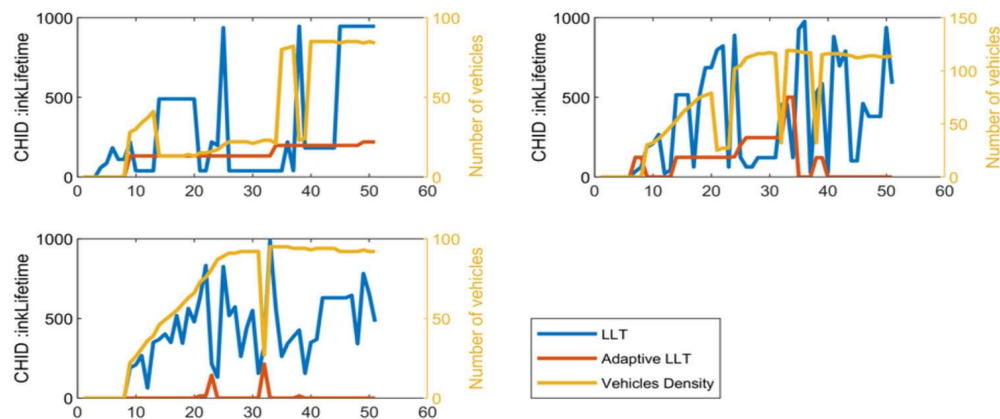


Figure 4.8 Evaluation of the proposed incentivized Adaptive-LLT

Finally, the combined ANB-I model is compared against the individual metrics in Figure 4.9. The full model, which fuses the three adaptive metrics through the CRITIC weighting, shows greater consistency over varying vehicle density than any single metric acting alone. This is the expected outcome of an objective multi-criteria decision: by giving more influence to metrics that are both discriminating and non-redundant, the weighted combination balances the strengths of the individual criteria

and avoids the blind spots of any one of them. The incentive layer, which rewards usual behaviour and penalises unusual behaviour before the weighting is applied, is what allows this balanced decision to track the real-time conduct of the vehicles.

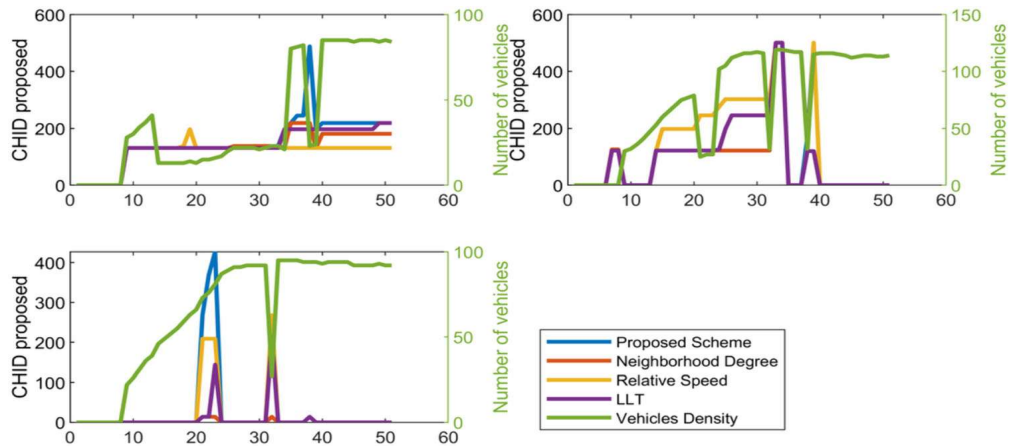


Figure 4.9 Evaluation of the proposed incentivized ANB-I model against the individual selection metrics over varying vehicle density

4.5.2 CH Lifetime and Leadership Performance

The purpose of a stable cluster head is to serve as a durable virtual backbone, coordinating intra-cluster communication and relaying towards the roadside unit without frequent re-election. The leadership quality of the elected heads follows directly from the design of the selection mechanism. Because the incentive rule rewards steady speed, reliable neighbours, and durable links, the vehicle that attains the maximum aggregate suitability is, by construction, the one whose behaviour is most conducive to a long tenure. The stability results of Section 4.4 confirm this: the frequency with which a single vehicle is retained as cluster head is highest for the full model in the majority of clusters and densities, which is the operational meaning of a long cluster-head lifetime.

The dependence of cluster-head lifetime on the stability of the cluster members is also captured by the mechanism. A cluster head can only remain in office while its members stay within range, so by rewarding vehicles whose neighbours exhibit low speed variance the Adaptive Neighborhood Degree indirectly protects the tenure of the head. This coupling between member stability and leadership stability is a deliberate feature of the behaviour-aware design, and it explains why the incentive-based model

sustains leadership more effectively than schemes that judge a candidate solely on its own instantaneous attributes.

4.6 Communication Performance Analysis

Having established the stability of the clustering, the evaluation turns to the communication performance of the framework, assessed through throughput and packet delay for both vehicle densities.

4.6.1 Packet Delivery and Throughput Analysis

Throughput for the sparse network of one hundred vehicles is shown in Figure 4.10. The proposed model maintains high throughput over the second and third clusters relative to the individual metrics, reflecting the fact that a stable cluster head with reliable members sustains a cleaner channel and delivers more packets successfully. The adaptive link-lifetime metric, used in isolation, produces the highest throughput in the first cluster, which is consistent with its emphasis on durable links, but the combined model provides the more balanced outcome across the clusters as a whole.

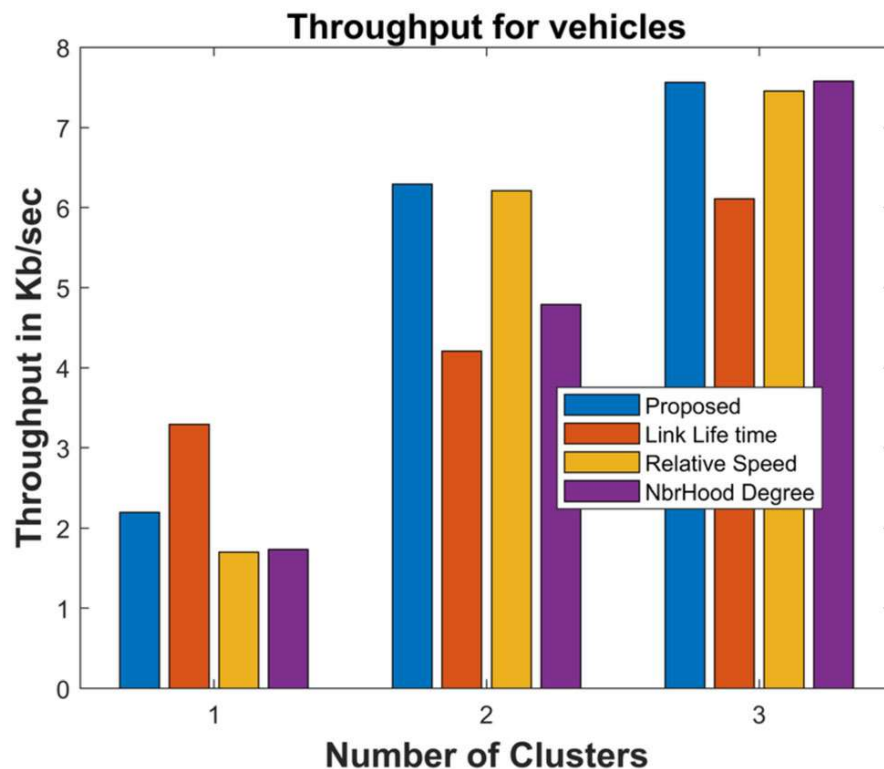


Figure 4.10 Cluster-wise evaluation of throughput for 100 vehicles.

Throughput for the dense network of one thousand vehicles is shown in Figure 4.11. Here the proposed model delivers high throughput across all three clusters, which indicates that at high density the incentivized selection contributes consistently to channel efficiency. The interaction between the reward-and-penalty rule and the cluster-head metrics proves effective under the more demanding conditions of a congested network, where a stable backbone is even more valuable because contention for the channel is greater. The consistency of the throughput advantage as the density rises is a strong indication of the scalability of the approach.

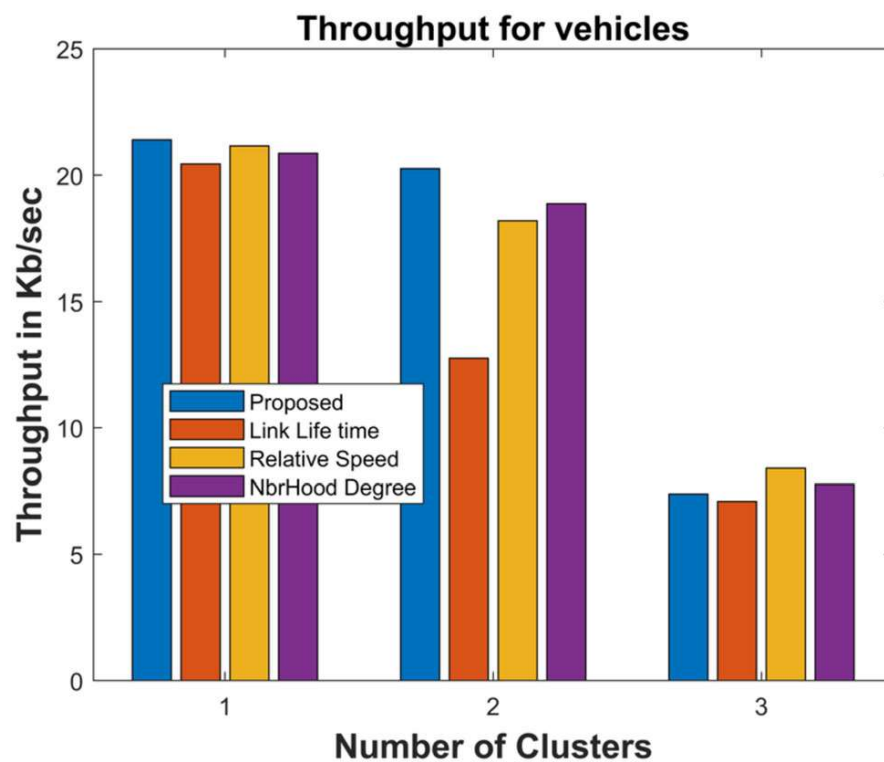


Figure 4.11 Cluster-wise evaluation of throughput for 1000 vehicles

4.6.2 Delay and Communication Reliability Analysis

Low packet delay is critical for VANET safety applications, since a delayed collision warning is of little use. Packet delay for the sparse network of one hundred vehicles is shown in Figure 4.12. The proposed model achieves low packet delay across the clusters, although the adaptive link-lifetime metric, while producing high throughput, introduces a higher delay in the first cluster. This reveals a trade-off between maximising link duration and minimising delay, and it is exactly the kind of

trade-off that the balanced, weighted decision of the full model is designed to manage rather than to resolve in favour of a single criterion.

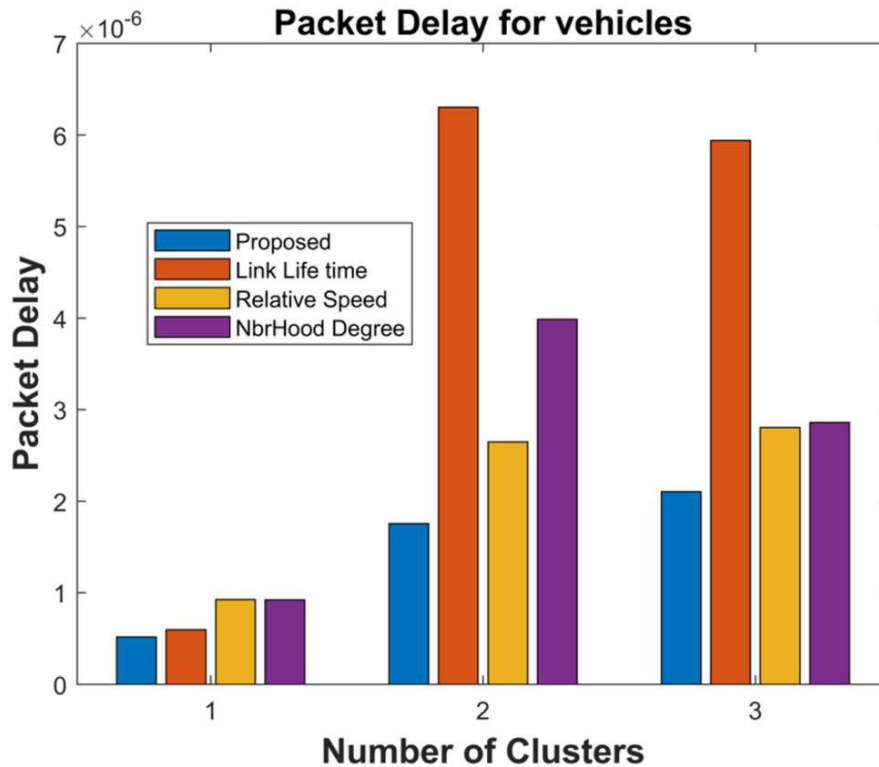


Figure 4.12 Cluster-wise evaluation of packet delay for 100 vehicles

Packet delay for the dense network of one thousand vehicles is shown in Figure 4.13. In contrast to the variation seen in the sparse case, the proposed model consistently maintains low packet delay across all three clusters at high density. This concurrent improvement in delay and stability underlines the positive effect of the incentive-based selection on overall network performance, and it demonstrates that the framework scales to a congested network without sacrificing the timeliness on which safety communication depends. Taken together, the throughput and delay results show that the framework combines efficient delivery with low latency across both the sparse and the dense regimes.

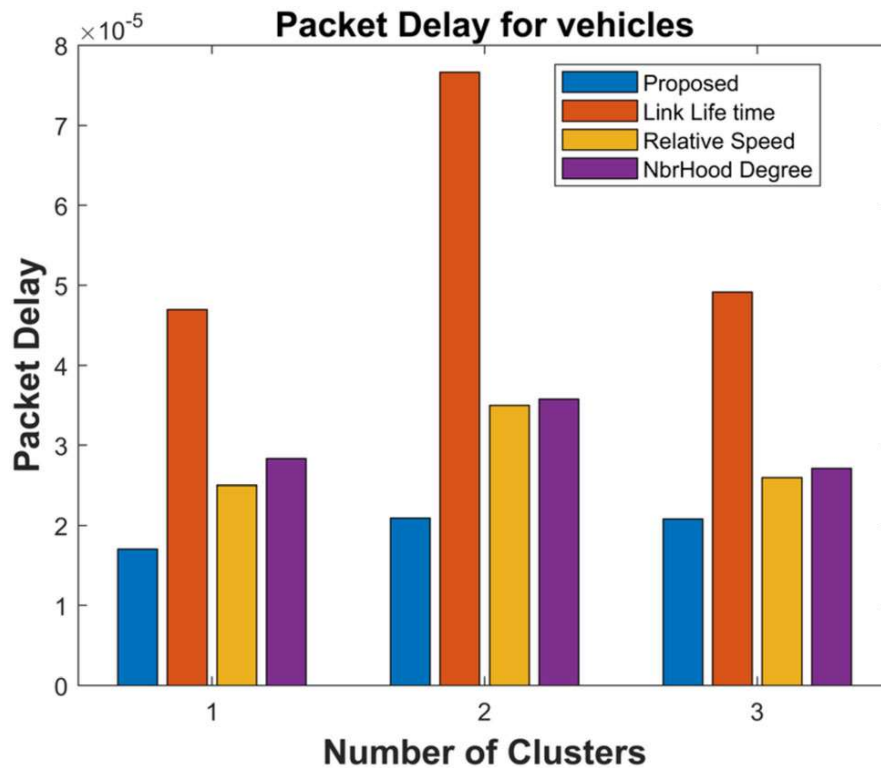


Figure 4.13 Cluster-wise evaluation of packet delay for 1000 vehicles

4.7 Comparative Performance Evaluation

To place the framework in context, it was benchmarked against representative state-of-the-art schemes in terms of both cluster-head stability and network performance. The comparison set comprises the hypergraph-based clustering model HGCM [10], the junction-based clustering scheme JCV [22], and the cluster-based VANET-oriented evolving graph CVoEG [30], each of which represents a distinct strand of the clustering literature.

4.7.1 Comparison with Existing Clustering Approaches

HGCM shares the hypergraph representation used here but selects cluster heads by a fuzzy decision over static criteria, and its cluster count varies with vehicle density. CVoEG divides the network into an optimal number of clusters using the eigen-gap heuristic and elects a head by maximum eigen-centrality, taking link reliability as its guiding criterion. JCV addresses stability at road junctions through the intelligent use of mobility parameters such as transmission range, direction at the next junction, and

vehicle density. The cluster-head stability and the number of clusters produced by each scheme are compared in Table 4.3.

Table 4.3 State-of-the-art comparison based on cluster-head stability

Reference	Method	CH stability (%)	Optimal clusters
Proposed model	Hypergraph clustering with ANB-I CH selection	75	3
HGCM [5]	Hypergraph clustering model	72	4
JCV [18]	Junction-based clustering	76	16
CVoEG [15]	Cluster-based evolving graph	65.5	20

The comparison shows that the proposed model attains a cluster-head stability of 75% using only three clusters. This exceeds the 72% of HGCM, and it does so while forming fewer clusters, which is a decisive practical advantage because each additional cluster raises the hand-off and management burden. Although JCV records a marginally higher stability of 76%, it requires 16 clusters to achieve it, and CVoEG, despite using 20 clusters, reaches only 65.5%. The proposed model therefore delivers competitive or superior stability with a small fraction of the clusters, which is exactly the balance a scalable urban deployment requires.

The network performance was compared in terms of packet delay, the quantity most critical to the safety applications that VANETs support. The results are given in Table 4.4.

Table 4.4 State-of-the-art comparison based on network performance (packet delay)

Reference	Method	Packet delay (s)
Proposed model	Hypergraph clustering with ANB-I CH selection	0.2071
HGCM [10]	Hypergraph clustering model	0.42
JCV [22]	Junction-based clustering	0.27
CVoEG [30]	Cluster-based evolving graph	0.34

The proposed model records the lowest packet delay of all the schemes at 0.2071 seconds, well below the delays of HGCM, JCV, and CVoEG. Because low delay

ensures that safety messages reach their destination promptly, this result confirms that the framework improves the quality of service in addition to the stability of the clustering.

4.7.2 Performance Improvement Assessment

Quantifying the gains, the proposed model achieves a 4.17% improvement in cluster-head stability over HGCM while using fewer clusters. Relative to CVoEG and JCV, the combination of hypergraph clustering and ANB-I based selection contributes stability improvements of 14.5 % and 1.33 % respectively, again with far fewer clusters than either baseline requires. The advantage is even more pronounced for packet delay, where the model delivers a substantial reduction relative to HGCM and clear reductions relative to CVoEG and JCV. The consistent pattern across both stability and delay is that the framework matches or beats the best baseline on the headline metric while being markedly more economical in the number of clusters, which is the property that most directly governs overhead and scalability.

The source of these gains is the behaviour-aware, incentive-based selection. Where the baselines rate candidates on static or purely mobility-derived criteria, the proposed mechanism accounts for the real-time conduct of each driver and rewards the steady, cooperative vehicles that are best suited to a durable leadership role. It is this dynamic and incentive-driven aspect of the selection that allows the model to hold competitive stability with fewer clusters, striking an effective balance between stability and scalability.

4.8 Scalability and Overhead Analysis

A framework intended for real urban deployment must remain effective as the number of vehicles grows and must not impose an undue burden of computation or signalling. This section assesses both aspects.

4.8.1 Scalability under Different Traffic Conditions

The evaluation across the sparse and dense networks provides direct evidence of scalability. The transition from 100 to 1000 vehicles is a tenfold increase in load, yet the framework maintained high throughput across all clusters and consistently low packet delay in the dense case, in several respects performing more uniformly at high

24 density than at low. This behaviour is a consequence of two design decisions. The first is that the optimal cluster count is fixed once, at maximum density, so the partition is already sized for the heaviest load and no costly re-clustering is triggered as traffic intensifies. The second is that the recurring per-instant work of the selection phase grows only with the number of vehicles and the small, constant number of criteria, so the cost of electing cluster heads scales gently with density. The result is a framework whose stability and communication performance are preserved as the network grows, which is the essential requirement for a congested urban setting.

The probabilistic assignment of the mTTM pipeline reinforces this scalability. Because the Gaussian-mixture stage models clusters of arbitrary shape and spread, the partition continues to reflect the true distribution of vehicles as density rises and the geometry of the traffic becomes more intricate, rather than degrading as a rigid, distance-based partition would. The framework therefore adapts to the changing structure of the network instead of forcing the network into a fixed structure.

4.8.2 Communication and Computational Overhead

The communication overhead of a clustered VANET is dominated by the signalling that accompanies cluster-head re-election and cluster reorganisation, so a stable cluster head is the most effective means of reducing it. By electing heads that are behaviourally suited to a long tenure, the incentive mechanism directly lowers the frequency of re-election and, with it, the signalling overhead, which is reflected in the improved throughput reported earlier. The economy in the number of clusters compounds this benefit, since fewer clusters mean fewer heads to maintain and fewer hand-offs to coordinate as vehicles move between them.

The computational overhead is concentrated in the clustering phase, where the eigen-decomposition of the Laplacian is the most demanding operation. Crucially, this computation is performed only once, at the instant of maximum density, after which the cluster count is reused for the whole simulation, so its cost is amortised over the entire run rather than repeated at every step. The per-instant selection phase, by contrast, involves only the computation of three adaptive metrics and a closed-form CRITIC weighting for each candidate, both of which are inexpensive. This

concentration of heavy computation in a single off-line pass, with lightweight recurring work, keeps the framework practical for real-time operation even in the dense regime.

4.9 Discussion

4.9.1 Key Findings

The evaluation supports several clear conclusions. First, fixing the number of clusters at three through the eigen-gap heuristic and holding it for the whole simulation produced a stable partition and eliminated the churn associated with continual re-clustering. Second, replacing the nearest-centroid assignment of k-means with a Gaussian-mixture assignment allowed the clusters to follow the irregular geometry of the urban road network, which is visible in the spatial coherence of the partition at maximum density. Third, and most importantly, the incentive-based selection of cluster heads, driven by the three adaptive behavioural metrics and combined through an objective CRITIC weighting, delivered a cluster-head stability of 75% with only three clusters, an improvement of 4.17% over the hypergraph baseline and a competitive result against schemes that require many more clusters. Fourth, these stability gains were accompanied by high throughput and low packet delay across both the sparse and the dense networks, confirming that the behavioural approach benefits communication quality as well as structural stability.

A recurring theme in the results is the value of adaptivity. Each of the three metrics, when rendered adaptive through the relativity rule, showed markedly less variation across changing density than its conventional counterpart, and the combined model was steadier still. The single instance in which an individual metric slightly outperformed the full model, in the third cluster of the dense network, is the exception that illustrates the rule: the weighted combination trades a small loss in one configuration for consistent strength across all the others, which is the behaviour a general-purpose selection mechanism should exhibit.

4.9.2 Practical Implications for VANET Deployment

For a real deployment, the practical significance of these findings lies in the combination of competitive stability with a small number of clusters. Because

overhead and hand-off cost scale with the number of clusters and the frequency of re-election, a scheme that holds 75% stability with 3 clusters is far more deployable in a dense city than one that reaches a similar figure with 16 or 20 clusters. The low packet delay is equally consequential, since the safety applications that justify vehicular networking, from collision alerts to hazard warnings, depend on the timely arrival of messages, and the framework preserves this timeliness even under heavy load.

The behaviour-aware design also has implications beyond the metrics measured here. By explicitly distinguishing steady from erratic drivers and steering leadership towards the former, the framework aligns the structure of the network with the reliability of its participants, which is a desirable property for any safety-critical system. A natural direction for future work, consistent with the trajectory of the wider field, is to extend the framework towards the Internet of Vehicles and to explore edge computing and edge intelligence so that behavioural assessment can be performed closer to the source, supporting vehicle-to-vehicle and vehicle-to-everything communication at scale.

4.10 Chapter Summary

This chapter presented and evaluated a modified incentive-based clustering framework for reliable VANET communication, building on the hypergraph and spectral foundation established in Chapter 3. The framework couples a Modified Tensor-Trace-Maximisation clustering pipeline, which uses a Gaussian-mixture assignment in place of k-means, with an Adaptive Neighboring Behavior-Incentivization mechanism that selects cluster heads from three adaptive behavioural metrics, the Adaptive Relative Average Speed, the Adaptive Neighborhood Degree, and the Adaptive Link Lifetime, reconciled through an objective CRITIC weighting.

The evaluation, conducted in MATLAB and SUMO over an urban Connaught Place scenario for sparse and dense networks of one hundred and one thousand vehicles, showed that the framework forms three stable clusters, elects cluster heads that are demonstrably more durable than those chosen by individual metrics, and achieves a cluster-head stability of 75% while using far fewer clusters than competing schemes. It improved cluster-head stability by 4.17% over the hypergraph baseline,

remained competitive with junction-based and evolving-graph schemes at a fraction of their cluster count, and delivered the lowest packet delay together with high throughput across both densities. These results establish that making cluster-head selection sensitive to driver behaviour, through an incentive mechanism grounded in adaptive metrics, is an effective route to stable and reliable vehicular communication. The next chapter builds on this foundation by turning to the security dimension of the cluster-head selection process.

CHAPTER 5

RELIABLE AND SECURE COMMUNICATION IN VANETs USING GRAPH-BASED DEEP LEARNING

5.1 Introduction

Intelligent transportation systems increasingly rely on vehicular ad hoc networks to exchange safety, mobility, and traffic management information among vehicles and the roadside infrastructure. Smart and semi-autonomous vehicles continuously generate and consume large volumes of data that support road safety, traffic optimization, and data-driven driving decisions [92]. At the core of this ecosystem lies the connection between vehicles and RSUs, which forms the communication backbone of the road infrastructure and helps ITS reduce the likelihood of road incidents [92]. However, the same open wireless medium that enables this connectivity also widens the attack surface, allowing adversaries to inject malicious traffic, manipulate exchanged information, or exhaust the resources of vehicles and RSUs [93].

Among the various threats, denial of service (DoS) is particularly damaging because it introduces unwanted traffic that prevents legitimate nodes from accessing network services [93], [94]. Distributed denial of service (DDoS) escalates the problem by coordinating requests from many locations at once, overwhelming the RSU and the vehicles it serves [95]. Because VANETs were not originally engineered with comprehensive security in mind, conventional protections such as static access control and encryption schemes are poorly matched to the mobility, scale, and resource constraints of vehicular environments [95]. Consequently, dynamic and reactive defences such as intrusion detection systems (IDS) have attracted considerable attention as a means of preserving reliable and secure communication in VANETs.

This chapter presents a graph-based deep learning framework, referred to as the knowledge graph based deep convolutional neural network, that is deployed at the RSU to distinguish legitimate access requests from malicious ones. The framework couples a knowledge graph and a graph convolutional network (GCN) for semantic feature extraction with an ensemble of convolutional neural network (CNN) learners for robust classification. The remainder of the chapter sets out the requirements for

reliable and secure communication, reviews the security challenges and existing detection approaches in VANETs, describes the proposed framework in detail, presents the experimental setup and results, and compares the framework with the state of the art.

Within the vehicular scenario adopted here, vehicles are organized into clusters, and each cluster is coordinated by a cluster head that aggregates the access requests of its members and relays them toward the RSU-hosted IDS. Because the cluster head is the aggregation and relay point on which every member depends, its reliability is decisive for cluster-wide communication. In line with the research objective of analyzing the reliability of the cluster head under the proposed model, this chapter not only measures detection accuracy but also derives, from the per-class detection behaviour of the IDS, quantitative indicators of how reliably the cluster head sustains legitimate service while containing malicious traffic.

5.1.1 Reliable and Secure Communication in VANETs

Reliable communication is a prerequisite for the safety-critical services that VANETs are expected to deliver. Vehicles routinely broadcast location updates, road-condition reports, and emergency notifications, and the value of this information collapses if messages are delayed, dropped, or corrupted. Ensuring both the availability and the integrity of the exchanged data is therefore essential [92]. In this work the RSU is treated as a semi-trusted entity that is distributed along the roadway to facilitate communication, mediate access to a central server, and relieve individual vehicles of part of their computational and storage burden.

Frequent data exchange, although beneficial for traffic efficiency, amplifies the risk of DoS at the RSU through a concrete causal chain. Modern RSUs are provisioned to process a high volume of safety and traffic messages, often exceeding hundreds of beacons per second under dense traffic [30], [96]. When each vehicle already transmits periodic cooperative awareness messages, the RSU may be handling thousands of packets per second before any attack begins [96]. Under such conditions, even a modest injection of a few hundred malicious packets per second, whether through packet flooding, Sybil-style requests, or channel jamming, is sufficient to push processing and bandwidth utilization beyond safe thresholds and to trigger packet loss

and processing delays [30]. This combination of a high legitimate baseline load and a low attacker overhead explains why DoS attacks are more feasible and more harmful in vehicular settings than in conventional wireless networks. Preserving reliable communication in this environment thus requires an IDS that can identify malicious activity quickly and accurately, so that legitimate service continues uninterrupted while hostile traffic is blocked at the RSU.

5.1.2 Security Challenges and Intrusion Detection in VANETs

Several characteristics of VANETs combine to make their security uniquely challenging. First, the wireless and open nature of vehicular communication exposes nodes to eavesdropping, message modification, and injection, allowing attackers to disrupt or exhaust the resources of both vehicles and the network [93], [94]. Second, high node mobility and rapidly changing topologies mean that heavyweight cryptographic and policy-based countermeasures, which assume relatively stable relationships between entities, are difficult to apply and quickly become outdated [95]. Third, the centralized role of the RSU makes it an attractive single point of attack, since a successful DoS or DDoS campaign against the RSU can deny service to every vehicle within its coverage [94]. The same concern applies one level down in the clustered hierarchy, because a cluster head that is flooded, exhausted, or compromised can disrupt the communication of its entire cluster, which makes the reliability of the cluster head a security concern in its own right.

A further and often overlooked challenge is that different attack types are not independent. For example, a DDoS attack is characterized by a high traffic volume and a large number of connections, whereas an SQL injection attack manifests as anomalous activity directed at a database or web-facing service. Although these attacks differ at the surface level, they share deeper semantic links, such as their common effect on system availability and data integrity. Attacks may also occur in sequence, with reconnaissance or spoofing frequently preceding a flooding attack. Conventional detectors that engineer each attack in isolation cannot capture these correlations and therefore struggle to detect coordinated or previously unseen intrusions. This gap motivates a representation, namely a knowledge graph coupled with a GCN, that

explicitly encodes the semantic relationships among attacks and feeds them to a deep classifier.

Research on VANET security has produced a wide spectrum of intrusion detection techniques that can broadly be grouped into machine learning methods, deep learning (DL) methods, ensemble methods, and graph-based methods.

Within the ML family, hybrid strategies have been popular. Bangui et al. [97] combined Random Forest for detecting known breaches with a coreset and clustering stage that flags previously unseen intruders. Alsarhan et al. [94] proposed a four-stage detector comprising a rule-based filter, a Dempster-Shafer evidence combiner, a node history database, and a Bayesian learner that labels each event as well behaved or misbehaving. Turukmane and Devendiran [98] introduced Multi SVM, in which an opposition-based Northern Goshawk optimizer performs feature selection before a Mud Ring aided multilayer support vector machine classifies the traffic, followed by hyperparameter tuning. Position-falsification misbehaviour has also been addressed with KNN and Random Forest classifiers [99].

Deep learning approaches remove much of the manual feature engineering by learning representations directly from raw traffic. Vitalkar et al. [95] applied a Deep Belief Network (DBN) to VANET intrusion detection and reported accuracy gains over classical ML, and related work enhanced DBN with rule-based detection for microgrid protection [100]. Kumar et al. [101] proposed the DLTIF framework, which pairs a deep feature extractor with cyber-threat-intelligence-driven detection and attack-type identification for IoT-enabled maritime systems. Blockchain and DL were combined in BDEdge [102] for connected and autonomous vehicles, using a sparse autoencoder enabled attention bidirectional gated recurrent unit over the link-load characteristics of an MEC-enabled RSU. Sequence reconstruction and thresholding with a deep neural network were used for anomaly detection on RSUs [103], while a generative adversarial network enabled distributed, network-wide IDS training across software-defined networking controllers without exchanging raw sub-network flows [104]. A stacked contractive autoencoder combined with a support vector machine merged deep and shallow learning to reduce analytical overhead [105], and a lightweight CNN identified nine categories of IoT cyber-attacks [106].

Ensemble learning has repeatedly been shown to outperform single classifiers. Dong et al. [107] built a bagging ensemble of upgraded C4.5 decision trees selected through the information gain ratio and combined by majority voting. Gad et al. [67] used an XGBoost ensemble with chi-square feature selection and synthetic minority oversampling on the ToN-IoT dataset, and El-Dalahmeh and Adeel [66] combined a DBN with a decision tree for software-defined VANETs. Jiang et al. [23] paired the Whale Optimization Algorithm with XGBoost for security in the Internet of Vehicles, and Raja et al. [108] proposed the SP-CIDS collaborative detector that blends logistic regression, Naive Bayes, and ensemble classifiers with differential privacy. Access-control-oriented designs have also appeared, such as the attribute-based encryption scheme with policy hiding, which uses a BiLSTM and a sparse stacked autoencoder to filter fraudulent data-access requests [73], and a hybrid semantic deep learning model that uses a Word2Vec layer with SVM, LSTM, and CNN components [69].

Finally, graph-based detectors have gained traction because they can represent intricate relationships in network traffic. Park et al. [71] and Song et al. [74] applied graph structures to CAN messages and reported resilience to variations in attack type, while a big-data pipeline using Spark and HDFS was used for distributed DDoS detection [109]. Graph neural network detectors such as E-GraphSAGE [110] and BS-GAT [111] have further advanced the field for IoT and edge computing. Table 5.1 summarizes representative approaches according to whether they exploit semantic relations between attacks and whether they employ ensemble learning.

Table 5.1 Overview of related work in VANET and related intrusion detection

Reference	Method used	Semantic relations used	Ensemble used
El-Dalahmeh M et al. [66]	Ensemble of DBN and decision tree	No	Yes
Gad et al. [67]	XGBoost	No	Yes
Prabhakaran V et al. [69]	HSDL with CNN, SVM, and LSTM	Yes	Yes
Park SB et al. [71]	Graph-based model (CAN)	Yes	No
Wang Z [73]	SSAE and three-layer BiLSTM	No	Yes
Bangui et al. [97]	Random Forest with clustering	No	No

Reference	Method used	Semantic relations used	Ensemble used
A. V. Turukmane et al. [98]	Northern Goshawk optimization with Multi-SVM	No	Yes
S. Ercan et al. [99]	kNN and Random Forest	No	Yes
P. Kuma et al. [102]	Hybrid SAE-ABIGRU (BDEdge)	No	No
T. Alladi et al. [103]	Deep Neural Network (DNN)	No	No
W. Wang et al. [105]	Stacked contractive autoencoder with SVM	No	No
K. H. Le et al. [106]	Lightweight CNN	No	No
R. H. Dong et al. [107]	Bagging ensemble of C4.5 decision trees	No	Yes
G. Raja et al. [108]	LR, Naive Bayes, ensemble with privacy	No	Yes
Y. Gao et al. [109]	Random Forest	No	No
G. Kaur et al. [112]	Hybrid optimization-based Deep Maxout Network	No	No
W. Jiang et al. [113]	Whale Optimization with XGBoost	No	Yes
M. K. Pulligilla et al. [114]	RideNN trained with RBSLO	No	No
Proposed	KG-DeepCNN (GCN plus tunnel ensemble)	Yes (GCN module)	Yes (tunnels)

The reviewed literature exposes three recurring limitations. First, a large share of the schemes concentrates on countermeasures such as encryption and policy control [69], [73], [108], [113], [114]. These are ill-suited to highly mobile vehicles and, even when they focus on detecting malicious content, they generally ignore the semantic relations among features. As noted earlier, treating DDoS and SQL injection as independent phenomena discards the shared structure that a detector could otherwise exploit.

Second, only a few approaches address semantic relations explicitly. The hybrid semantic deep learning model in [69] uses a Word2Vec embedding layer, and the graph-based CAN detectors in [71], [74] encode interactions between messages. However, embedding methods such as Word2Vec learn representations from co-occurrence statistics and largely ignore the features of neighbouring nodes, and the

available graph-based detectors are confined to CAN traffic rather than the broader VANET setting.

Third, although ensembles consistently outperform single classifiers [107], [113], most existing ensembles are homogeneous, combining models of the same architecture. A single learner can become trapped in local minima or constrained by its search space, yet simply repeating one architecture does not fully exploit the complementary views that different receptive fields and depths can provide. These gaps motivate a model that captures semantic relations among attacks through a knowledge graph and GCN, and that classifies the resulting embeddings with a heterogeneous ensemble of CNN learners, which is precisely the design pursued in the remainder of this chapter.

5.2 Proposed KG-DeepCNN Framework

The proposed framework, KG-DeepCNN, is an IDS-aware access-control scheme that operates at the RSU. Its purpose is to evaluate every incoming data-access request and to permit only legitimate traffic, thereby strengthening V2V communication security. It is organized in two stages, namely a knowledge graph coupled with a graph convolutional network that produces semantically enriched embeddings, and a heterogeneous ensemble of CNN learners that classifies those embeddings. The following subsections describe the overall architecture, the feature representation, and the detection mechanism in turn.

5.2.1 Framework Architecture

The system model comprises three interacting components, namely vehicles, RSUs, and a central server, as illustrated in Figure 5.1.

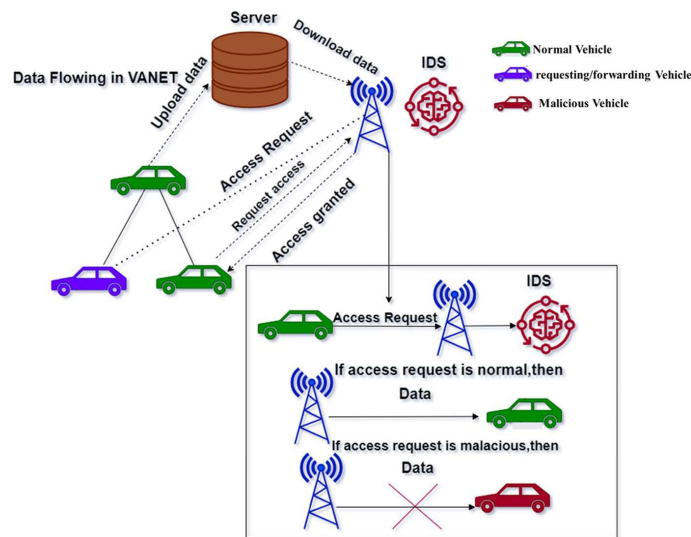


Figure 5.1 The proposed VANET system model, in which the RSU applies the IDS to grant or deny access requests

Vehicles act simultaneously as producers and consumers of data. They upload information to the central server for sharing and request access to data or exchange data with neighbouring vehicles through the V2V protocol. Each access request is submitted to the RSU, which behaves as a semi-trusted entity that monitors communications, manages outsourced tasks within its range, and inspects packets in its transmission range. The RSU also mediates between the vehicles and the server, reducing the computational and storage load on individual vehicles. The server is likewise treated as a semi-trusted entity with ample processing and storage capacity. In the deployed scenario the vehicles are partitioned into clusters, and within each cluster an elected cluster head aggregates the data-access requests of its members and forwards them to the RSU, so that the access-control decision made by the IDS at the RSU effectively governs the traffic that each cluster head is permitted to relay to and from its members.

The detection logic embedded in the RSU is summarized in Algorithm 5.1. After random vehicles are initialized and the RSU is deployed, a vehicle transmits a data-request packet that carries its identifier, header information, and the specific data being sought. The RSU forwards the packet features to the trained KG-DeepCNN model, which returns a decision score. If the request is judged legitimate according to the patterns learned from the training datasets, the requested data is delivered; otherwise,

the request is denied. In this way the RSU continuously screens every incoming packet, admitting legitimate requests and blocking malicious ones so that the integrity of data communication in the VANET is maintained.

Algorithm 5.1 Overall operation of the RSU-based IDS**Input: datasets, model parameters, vehicle data-access request packet T**

1. Initialize random vehicles and deploy the RSU
2. Initiate the vehicle data request
3. Begin
4. RSU receives T (packet information) from the vehicles
5. $P_final = KG-DeepCNN(T)$
6. If P_final is legitimate (normal as per the training datasets) then
 - i. RSU sends the requested data to the vehicle
 - ii. Else
 - iii. Deny the request
7. End if
8. End

Output: access to the requested data if all checks pass**5.2.2 Knowledge Graph-Based Feature Representation**

The first stage of KG-DeepCNN converts preprocessed traffic into a semantically enriched representation. Training and testing use four benchmark datasets, namely NSL-KDD [115], UNSW-NB15 [116], ToN-IoT [117], and BoT-IoT [118], each partitioned into normal and malicious traffic. Preprocessing removes null values and normalizes the raw features. For BoT-IoT, non-informative attributes such as source and destination addresses, protocol, packet sequence identifier, flags, and state are dropped, and categorical attack labels are encoded into numerical values before normalization. The overall architecture, comprising the knowledge graph, the GCN module, and the ensemble of CNN tunnels, is depicted in Figure 5.2.

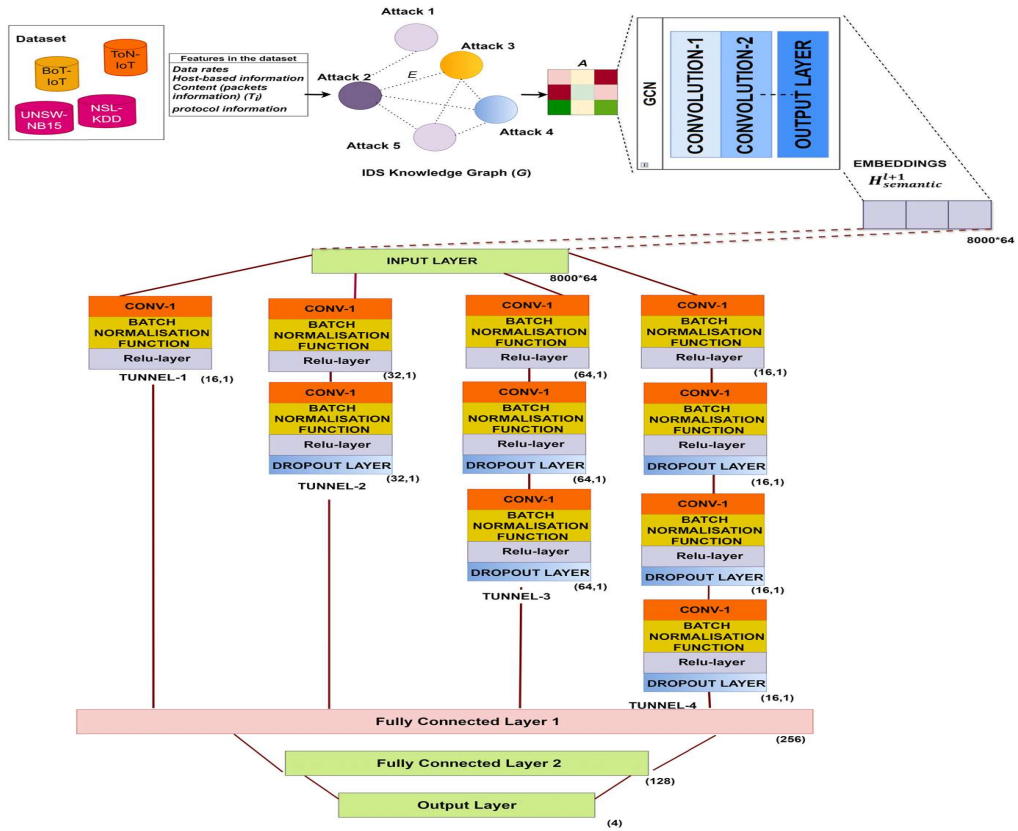


Figure 5.2 The proposed KG-DeepCNN model

Let T denote the collection of real-time traffic received by the RSU. Each instance corresponds to a packet from a vehicle and is expressed as a set of header features, as given in Equation (5.1).

$$T_i = \{h_{ij} \mid j \in [1, H]\} \quad (5.1)$$

where H is the total number of header features and h_{ij} is the j -th feature of packet T_i . The extracted feature values and their interactions are represented as the nodes and edges of a traffic attribute graph, so that features belonging to the same packet are linked and packets that share identical nodes reveal commonalities across the hidden structure of the graph.

Knowledge graph construction

The knowledge graph is defined as $G = (V, E)$, where the nodes V represent the labels of the attacks and the edges E capture the relations among packets. In the BoT-IoT case, for example, the nodes correspond to the four attack categories used in the study. Nodes are enriched with attributes such as packet size, transmission rate, and protocol

type, while edges encode semantic relations that include similarity (two attacks that both cause denial of service), causality (abnormal packet frequency leading to congestion), and co-occurrence (spoofing that often precedes packet flooding). The structural information of G is stored as an adjacency matrix A , whose entries are derived from the cosine similarity between feature vectors, as expressed in Equation (5.2).

$$A = \{a_{ij} \text{ if } a_{ij} \geq r, \text{ else } 0\} \quad (5.2)$$

Where, $a_{ij} = \frac{v_1 \cdot v_2}{||v_1|| ||v_2||}$ is cosine similarity between the feature vectors of two intrusion instances, computed over quantities such as packet flow and rate, and τ is a similarity threshold. An edge is retained only when the similarity between two nodes meets or exceeds τ , so the threshold directly controls the sparsity and density of the graph.

To study the sensitivity of the model to this parameter, the adjacency matrix was constructed for thresholds of 0.3, 0.5, 0.7, and 0.9. As shown in Figure 5.3, a low threshold retains many edges and yields a dense graph that preserves potential relationships but also admits noisy or weakly correlated connections, which reduces accuracy. A value of 0.5 offers the best balance, since connectivity is improved, semantically meaningful relations are preserved, and the GCN layers can propagate structural information effectively. At this setting the model reaches its highest accuracy on every dataset, namely 98.7% on ToN-IoT, 98% on UNSW-NB15, 99% on BoT-IoT, and 99% on NSL-KDD. As the threshold rises further, the graph becomes too sparse and useful associations are discarded, which weakens the relational structure and gradually lowers accuracy. The threshold was therefore fixed at 0.5 for all subsequent experiments.

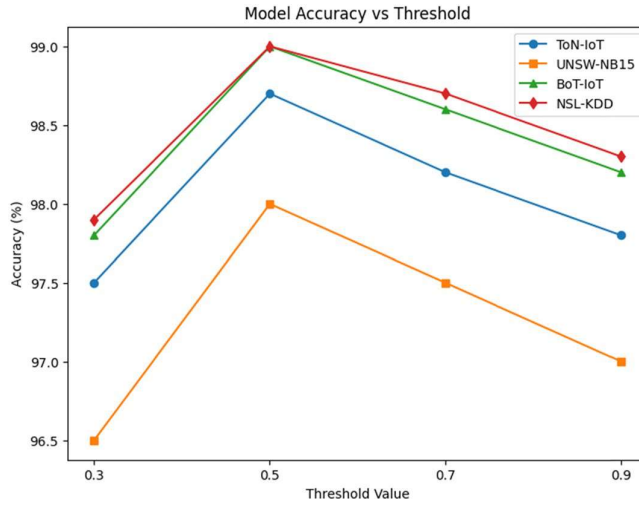


Figure 5.3 Impact of the cosine-similarity threshold on classification accuracy across the four datasets

Semantic feature generation with the GCN

After normalization of the relations, a graph convolutional network extracts local structural features. The layer-wise propagation rule is given in Equation (5.3).

$$H_{semantic}^{l+1} = \sigma \left(D^{-\frac{1}{2}} \tilde{A} D^{-\frac{1}{2}} H_{semantic}^l W^{(l)} \right) \quad (5.3)$$

Here, $\tilde{A} \triangleq A + I_N$ is the adjacency information of G . I_N is the identity information, $D_{ii}^{-\frac{1}{2}} = \sum_j A_{ij}^{-\frac{1}{2}}$ and $W^{(l)}$ weight matrix for layers l . $\sigma(\cdot)$ denotes an activation function, such as $ReLU(\cdot) = \max(0, \cdot)$. $H^{(l)} \in R^{N \times D}$ is the matrix for the activation function in l^{th} layer, $H^{(0)} = X$. However, as noticed, the process in Z^{GCN} the update of structural information A over convolution layers is limited to immediate neighbour attacks. Thus, the implicit intrusion relations among the attacks are captured through the component Z^{MHGAT} in Equation 5.1 The relations are normalized following the procedure in [119, 120].

Because a plain GCN aggregates information only from immediate neighbours, implicit relations that span more distant attacks are captured by an additional attention-based component in the embedding pipeline. The resulting embeddings are trained with the cross-entropy objective in Equation (5.4).

$$L_{gra} = - \sum_{u \in V} \hat{z}_u * \log(x_u) \quad (5.4)$$

where x_u carries the ground-truth label indicating whether instance u is benign or an attack, and $\hat{z}_u$ is the estimated label produced by the combined model. A distinctive property of the learned embeddings is orthogonality. Figure 5.4 contrasts the clusters obtained from the CNN representation with those obtained from the GCN embeddings on the ToN-IoT dataset. The GCN both extracts the unique relations of the traffic and encourages independence among the learned features, which produces dense, well-separated clusters, whereas the CNN clusters overlap. This separation allows the downstream ensemble to distinguish network activities that would otherwise share nearly identical raw representations.

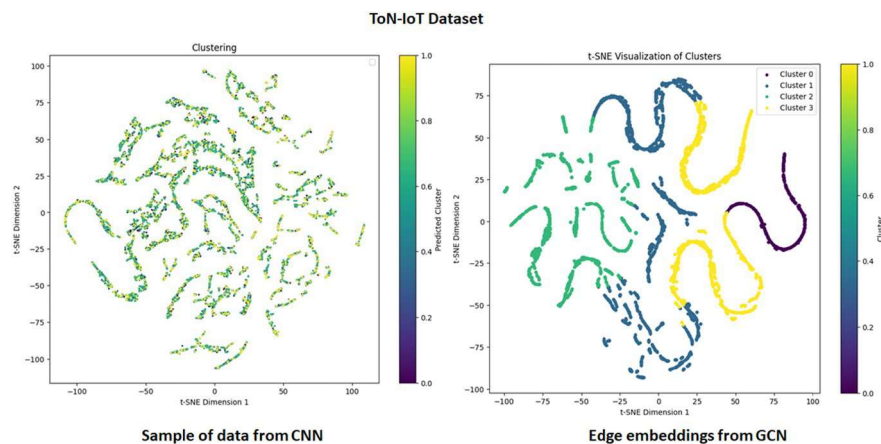


Figure 5.4 The overlapping clusters from the CNN (left) contrast with the well-separated, orthogonal clusters from the GCN embeddings (right)

5.2.3 DeepCNN-Based Attack Detection Mechanism

The second stage classifies the semantic embeddings using a weighted ensemble of CNN learners, organized as four interconnecting tunnels of increasing depth. In contrast to homogeneous ensembles, each tunnel is configured differently so that it contributes a complementary view of the data. The ensemble can be expressed compactly as the sum of the four tunnels, where Tunnel-1 contains a single CNN unit, Tunnel-2 contains two, Tunnel-3 contains three, and Tunnel-4 contains four.

Tunnel-1 uses small kernels to detect localized anomalies in packet attributes, Tunnel-2 applies larger kernels to capture long-range dependencies across communication sequences, Tunnel-3 stacks deeper layers to model higher-level semantic correlations, and Tunnel-4 relies on dropout regularization and residual connections to counter

overfitting. By blending shallow and deep architectures with varied receptive fields, the ensemble exploits both fine-grained and global semantic features, which improves generalization relative to any single CNN or a homogeneous ensemble.

Each tunnel receives node embeddings of size 8000 by 64, corresponding to the number of training samples multiplied by the hidden dimension. The number of samples differs by dataset, with 10,000 records used for BoT-IoT, UNSW-NB15, and NSL-KDD, and 15,000 for ToN-IoT, of which 80% form the training set. Within a tunnel, a Conv2d unit is followed by ReLU activation and BatchNorm1d. Tunnel-1 contains a single such unit at hidden size 16; Tunnel-2 adds a second unit at hidden size 32 that also includes dropout; Tunnel-3 extends this to three units at hidden size 64; and Tunnel-4 uses four units at hidden size 16, each combining Conv2d, BatchNorm1d, ReLU, and dropout. The outputs of the four tunnels are concatenated and passed through fully connected layers, with the first two layers set to sizes 256 and 128 and a final output layer whose width equals the number of attack classes in the dataset. The convolutional configuration common to all tunnels is listed in Table 5.2.

Table 5.2 Parameter setting for the CNN used in tunnels 1 to 4

Parameter	Value
Layers	Conv2d, BatchNorm, ReLU
Kernel size	3 x 3
Padding	1
Activation function	ReLU
Epochs	50
Reshape inputs at the ensemble	(-1, 2, 1)
Optimizer at the ensemble	Adam
Batch size at the ensemble	64

The output of a convolutional unit is computed as shown in Equation (5.5).

$$x_i^{l,j} = g \left(\sum_{a=1}^m w_a^j \cdot H_{semantic+a}^l + b_j \right) \quad (5.5)$$

matrix, $x_{i(l,j)}$ is the outcome with the j^{th} feature map based on the i^{th} unit of the l^{th} convolutional layer, and $b(j)$ is the bias for the j^{th} convolutional feature map

where $w_{a(j)}$ is the convolutional kernel, $x_{i(l,j)}$ is the j^{th} feature map at the i^{th} unit of layer l , $b(j)$ is the corresponding bias, and $g(\dots)$ is the activation. Using ReLU, a representative first-layer computation is written as in Equation (5.6).

$$x_1^{1,1} = \text{relu}(w_1^1 x_1^{0,1} + w_2^1 x_2^{0,1} + w_3^1 x_3^{0,1} + b_1) \quad (5.6)$$

To reduce the dimensionality of the extracted features while preserving their internal relationships, a max-pooling operation is applied, as expressed in Equation (5.7).

$$x_i^{l,j} = \max_{i,j=1}^r (x_{i,j}) \quad (5.7)$$

The complete training procedure is given in Algorithm 5.2. After preprocessing and adjacency-matrix construction, the GCN embeddings are produced by forward propagation through the graph layers. The DeepCNN ensemble is then trained with five-fold cross-validation, in which the training data is split into five equal subsets, each classifier is trained on its subset, and the predictions of earlier classifiers are used to build the training data for the next one. A set of weights scales the contribution of each tunnel, and the weighted outputs are accumulated into a total score whose argmax yields the final decision. Optimization uses the Adam optimizer and the cross-entropy loss.

Algorithm 5.2 Training procedure of the proposed KG-DeepCNN

Input: datasets, T_i , model parameters

1. Initialize the data preprocessing steps
2. Calculate A (adjacency matrix). # extracting the semantic relations among the attacks.
3. Produce the embeddings $H_{semantic}^l$ by carrying out forward propagation over GCN layers in accordance with Equation 3 (a numerical representation of the GCN equation).
4. Initialize the DeepCNN model
5. Initialize $P_{total} = 0$ # Start with the total prediction score
6. For each classifier j in $\{1, 2, 3, 4\}$:
 - i. Train Classifier C_j with Cross-Validation
7. For each fold m in $\{1, 2, \dots, M\}$: #Where M is the number of cross-validation folds
 - i. Train a classifier C_j on the training subset D_m .
 - ii. Generate the dataset for training the next classifier C_{j+1} using predictions from classifiers C_1 to C_j .
8. End For
9. Compute the weighted (w) output for the classifier C_j : $S_j = \Sigma f(D_{test})$
#Sum the predictions for test node embeddings D_{test} .
10. Update P_{total} by adding the weighted output: $P_{total} += w_j * S_j$
11. End For
12. Determine the final decision based on the cumulative score:

$$P_{final} = \text{argmax}(P_{total}).$$

Output: Classified attack P_{final} .

For completeness, the computational cost of the framework can be bounded as follows. An L-layer GCN has time complexity $O(LmF + LnF^2)$, where the first term reflects sparse-dense multiplication and the second reflects feature transformation, with n nodes, m edges, and feature dimension F . A single-layer CNN has complexity $O(N)$ for its weights and $O(NMPQR * S)$ for the convolution, where the symbols denote the input and filter dimensions and the strides; the dropout in the deeper tunnels adds only negligible overhead. The final fully connected stage, formed from the concatenated tunnel outputs, has complexity $O(1024)$.

5.3 Experimental Setup

5.3.1 Dataset and Simulation Environment

All experiments were conducted in the Google Colab Pro environment using an NVIDIA Tesla T4 GPU with 16 GB of memory, which provided accelerated computation for both the deep learning models and the network simulation. The VANET scenario was modelled on the Connaught area of Delhi, a busy urban region that offers a dense and complex setting for evaluating the RSU-based IDS. Within this scenario the RSU continuously monitors vehicle behaviour using gathered attributes that include the vehicle identifier, lane identifier, speed, location, and packet information. A total of 1000 vehicles transmit 200-byte packets, the vehicles are grouped into four clusters, each coordinated by a cluster head that aggregates and relays member requests, over a simulation time of 1500 seconds, and the RSU analyses the resulting traffic to identify the attack types associated with each dataset. The simulation settings are listed in Table 5.3, and the learning hyperparameters for the GCN and the DeepCNN are listed in Table 5.4.

Table 5.3 Simulation settings for the VANET scenario

Parameter	Value
Traffic area	Connaught Area, Delhi
Vehicle transmission range	200
Vehicle density	1000
Vehicle attributes collected	ID, lane ID, speed, location, packet information
Packet size	200 bytes
Simulation time	1500 seconds

Parameter	Value
Modulation scheme	BPSK
Attack types	DoS, DDoS, injection, scanning, backdoor (ToN-IoT); keylogging, service scan, OS attacks, data exfiltration (BoT-IoT); probe, R2L, U2R (NSL-KDD); exploits, worms, generic, shellcode (UNSW-NB15)
Data rate	14 Mbps
Number of clusters	4

Table 5.4 Learning hyperparameters for the GCN and the DeepCNN

Parameter	Value
Number of epochs for GCN and DeepCNN	50
Number of hidden layers in GCN	2
Learning rate in GCN	0.01
Hidden dimension in GCN	64
DeepCNN batch size	64
DeepCNN learning rate	0.001

Four benchmark datasets were used, chosen because they represent contemporary and increasingly relevant attack vectors for smart transportation. ToN-IoT, developed by the UNSW Canberra IoT Labs, contains telemetry, system logs, and network traffic with 44 attributes per record and covers attacks such as ransomware, DoS, DDoS, scanning, and man-in-the-middle. Its telemetry from connected devices can emulate real-time vehicle communication, which makes it well suited to training the RSU module to detect malicious requests. BoT-IoT, generated in the UNSW Canberra Cyber Range Lab, captures normal and botnet traffic and includes DDoS, DoS, keylogging, service-scan, and data-exfiltration attacks. It comprises more than 72 million records and is widely used for IoT network IDS research, and its attack mix corresponds well to scenarios that could disrupt V2V communication.

NSL-KDD is a refined version of the KDD Cup 1999 dataset and remains a standard reference for comparing intrusion detection techniques, providing a structured basis for assessing how well the RSU distinguishes legitimate requests from malicious ones. UNSW-NB15 contains nine categories of network attacks, including DoS, worms, and

backdoors, with 175,341 training and 82,332 testing records, and is used here to verify that the detector generalizes to a broader network dataset that is not tailored exclusively to IoT or VANET traffic.

5.3.2 Performance Evaluation Metrics

The detector was assessed with the standard classification metrics of accuracy, precision, recall, and F1-score, together with the training loss. Accuracy measures the proportion of correctly classified instances, as defined in Equation (5.8).

$$\text{Accuracy} = \frac{TP+TN}{FP+FN+TN+TP}. \quad (5.8)$$

Precision, defined in Equation (5.9), is the proportion of correctly predicted positives among all predicted positives, and is particularly important in VANETs because it reflects how rarely a legitimate request is wrongly flagged as malicious.

$$\text{Precision} = \frac{TP}{FP+TP} \quad (5.9)$$

$$\text{Recall} = \frac{TP}{TP+F} \quad (5.10)$$

Recall, defined in Equation (5.10), is the proportion of actual positives that are correctly identified, and it captures the ability of the detector to avoid missing genuine attacks.

The F1-score, defined in Equation (5.11), is the harmonic mean of precision and recall and provides a single balanced measure of performance. In these expressions TP, TN, FP, and FN denote true positives, true negatives, false positives, and false negatives, respectively.

$$\text{F1-score} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \quad (5.11)$$

5.4 Results and Discussion

5.4.1 Intrusion Detection Performance

The proposed model was evaluated on a tunnel-wise basis and compared against the full ensemble, denoted P_{final} , across the four datasets.

BoT-IoT

The classification metrics on the BoT-IoT dataset reports Figure 5.5. The ensemble attains 99% across accuracy, precision, recall, and F1-score, confirming that it identifies both normal and malicious traffic effectively. The advantage of aggregation is clear when the ensemble is compared with the weaker tunnels, since it improves on Tunnel-2 by 24%, 12%, 14%, and 13% in accuracy, precision, recall, and F1-score, respectively. In terms of recall it exceeds Tunnels 2, 3, and 4 by 14%, 33%, and 31%, and in terms of F1-score by 25%, 14%, and 34%. Its accuracy advantage over Tunnels 3 and 4 reaches 19% and 33%. Notably, the simple Tunnel-1 also achieves 99% accuracy on this dataset, because the GCN embeddings already encode spatial and relational structure and the compact pipeline of Tunnel-1 refines those features efficiently.

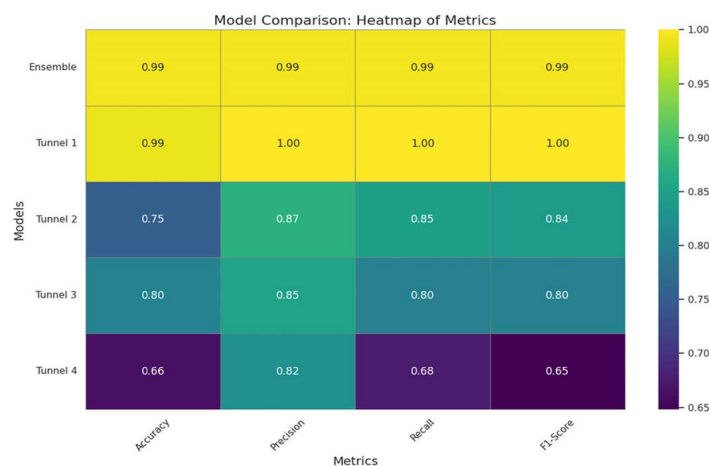


Figure 5.5 Classification metrics of the proposed ensemble and tunnels 1 to 4 on the BoT-IoT dataset

ToN-IoT

The ensemble achieves consistently high scores of 98% across all metrics, indicating that the GCN embeddings capture meaningful relationships among attacks and their features as shown in Figure 5.6. The accuracy improvement over the individual tunnels is 1.02%, 15.3%, 6.12%, and 25.5% for Tunnels 1 to 4, respectively, and precision improves by 18%, 5%, and 11% over Tunnels 2, 3, and 4. Tunnel-1 exhibits very high precision, recall, and F1-score but a slightly lower accuracy, which reflects a higher false-positive rate, since it occasionally misclassifies normal behaviour as an intrusion, leading to a 1.02% accuracy shortfall relative to the ensemble.

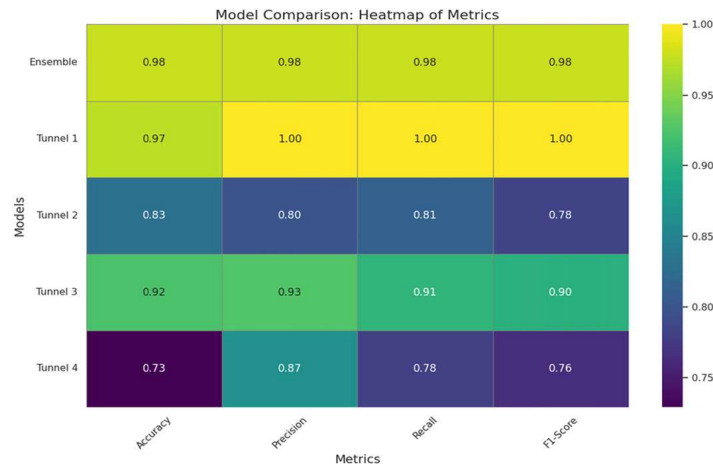


Figure 5.6 Classification metrics of the proposed ensemble and tunnels 1 to 4 on the ToN-IoT dataset

NSL-KDD

Figure 5.7 presents the results on NSL-KDD, where the ensemble reaches 99% across the metrics. Its accuracy improves on Tunnels 2, 3, and 4 by 12.12%, 4%, and 11%, precision improves by 15%, 8%, and 8%, and recall improves by 21%, 8.1%, and 22%. These gains indicate that combining the tunnels allows the RSU to evaluate the legitimacy of access requests with high reliability. Individual deeper tunnels, especially Tunnel-4, perform less well because, as depth increases, a single tunnel is more prone to overfitting and loss of generalization, whereas the ensemble mitigates this through the concatenation of complementary outputs. Once again Tunnel-1 attains 99% accuracy owing to the expressiveness of the GCN embeddings.

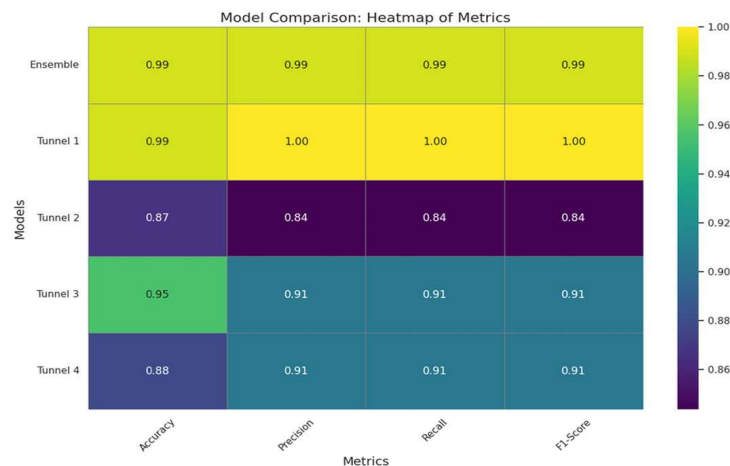


Figure 5.7 Classification metrics of the proposed ensemble and tunnels 1 to 4 on the NSL-KDD dataset.

UNSW-NB15

The results on UNSW-NB15 are shown in Figure 5.8. The ensemble records 98% accuracy, exceeding Tunnels 1 to 4 by 1.05%, 15%, 6.12%, and 25.5%, respectively. Precision improves by 18.3%, 5%, and 11.22% and recall by 17%, 7.14%, and 20.4% over Tunnels 2, 3, and 4. The high precision is significant for VANET security because it minimizes false positives, ensuring that legitimate data exchanges are not needlessly interrupted, while the high recall reduces the chance of missing a genuine threat that could otherwise lead to security breaches or safety hazards.

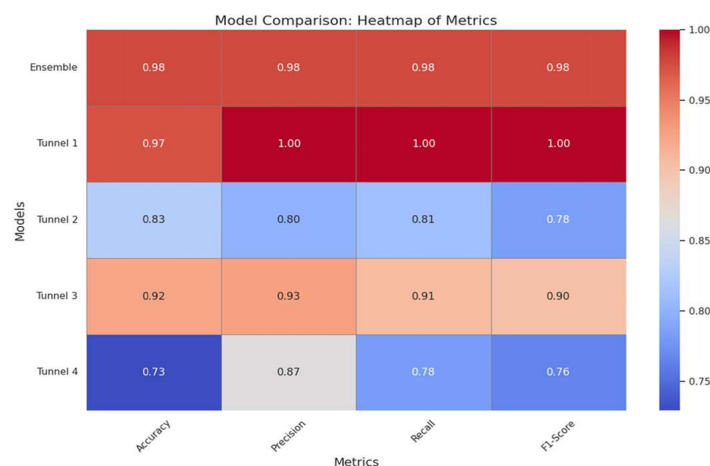


Figure 5.8 Classification metrics of the proposed ensemble and tunnels 1 to 4 on the UNSW-NB15 dataset

Model Loss

The training loss over 50 epochs for all four datasets shown in Figure 5.9. The ensemble and Tunnel-1 consistently maintain the lowest loss, with the ensemble converging to a low value of about 0.19 across datasets. Tunnels 3 and 4 stabilize more slowly and settle at higher loss, with Tunnel-4 reaching roughly 0.225 on UNSW-NB15, 0.22 on NSL-KDD, 0.29 on ToN-IoT, and 0.2 on BoT-IoT, which is symptomatic of overfitting and instability in the deeper single-tunnel configurations. The ability of the ensemble to combine the strengths of all tunnels produces a balanced and robust model with low loss, which in turn supports fast and reliable decisions at the RSU.

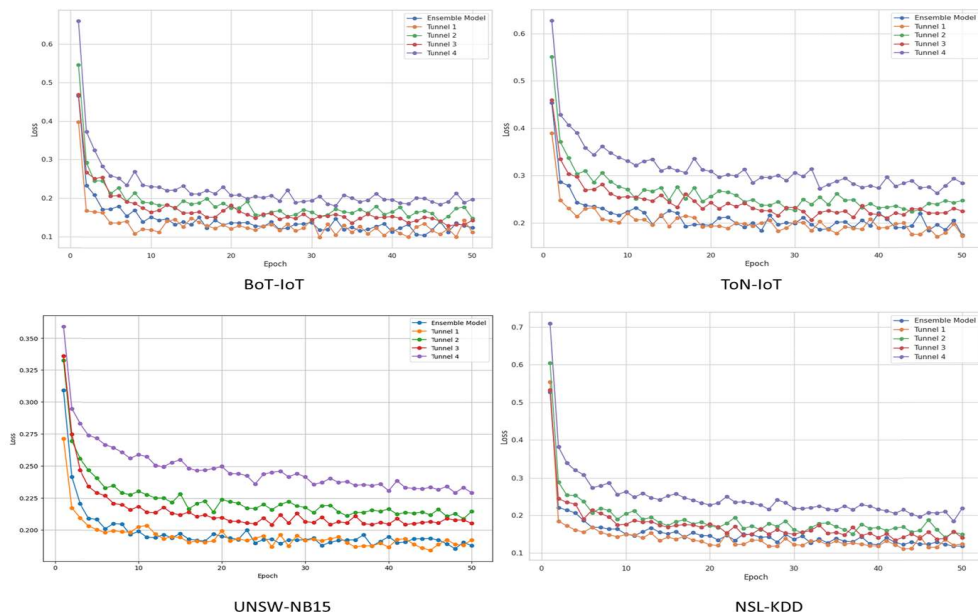


Figure 5.9 Training loss curves for the ensemble and tunnels 1 to 4 on the BoT-IoT, ToN-IoT, UNSW-NB15, and NSL-KDD datasets

Per-class behaviour

The confusion matrices in Figures 5.10 to 5.13 provide a finer view of performance across individual attack classes. On NSL-KDD (Figure 5.10), 623 of 670 normal samples are classified correctly, with only 47 errors, all DoS instances are detected without cross-class confusion, 2925 of 2939 probe or reconnaissance samples are identified, and 2354 of 2366 user-to-root or remote-to-local samples are recognized, leaving only 12 errors on these rarer but more sophisticated attacks.

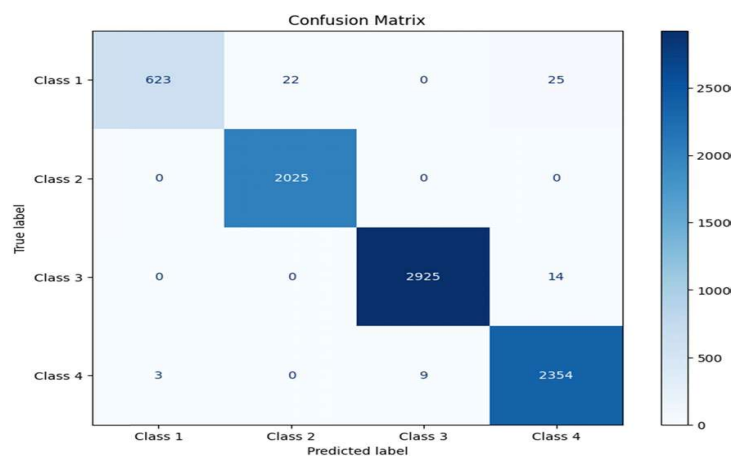


Figure 5.10 Confusion matrix for the NSL-KDD dataset

On BoT-IoT (Figure 5.11), normal traffic is classified almost perfectly, with 3202 of 3253 samples correct, 2630 of 2676 DDoS samples are detected, all 1378 data-exfiltration or keylogging samples are identified without error, and 1714 of 1720 password or brute-force samples are recognized with only 6 mistakes. The flawless detection of stealthy exfiltration and the strong performance on brute-force attacks are important because compromised nodes can otherwise be used to launch secondary attacks.

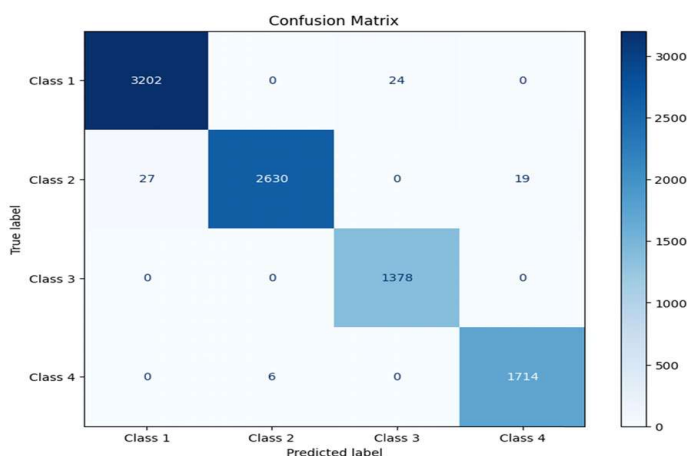


Figure 5.11 Confusion matrix for the BoT-IoT dataset

On ToN-IoT (Figure 5.12), 5790 of 5803 normal samples are correctly classified with only 13 false positives, 3998 of 4118 DoS or DDoS samples are detected, and 1556 of 1575 reconnaissance or scanning samples are identified with only 19 false positives. The most challenging class is the group of passwords, man-in-the-middle, and other complex attacks, of which 417 of 504 are correctly classified while 87 are confused with normal traffic, reflecting the difficulty of separating stealthy, low-volume intrusions from benign activity.

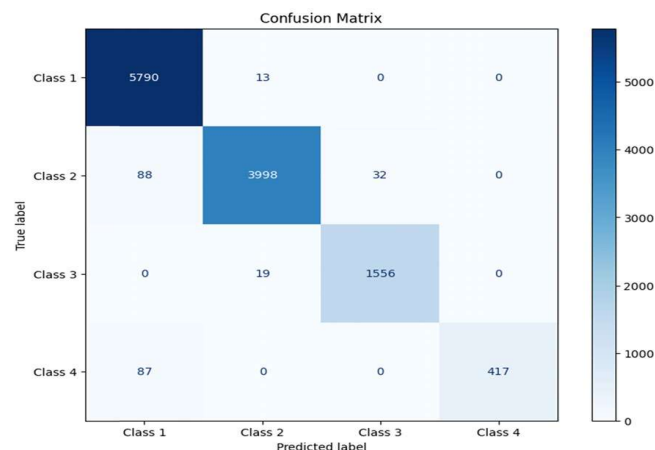


Figure 5.12 Confusion matrix for the ToN-IoT dataset

On UNSW-NB15 (Figure 5.13), 63 of 65 normal samples are classified correctly with only two errors, and all 202 attack samples are detected with zero false negatives, which is especially valuable because a single undetected intrusion could compromise VANET nodes and propagate malicious traffic.

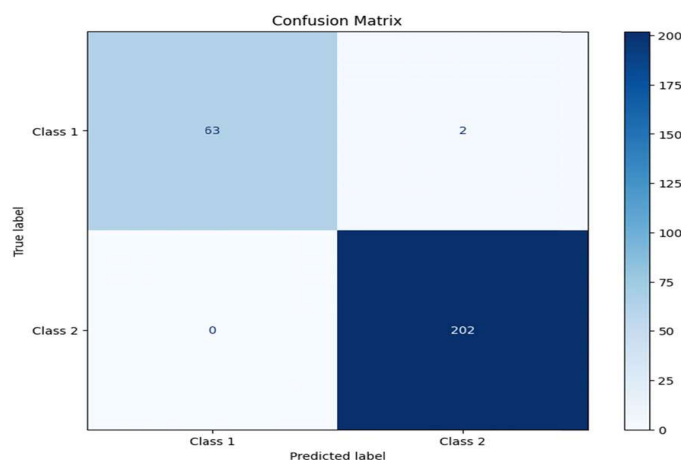


Figure 5.13 Confusion matrix for the UNSW-NB15 dataset

Taken together, these results confirm that NSL-KDD demonstrates robustness across a wide range of attack categories, BoT-IoT validates resilience against modern IoT-driven threats such as large-scale DDoS and stealthy exfiltration, ToN-IoT shows that the detector handles diverse and realistic attack scenarios including reconnaissance and large-scale flooding, and UNSW-NB15 confirms that the model generalizes to a broader network dataset not tailored exclusively to IoT or VANET contexts.

5.4.2 Reliable Communication Performance

This section addresses the research objective of analyzing the reliability of the cluster head under the proposed model. In the clustered topology described in Section 5.2.1 and configured in Table 5.3, each cluster head aggregates the access requests of its members and relays them through the RSU-hosted IDS. The cluster head can therefore forward traffic reliably only to the extent that the IDS makes correct admit-or-deny decisions on the requests passing through it. Two failure modes threaten this reliability. A false positive, in which a legitimate request is wrongly labelled malicious, forces the cluster head to drop a valid packet and interrupts service for a well-behaved member. A false negative, in which an attack is admitted as normal, allows hostile traffic to enter the cluster and can exhaust or compromise the cluster head itself, degrading communication for every member. High precision therefore protects service continuity at the cluster head, while high recall protects the cluster head from being overwhelmed or subverted.

To make this analysis concrete, three reliability indicators are derived directly from the per-class counts in the confusion matrices of Figures 5.10 to 5.13, without any additional assumptions beyond the reported values. Legitimate service continuity is the fraction of genuine (normal) requests that the cluster head correctly admits, computed as the number of normal samples classified as normal divided by the total number of normal samples. Malicious containment is the fraction of attack requests that the cluster head does not admit as normal, computed as one minus the number of attack samples misclassified as normal divided by the total number of attack samples. The composite cluster head reliability is the overall proportion of correct gating decisions, computed as the sum of correctly admitted normal requests and correctly contained attack requests divided by the total number of requests. The resulting values are reported in Table 5.5 and visualized in Figure 5.14.

Table 5.5 Cluster head reliability indicators derived from the per-class confusion-matrix counts of Figures 5.10 to 5.13

Dataset	Legitimate service continuity (%)	Malicious containment (%)	Cluster head reliability (%)
NSL-KDD	92.99	99.96	99.38
BoT-IoT	98.43	99.53	99.14

Dataset	Legitimate service continuity (%)	Malicious containment (%)	Cluster head reliability (%)
ToN-IoT	99.78	97.18	98.43
UNSW-NB15	96.92	100.00	99.25

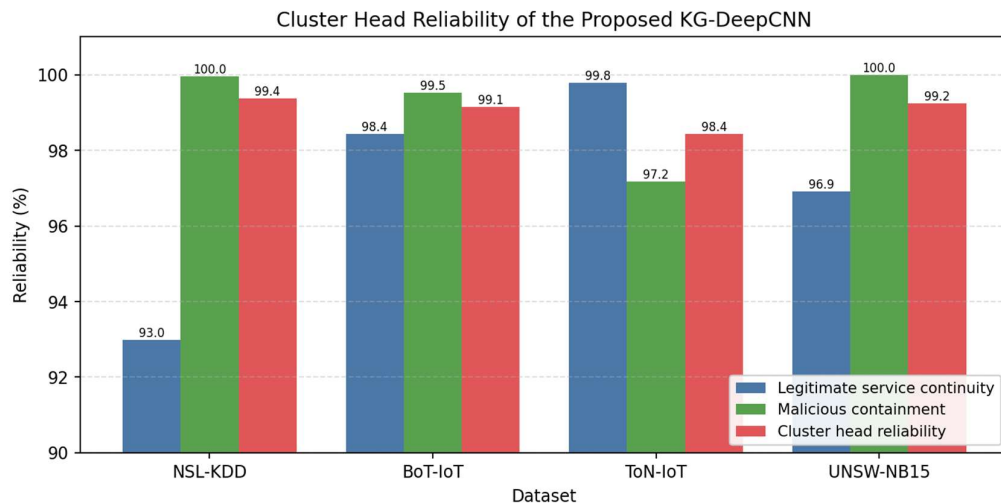


Figure 5.14 Cluster head reliability of the proposed KG-DeepCNN

The results show that the cluster head remains highly reliable under every dataset, with a composite reliability between 98.43% and 99.38%. Legitimate service continuity is strongest on ToN-IoT (99.78%) and BoT-IoT (98.43%), meaning that almost every honest member request is relayed without interruption, and it is lowest on NSL-KDD (92.99%), where 47 of 670 normal samples are wrongly flagged and would be dropped by the cluster head. Malicious containment is essentially perfect on UNSW-NB15 (100.00%, with no attack admitted as normal) and NSL-KDD (99.96%, with only three attacks leaking through), and it is lowest on ToN-IoT (97.18%), where 175 of 6197 attack samples, mainly stealthy man-in-the-middle and low-volume complex attacks together with a portion of the DoS and DDoS class, are misread as benign. It is significant for availability that the DoS, DDoS, and reconnaissance classes are contained almost completely, since these are precisely the flooding and probing behaviours that would otherwise exhaust a cluster head, as discussed in Section 5.1.1. This reliability is reinforced by the loss behaviour in Figure 5.9. The consistently low loss achieved by the ensemble, converging to about 0.19, indicates that the detector reaches confident and stable decisions, which is essential because a cluster head must

admit or deny each relayed request without delay in a latency-sensitive environment. Taken together, the high legitimate service continuity keeps honest members connected through their cluster head, while the high malicious containment prevents the cluster head from relaying or being crippled by hostile traffic, so the accuracy of the proposed model translates directly into a reliable cluster head and, through it, into dependable cluster-wide communication.

5.5 Comparative Analysis

5.5.1 Comparison with Existing Methods

The proposed model was compared with a range of state-of-the-art methods on each dataset, including the graph neural network detectors GraphSAGE [37] and BS-GAT [38], optimization-driven ensembles, and classical ensemble and deep learning models. Table 5.6 summarizes the comparison in terms of overall accuracy achieved by KG-DeepCNN.

Table 5.6 State-of-the-art comparison of the proposed KG-DeepCNN across the four datasets

Dataset	Reference and method	Accuracy (%)	Increase (%)
ToN-IoT	Proposed model	98.7	-
	Graph Sage [110]	97.78	0.93
	BS-GAT [111]	97.90	0.81
	XGBoost [67]	97	1.72
	Deep Belief Network and Decision Tree [66]	97.4	1.31
	Whale optimization and XGBoost [69]	98.41	0.29
UNSW-NB15	Proposed model	98	-
	Graph Sage [110]	96.75	1.27
	BS-GAT [111]	92.16	5.84
	SSAE and BiLSTM [73]	97	1.01
	Ensemble of LSTM, CNN, SVM [69]	96	2.01
	Random Forest [109]	97	1.01
BoT-IoT	Proposed model	99	-
	Graph Sage [110]	96.5	2.52

Dataset	Reference and method	Accuracy (%)	Increase (%)
	BS-GAT [111]	98.35	0.65
	Whale optimization and XGBoost [113]	98.41	0.59
	CNN with Rider-based Lion optimization [114]	90	9.03
NSL-KDD	Proposed model	99	-
	Graph Sage [110]	97.95	1.06
	BS-GAT [111]	94.51	4.53
	Ensemble of Naive Bayes and Logistic Regression [108]	96	3
	Random Forest [109]	97	2.01

Table 5.6 presents a comparative evaluation of the proposed KG-DeepCNN framework against several state-of-the-art intrusion detection methods on the ToN-IoT, UNSW-NB15, BoT-IoT, and NSL-KDD datasets. On the ToN-IoT dataset, the proposed model achieves an accuracy of 98.7%, outperforming XGBoost, the Deep Belief Network with Decision Tree, GraphSAGE, BS-GAT, and Whale-Optimized XGBoost. For the UNSW-NB15 dataset, the model attains 98% accuracy, surpassing SSAE-BiLSTM, the LSTM-CNN-SVM ensemble, Random Forest, GraphSAGE, and BS-GAT. On the BoT-IoT dataset, the proposed framework records 99% accuracy, exceeding the CNN with Rider-Based Lion Optimization, GraphSAGE, BS-GAT, and Whale-Optimized XGBoost. Similarly, on the NSL-KDD dataset, it achieves 99% accuracy, outperforming the Naïve Bayes–Logistic Regression ensemble, Random Forest, GraphSAGE, and BS-GAT. These results demonstrate the effectiveness of the proposed KG-DeepCNN framework in accurately detecting network intrusions across diverse benchmark datasets, thereby supporting reliable and secure communication in VANETs.

Several observations explain these gains. Methods that rely on decision tree ensembles, such as XGBoost [67] and Random Forest [109], depend on surface-level features and do not integrate prior knowledge, whereas the tunnel architecture of KG-DeepCNN learns attack features more effectively. Optimization-driven schemes such as the Whale-optimized XGBoost [113] and the Rider-based approaches [114] suffer

from slow convergence in high-mobility settings and do not exploit prior structural knowledge, which is why the largest single gain, 9.03%, is recorded against the shallow optimization-heavy CNN in [114]. Compared with the SSAE and BiLSTM model [73], which emphasizes temporal learning, the CNN-based design captures spatial structure more effectively and yields a 2.01% improvement, and compared with the Word2Vec-based semantic model [69], the GCN learns embeddings by aggregating information from the neighbours of an attack rather than from co-occurrence statistics alone.

The comparison with graph neural network baselines is especially instructive. GraphSAGE [110] generates embeddings by sampling and aggregating neighbour information, which is computationally efficient but largely restricted to local statistics, limiting its ability to model long-range dependencies. BS-GAT [111] constructs edges from behaviour similarity and adds attention, which improves information extraction but relies on predefined similarity measures that are vulnerable to noisy or incomplete metrics and that tend to emphasize local patterns. KG-DeepCNN improves on both by combining knowledge graph construction with GCN-based semantic embeddings before classification, since it leverages structured attack-taxonomy relations to capture both local and global dependencies, builds embeddings from verified and preprocessed relations rather than similarity approximations, and produces orthogonal, well-separated clusters even when the raw data overlaps. These properties yield consistent accuracy gains of up to 5.95% over the strongest graph baseline.

5.5.2 Impact on Communication Reliability

The percentage improvements in Table 5.6, although sometimes modest in absolute terms, are consequential in the VANET context because each fractional gain directly reduces the number of misclassified requests at the RSU. Fewer misclassifications mean fewer false negatives, so fewer attacks slip through to disrupt the network, and fewer false positives, so fewer legitimate packets are dropped without cause. Both effects are central to maintaining real-time communication reliability. This is why even a numerically small improvement, such as the 1.01% gain over the SSAE and BiLSTM baseline on UNSW-NB15, is meaningful, since it lowers the likelihood of undetected

intrusions that exploit structural traffic patterns and translates into fewer false negatives in a safety-sensitive environment.

These gains are especially important at the cluster head, which is the busiest relay point in the topology. Every accuracy improvement over a competing method reduces the number of requests that a cluster head would either drop in error or forward maliciously, which is exactly what the cluster head reliability indicators in Table 5.5 capture. The advantage over graph baselines is relevant here as well, since the higher accuracy of KG-DeepCNN over GraphSAGE (up to 2.52%) and BS-GAT (up to 4.53%) implies correspondingly fewer gating errors at the cluster head than those baselines would produce on the same traffic.

The robustness of the detector across four heterogeneous datasets further supports its suitability for reliable communication. VANETs are characterized by rapidly changing topologies and diverse traffic behaviours, and detectors that rely solely on local aggregation or on similarity heuristics can misclassify under these conditions. By contrast, the combination of knowledge-driven GCN embeddings with a heterogeneous DeepCNN classifier produces representations that are both discriminative and resilient to noise, and that adapt to evolving traffic. This resilience underpins the consistent accuracy across datasets and, through the access-control mechanism at the RSU and the aggregation performed by each cluster head, sustains the availability and integrity of V2V communication under attack.

5.6 Chapter Summary

This chapter addressed reliable and secure communication in VANETs through a graph-based deep learning intrusion detection framework deployed at the RSU. Motivated by the observation that frequent legitimate data exchange leaves RSUs highly exposed to low-overhead denial-of-service attacks, and that conventional cryptographic and policy-based defences are ill-suited to highly mobile vehicles, the chapter proposed KG-DeepCNN, an IDS-aware access-control scheme. The framework first constructs a knowledge graph in which nodes represent attacks and packet attributes and edges encode semantic relations such as similarity, causality, and co-occurrence, and it then uses a graph convolutional network to generate orthogonal, semantically enriched embeddings. These embeddings are classified by a

heterogeneous weighted ensemble of four CNN tunnels of increasing depth, which captures both fine-grained and global patterns and improves generalization over single or homogeneous models.

22 Evaluated on the BoT-IoT, ToN-IoT, UNSW-NB15, and NSL-KDD datasets, the framework achieved accuracies of 99%, 98%, 98%, and 99%, respectively, and improved on state-of-the-art methods by up to 4.53%, including a 1.01% gain over a strong SSAE and BiLSTM baseline on UNSW-NB15. A threshold analysis identified a cosine-similarity threshold of 0.5 as the best trade-off between graph density and noise, and confusion-matrix analysis confirmed strong per-class detection, including flawless detection of data exfiltration on BoT-IoT and zero false negatives on UNSW-NB15. Because the detector operates as an access-control gate at the RSU, its high precision and recall translate directly into reliable communication, since legitimate V2V exchanges are preserved through low false-positive rates while malicious flooding and probing are blocked through high recall. Addressing the objective of analyzing cluster head reliability, the chapter derived, from the per-class detection results, a composite cluster head reliability of between 98.43% and 99.38% across the four datasets, showing that the cluster head both sustains service for honest members and contains hostile traffic under the proposed model. Building on these results, the next stage of the research aims to profile the framework on RSU-grade embedded hardware to obtain latency and memory benchmarks, and to extend the evaluation to VANET-specific datasets such as VeReMi and DSRC traces.

CHAPTER 6

CONCLUSION, FUTURE SCOPE AND SOCIAL IMPACT

6.1 Conclusion

This thesis addressed the twin challenges of stability and reliability in vehicular ad hoc networks, where the high mobility of vehicles makes the network topology volatile, so that frequent cluster head re-election degrades communication, while the constant exchange of messages between vehicles and roadside units exposes safety critical services to intrusion. The work was guided by four research objectives: to study and analyze the different clustering algorithms used in VANETs, to develop intelligent algorithms based on scalable spectral clustering, to implement an efficient stability-based model that can be compared with recent state of the art works, and to analyze the reliability of the cluster head in the proposed models.

The first objective was met through a foundational study of the field, comprising a bibliometric analysis of clustering research in VANETs [121] and an investigation into the reliability of vehicle to vehicle and vehicle to infrastructure communication [122]. Building on this understanding, the second and third objectives were realised through two spectral clustering frameworks. The first framework [123] represented the network as a weighted hypergraph, fixed the optimal number of clusters through an eigen gap heuristic, formed clusters with an improved tensor trace maximisation (iTMM) spectral procedure, and elected cluster heads through an objective weighted CRITIC decision over four vehicle criteria, achieving an 84% cluster head stability with only three clusters together with improved throughput and reduced packet delay. The second framework [124] retained the hypergraph foundation but introduced a modified tensor trace maximisation pipeline with Gaussian mixture assignment and an incentivized cluster head selection mechanism, achieving a 75% stability with three clusters, a 4.17% improvement over the hypergraph baseline, and consistently low packet delay across sparse and dense networks. This incentive driven, behaviour aware selection was further advanced through an Adaptive Neighbour Behaviour approach

that rewards steady vehicles and penalises erratic ones [125], directly supporting the fourth objective by producing more reliable cluster heads.

Reliability of communication was finally reinforced at the security layer through KG-DeepCNN, a knowledge graph driven deep convolutional neural network intrusion detection system deployed at the RSU for intrusion aware access control [126]. Evaluated on the BoT-IoT, ToN-IoT, UNSW-NB15 and NSL-KDD benchmarks, this model reached accuracies of approximately 99%, 98%, 98% and 99% respectively, with a 2.01% accuracy gain on UNSW-NB15 over state-of-the-art techniques. Taken together, the studies show that objective, behaviour aware clustering on a scalable spectral foundation [123], [124], [125], informed by systematic analysis of the field [121], [122] and protected by knowledge driven intrusion detection [126], yields VANET communication that is stable, reliable and secure.

6.2 Research Contributions

The contributions of this thesis are organised below according to the four research objectives, with the associated publications indicated.

Objective 1: To study and analyze the different clustering algorithms in VANETs

- A bibliometric analysis of clustering in VANETs that maps the research landscape, identifies dominant methods, influential works and open gaps, and thereby motivates the direction of this thesis [121].
- An investigation into the reliability of vehicle to vehicle and vehicle to infrastructure communication that characterises the factors affecting link dependability and establishes the reliability criteria later used to design and assess the clustering models [122].

Objective 2: To develop intelligent algorithms based on scalable spectral clustering

- A weighted hypergraph representation of the vehicular network combined with an eigen gap heuristic that determines the optimal number of clusters without prior assumption, and an improved tensor trace maximisation spectral procedure built on a modularity-based Laplacian [123].

- A modified tensor trace maximisation clustering pipeline that replaces k-means with a probabilistic, covariance aware Gaussian mixture assignment suited to the irregular geometry of urban traffic [124].

Objective 3: To implement an efficient stability-based model and compare with state-of-the-art works

- An objective weighted CRITIC cluster head selection scheme over four vehicle attributes that removes the subjectivity of manually assigned weights and attains 84% cluster head stability with only three clusters, benchmarked against representative state of the art schemes [123].
- An incentivized stable cluster head selection mechanism that delivers 75% stability with three clusters, a 4.17% improvement over the hypergraph baseline, evaluated on a realistic urban scenario derived from Connaught Place, Delhi, under sparse and dense densities [124].
- An Adaptive Neighbour Behaviour incentivization strategy that refines cluster head election through reward and penalty rules applied to adaptive behavioural metrics, extending the stability model to dynamic neighbourhood conditions [125].

Objective 4: To analyze the reliability of the cluster head of the proposed models

- A reliability analysis of the elected cluster heads through metrics such as cluster head stability, cluster lifetime and packet delay across varying vehicle densities, confirming durable clusters with low overhead [123], [124].
- Adaptive behavioural metrics, namely the adaptive relative average speed, the adaptive neighbourhood degree and the adaptive link lifetime, that quantify and improve the dependability of cluster heads under changing traffic [125], grounded in the communication reliability criteria established earlier [122].
- An IDS aware access control layer, KG-DeepCNN, in which the RSU screens every data access request so that reliable cluster-based communication is also secure, using knowledge graph and graph convolutional network embeddings classified by a weighted ensemble of four heterogeneous CNN tunnels [126].

6.3 Research Limitations

The following limitations delimit the scope of the present work.

- The eigen decomposition of the Laplacian, although computed once at maximum density, carries a high computational cost for very large networks [123], [124].
- The proposed frameworks assume an RSU assisted, centralised architecture, which represents a potential single point of failure [123], [124], [125].
- Security is not embedded within the cluster head election itself; the clustering frameworks do not screen malicious candidates [123], [124], [125], while the intrusion detection system operates at the RSU rather than inside the election procedure [126].
- The clustering evaluation was confined to a single urban scenario and two vehicle densities, so performance on highways and in other traffic regimes remains to be established [123], [124], [125].
- The intrusion detection model was validated on established benchmark datasets rather than operational vehicular traffic, ran on a cloud GPU platform so its latency and memory footprint on RSU grade embedded hardware are unverified, and relied on an empirically selected similarity threshold [126].

6.4 Future Research Directions

Several directions follow naturally from this work.

- Integrating trust and intrusion awareness directly into cluster head election so that malicious or compromised vehicles are detected and excluded at selection time, unifying the clustering [123], [124], [125] and security [126] threads of this thesis.
- Extending the frameworks towards the Internet of Vehicles with edge computing and edge intelligence, so that clustering, behavioural assessment and detection are performed closer to the vehicles and dependence on a single RSU is reduced [122], [123], [124].

- Improving the scalability of the spectral clustering procedures for very large networks and validating them across diverse urban and highway scenarios and a wider range of traffic densities [123], [124], [125].
- Profiling the intrusion detection system on RSU grade embedded hardware, and making its knowledge graph dynamic so that relations for newly emerging attacks are learned incrementally, with federated training across RSUs to preserve privacy [126].

6.5 Social Impact

Beyond its technical contributions, this research carries tangible social value. Vehicular networks underpin the safety services of intelligent transportation systems, and their usefulness depends on the timely, reliable and trustworthy delivery of messages such as collision alerts and hazard warnings. Stable cluster heads and low packet delay [123], [124], [125] help ensure that safety critical information reaches vehicles promptly, which can reduce the incidence and severity of road accidents, while dependable screening of requests at the RSU [126] protects the availability and integrity of those messages against attack. More stable clustering also lowers signalling overhead and supports efficient use of the wireless channel, contributing to smoother traffic management, reduced congestion and lower fuel consumption and emissions in dense urban settings, in line with the reliability goals that motivated the study [121], [122]. Safeguarding data access additionally protects the personal and location information of commuters, reinforcing digital privacy in public infrastructure. As cities move towards connected and cooperative mobility, communication that is simultaneously stable, reliable and secure underpins smart city services, emergency response coordination and public confidence in autonomous transport, ultimately supporting safer and more efficient roads for everyone.

REFERENCES

1. W. Ahsan, M. F. Khan, F. Aadil, M. Maqsood, S. Ashraf, Y. Nam, and S. Rho, "Optimized node clustering in VANETs by using meta-heuristic algorithms," *Electronics*, vol. 9, no. 3, p. 394, 2020.
2. G. Khayat, C. X. Mavromoustakis, G. Mastorakis, J. M. Batalla, H. Maalouf, and E. Pallis, "VANET clustering based on weighted trusted cluster head selection," in *Proc. Int. Wireless Communications and Mobile Computing Conf. (IWCMC)*, IEEE, 2020, pp. 623–628.
3. J. Mihelj, A. Kos, and U. Sedlar, "Implicit aggressive driving detection in social VANET," *Procedia Computer Science*, vol. 129, pp. 348–352, 2018.
4. Z. Y. Rawashdeh and S. M. Mahmud, "A novel algorithm to form stable clusters in vehicular ad hoc networks on highways," *EURASIP J. Wireless Communications and Networking*, vol. 2012, pp. 1–13, 2012.
5. F. Hu, Y. Zhu, J. Liu, and Y. Jia, "Computing communities in complex networks using the Dirichlet processing Gaussian mixture model with spectral clustering," *Physics Letters A*, vol. 383, no. 9, pp. 813–824, 2019.
6. H. R. Arkian, R. E. Atani, A. Pourkhalili, and S. Kamali, "Cluster-based traffic information generalization in vehicular ad-hoc networks," *Vehicular Communications*, vol. 1, no. 4, pp. 197–207, 2014.
7. H. Hu and M. J. Lee, "Graph neural network-based clustering enhancement in VANET for cooperative driving," in *Proc. Int. Conf. Artificial Intelligence in Information and Communication (ICAIIIC)*, IEEE, 2022, pp. 162–167.
8. S. Kamakshi and V. S. Sriram, "Modularity based mobility aware community detection algorithm for broadcast storm mitigation in VANETs," *Ad Hoc Networks*, vol. 104, p. 102161, 2020.
9. R. Kaur, R. K. Ramachandran, R. Doss, and L. Pan, "The importance of selecting clustering parameters in VANETs: A survey," *Computer Science Review*, vol. 40, p. 100392, 2021.
10. M. K. Jabbar and H. Trabelsi, "A novelty of hypergraph clustering model (HGCM) for urban scenario in VANET," *IEEE Access*, vol. 10, pp. 66672–66693, 2022.

11. M. V. N. de Zepeda, F. Meng, J. Su, X. J. Zeng, and Q. Wang, "Dynamic clustering analysis for driving styles identification," *Engineering Applications of Artificial Intelligence*, vol. 97, p. 104096, 2021.
12. Y. Zhang, Y. Tie, and Y. Liu, "Research on driving behaviour assessment based on evidence theory in 5G-VANET," *IET Communications*, vol. 16, no. 11, pp. 1344–1354, 2022.
13. T. Liu, S. Shi, and X. Gu, "Naive Bayes classifier-based driving habit prediction scheme for VANET stable clustering," *Mobile Networks and Applications*, vol. 25, pp. 1708–1714, 2020.
14. S. S. Wang and Y.-S. Lin, "PassCAR: A passive clustering aided routing protocol for vehicular ad hoc networks," *Computer Communications*, vol. 36, no. 2, pp. 170–179, 2013.
15. M. Ran and X. Bai, "Vehicle cooperative network model based on hypergraph in vehicular fog computing," *Sensors*, vol. 20, no. 8, p. 2269, 2020.
16. G. Liu, N. Qi, J. Chen, C. Dong, and Z. Huang, "Enhancing clustering stability in VANET: A spectral clustering-based approach," *China Communications*, vol. 17, no. 4, pp. 140–151, 2020.
17. J. E. Z. Gbadouissa, A. A. A. Ari, C. Titouna, A. M. Gueroui, and O. Thiare, "HGC: HyperGraph based clustering scheme for power-aware wireless sensor networks," *Future Generation Computer Systems*, vol. 105, pp. 175–183, 2020.
18. M. Ren, L. Khoukhi, H. Labiod, J. Zhang, and V. Veque, "A mobility-based scheme for dynamic clustering in vehicular ad-hoc networks (VANETs)," *Vehicular Communications*, vol. 9, pp. 233–241, 2017.
19. S. Bhandari, X. Wang, and R. Lee, "Mobility and location-aware stable clustering scheme for UAV networks," *IEEE Access*, vol. 8, pp. 106364–106372, 2020.
20. I. T. Abdel-Halim, H. M. A. Fahmy, and A. M. Bahaa-El Din, "Mobility prediction-based efficient clustering scheme for connected and automated vehicles in VANETs," *Computer Networks*, vol. 150, pp. 217–233, 2019.
21. S. Sennan, S. Ramasubbareddy, S. Balasubramaniyam, A. Nayyar, C. A. Kerrache, and M. Bilal, "MADCR: Mobility aware dynamic clustering-based routing protocol in Internet of Vehicles," *China Communications*, vol. 18, no. 7, pp. 69–85, 2021.
22. M. Mukhtaruzzaman and M. Atiquzzaman, "Junction-based stable clustering algorithm for vehicular ad hoc network," *Annals of Telecommunications*, pp. 1–10, 2021.

23. M. A. Khan, A. A. Khattak, A. Almogren, et al., "Cluster Stability-Driven Optimization for Enhanced Routing in Heterogeneous Vehicular Networks," *Vehicular Communications*, vol. 47, Art. no. 100745, 2024.
24. K. A. Awan, I. U. Din, A. Almogren, M. Guizani, and S. Khan, "StabTrust—A stable and centralized trust-based clustering mechanism for IoT enabled vehicular ad-hoc networks," *IEEE Access*, vol. 8, pp. 21159–21177, 2020.
25. P. Kumar, P. K. Dahiya, and B. K. Singh, "Modularized hypergraph clustering scheme model for stable VANET," *Concurrency and Computation: Practice and Experience*, vol. 34, no. 26, p. e7283, 2022.
26. M. S. Sheikh, J. Liang, and W. Wang, "Security and privacy in vehicular ad hoc network and vehicle cloud computing: A survey," *Wireless Communications and Mobile Computing*, vol. 2020, pp. 1–25, 2020.
27. M. S. Sheikh, J. Liang, and W. Wang, "A survey of security services, attacks, and applications for vehicular ad hoc networks (VANETs)," *Sensors*, vol. 19, no. 16, p. 3589, 2019.
28. Z. Khan, A. Koubaa, S. Fang, M. Y. Lee, and K. Muhammad, "A connectivity-based clustering scheme for intelligent vehicles," *Applied Sciences*, vol. 11, no. 5, p. 2413, 2021.
29. D. Singh, A. K. Maurya, Ranvijay, and R. S. Yadav, "A trust-based clustering approach to form stable clusters in vehicular ad hoc networks," *J. Ambient Intelligence and Humanized Computing*, vol. 14, no. 12, pp. 16209–16228, 2023.
30. Z. Khan, P. Fan, S. Fang, and F. Abbas, "An unsupervised cluster-based VANET-oriented evolving graph (CVoEG) model and associated reliable routing scheme," *IEEE Trans. Intelligent Transportation Systems*, vol. 20, no. 10, pp. 3844–3859, 2019.
31. M. L. Chiang, H. C. Hsieh, W. C. Tsai, T. L. Lin, and Y. H. Lin, "Increasing cluster stability in VANET by candidate cluster head nomination algorithm," *Peer-to-Peer Networking and Applications*, vol. 19, Art. no. 104, 2026, doi: 10.1007/s12083-026-02263-1.
32. Ucar S, Ergen SC, Ozkasap O (2015) Multihop cluster-based IEEE 802.11 p and lte hybrid architecture for vanet safety message dissemination. *IEEE Trans Veh Technol* 65(4):2621–2636.
33. Abdulrazzak, H. N., Hock, G. C., Mohamed Radzi, N. A., Tan, N. M., & Kwong, C. F. (2022). Modeling and analysis of new hybrid clustering technique for vehicular ad hoc network. *Mathematics*, 10(24), 4720.

34. Zhao, H., Tang, J., Adebisi, B., Ohtsuki, T., Gui, G., & Zhu, H. (2022). An adaptive vehicle clustering algorithm based on power minimization in vehicular ad-hoc networks. *IEEE Transactions on Vehicular Technology*, 71(3), 2939-2948.
35. Oubabas, S., Aoudjit, R., Rodrigues, J. J., & Talbi, S. (2018). Secure and stable vehicular ad hoc network clustering algorithm based on hybrid mobility similarities and trust management scheme. *Vehicular Communications*, 13, 128-138.
36. Kumar P, Dahiya PK, Singh BK (2021) HGC: hypergraph-based dynamic stable clustering scheme model for vehicular ad-hoc networks (VANET). *IntJCommunSyst* 35:e5068.
37. M. Elhoseny, I. M. El-Hasnony, and Z. Tarek, "Intelligent Energy Aware Optimization Protocol for Vehicular Ad Hoc Networks," *Scientific Reports*, vol. 13, Art. no. 9019, 2023.
38. Radhika D (2021) Deep learned Ruzicka similaritive spectral clustered oppositional dragonfly opti mized routing In VANET. *Turk J Comput Math Educ* 12(12):3069–3086.
39. Khot PS, Naik U (2021) Particle-water wave optimization for secure routing in wireless sensor network using cluster head selection. *Wirel Pers Commun* 119:2405–2429.
40. Dutta AK, Elhoseny M, Dahiya V, Shankar K (2020) An efficient hierarchical clustering protocol for multihop Internet of vehicles communication. *Trans Emerg Telecommun Technol* 31(5):e3690.
41. Kadam, M. V., Vaze, V. M., & Todmal, S. R. (2023). TACR: Trust aware clustering-based routing for secure and reliable VANET communications. *Wireless Personal Communications*, 132(1), 305-328.
42. Yan, Chao, and Zhenhai Chang. "Modularized tri-factor nonnegative matrix factorization for community detection enhancement." *Physica A: Statistical Mechanics and its Applications* 533 (2019): 122050.
43. Wang, T., Cao, X., & Wang, S. (2019). Self-adaptive clustering and load-bandwidth management for uplink enhancement in heterogeneous vehicular networks. *IEEE Internet of Things Journal*, 6(3), 5607-5617.
44. Vasudeva, A., & Sood, M. (2022). On the vulnerability of the mobile ad hoc network to transmission power controlled Sybil attack: Adopting the mobility-based clustering. *Journal of King Saud University-Computer and Information Sciences*, 34(9), 7025-7044.
45. Mukhtaruzzaman, M., & Atiquzzaman, M. (2023). Stable dynamic feedback-based predictive clustering protocol for vehicular Ad hoc networks. *Computer Networks*, 235, 109960.

46. Salim, A., Khedr, A. M., & Osamy, W. (2023). IoVSSA: efficient mobility-aware clustering algorithm in internet of vehicles using sparrow search algorithm. *IEEE Sensors Journal*, 23(4), 4239-4255.
47. Elaraby, Samar, and Sherif M. Abuelenin. "Connectivity analysis of directed highway vehicular ad hoc networks using graph theory." *International Journal of Communication Systems* 34, no. 5 (2021): e4745.
48. Gbadoubissa, J. E. Z., Ari, A. A. A., & Gueroui, A. M. (2020). Efficient k-means based clustering scheme for mobile networks cell sites management. *Journal of King Saud University-Computer and Information Sciences*, 32(9), 1063-1070.
49. Kandali, K., Bennis, L., & Bennis, H. (2021). A new hybrid routing protocol using a modified K-means clustering algorithm and continuous hopfield network for VANET. *IEEE access*, 9, 47169-47183.
50. Pal, R., Gupta, N., Prakash, A., & Tripathi, R. (2018). Adaptive mobility and range based clustering dependent MAC protocol for vehicular ad hoc networks. *Wireless Personal Communications*, 98(1), 1155-1170.
51. B. Dhakad, S. Mishra, S. S. Ojha, et al., "Dynamic Clustering Based Risk-Aware Congestion Control Technique for Vehicular Network," *Scientific Reports*, vol. 14, Art. no. 24774, 2024, doi: 10.1038/s41598-024-75648-y.
52. Garg, D., Kaur, A., Benslimane, A., Bali, R. S., Kumar, N., Tanwar, S., ... & Obaidat, M. S. (2021, December). Truclu: trust-based clustering mechanism in software defined vehicular networks. In *2021 IEEE global communications conference (GLOBECOM)* (pp. 1-6). IEEE.
53. Babu, D., & Rao, T. S. (2024). A secure routing protocol using trust-based clustering and bionic intelligence algorithm for UAV-assisted vehicular ad hoc networks. *Transactions on Emerging Telecommunications Technologies*, 35(5), e4977.
54. Vengala, G. C., Pavuluri, V. K., Paranjothi, A., & Mukhtaruzzaman, M. (2025, October). TCESS: Trust Score-based Clustering Approach for Enhanced Security and Stability in Vehicular Ad hoc Networks. In *2025 IEEE 102nd Vehicular Technology Conference (VTC2025-Fall)* (pp. 1-5). IEEE.
55. Awan, K. A., Din, I. U., Syed, T. A., Cengiz, K., & Almogren, A. (2025). Trustworthy Clustering for Wireless Energy Sharing in Vehicular Systems for Consumer Applications. *IEEE Transactions on Consumer Electronics*, 71(4), 12236-12243.

56. Mohamed, A., Bai, Z., Pang, K., Zhao, J., Xu, H., Zhang, L., ... & Kwak, K. (2023). Spectral clustering and deep reinforcement learning-based dynamic resource allocation in SM-MIMO vehicular system. *IEEE Transactions on Intelligent Transportation Systems*, 25(5), 3777-3792.
57. Feng, Z., Li, K., & Li, B. (2024). A spectral clustering-based deployment strategy for roadside units in vehicular edge computing environments. *Ad Hoc Networks*, 158, 103483.
58. M. A. Saleem, Z. Shijie, M. U. Sarwar, T. Ahmad, A. Maqbool, C. S. Shivachi, and M. Tariq, "Deep learning-based dynamic stable cluster head selection in VANET," *J. Advanced Transportation*, vol. 2021, p. 9936299, 2021.
59. Y. A., Liu, Z., Chan, K. Y., & Guan, X. (2020). Energy-spectral efficiency optimization in vehicular communications: Joint clustering and pricing-based robust power control approach. *IEEE Transactions on Vehicular Technology*, 69(11), 13673-13685.
60. Yang, F., Han, J., Ding, X., Wei, Z., & Bi, X. (2020). Spectral efficiency optimization and interference management for multi-hop D2D communications in VANETs. *IEEE Transactions on Vehicular Technology*, 69(6), 6422-6436.
61. Zhang, C., Huang, W., Niu, T., Liu, Z., Li, G., & Cao, D. (2023). Review of clustering technology and its application in coordinating vehicle subsystems. *Automotive Innovation*, 6(1), 89-115.
62. A. W. Khan, J. I. Bangash, S. Kamal, *et al.*, "Multi-Criteria Based Stable Clustering Technique for Vehicular Ad Hoc Networks," *Scientific Reports*, vol. 16, Art. no. 17086, 2026, doi: 10.1038/s41598-026-47837-4.
63. Kakkasageri, M.S., Manvi, S.S.: Multiagent driven dynamic clustering of vehicles in VANETs. *J. Netw. Comput. Appl. Netw. Comput. Appl.* 35(6), 1771–1780 (2012).
64. G. Kaur, M. Khurana, and A. Kaur, "Cluster Head Selection Metric in VANETs Under Dynamic Environmental Conditions," *Journal of Transformative Technologies and Sustainable Development*, vol. 9, art. no. 5, 2025, doi: 10.1007/s41314-025-00069-2.
65. M.-L. Chiang, H.-C. Hsieh, W.-C. Tsai, T.-L. Lin, *et al.*, "Increasing Cluster Stability in VANET by Candidate Cluster Head Nomination Algorithm," *Peer-to-Peer Networking and Applications*, vol. 19, art. no. 104, 2026, doi: 10.1007/s12083-026-02263-1.
66. El-Dalahmeh M, Adeel U (2023) Intrusion detection system for SDN based VANETs using a deep belief network, decision tree, and ToN-IoT dataset. In: 2023 IEEE IAS Global Conference on Emerging Technologies (GlobConET), pp 1–6.

67. Gad AR, Nashat AA, Barkat TM (2021) Intrusion detection system using machine learning for vehicular ad hoc networks based on ToN-IoT dataset. *IEEE Access* 9:142206–142217.
68. Jiang W, Lv X, Tao J (2024) A secure authentication framework for IoV based on blockchain and ensemble learning. *Vehicular Commun.* <https://doi.org/10.1016/j.vehcom.2024.100836>.
69. Prabhakaran V, Kulandasamy A (2021) Hybrid semantic deep learning architecture and optimal advanced encryption standard key management scheme for secure cloud storage and intrusion detection. *Neural Comput Appl* 33(21):14459–14479.
70. Lo WW, Layeghy S, Sarhan M, Gallagher M, Portmann M (2022) E-GraphSAGE: a graph neural network based intrusion detection system for IoT. In: *NOMS 2022–2022 IEEE/IFIP Network Operations and Management Symposium, Budapest, Hungary*, pp 1–9. 51.
71. Park SB, Jo HJ, Lee DH (2023) G-ids: graph-based intrusion detection and classification system for can protocol. *IEEE Access* 11:39213–39227.
72. Wang Y, Han Z, Du Y, Li J, He X (2025) BS-gat: a network intrusion detection system based on graph neural network for edge computing. *Cybersecurity* 8(1):27.
73. Wang Z, Wang J, Liu Y, Yang X, Qi F, Song W (2024) Privacy-preserving attribute-based access control scheme with intrusion detection and policy hiding for data sharing in VANET. *IEEE Internet Things J.* <https://doi.org/10.1109/JIOT.2024.3384753>.
74. Song J, Qin G, Liang Y, Yan J, Sun M (2024) DGIDS: Dynamic graph-based intrusion detection system for CAN. *Comput Secur.* <https://doi.org/10.1016/j.cose.2024.104076>.
75. B. Karthiga, D. Durairaj, N. Nawaz, T. K. Venkatasamy, G. Ramasamy, and A. Hariharasudan, "Intelligent intrusion detection system for VANET using machine learning and deep learning approaches," *Wireless Communications and Mobile Computing*, vol. 2022, 5069104, 2022.
76. Rossi GV, Fan Z, Chin WH, Leung KK (2017) Stable clustering for ad-hoc vehicle networking. In: *2017 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, pp 1-6.
77. Alsabah MKJ, Trabelsi H, Jerbi W (2021) Survey on clustering in VANET networks. In: *2021 18th International Multi-Conference on Systems, Signals & Devices (SSD)*. IEEE, pp 493-502.
78. Newman ME, Girvan M (2004) Finding and evaluating community structure in networks. *Physical Review E* 69(2):026113.

79. Shi J, Malik J (2000) Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22(8):888-905.
80. Ng A, Jordan M, Weiss Y (2001) On spectral clustering: analysis and an algorithm. *Advances in Neural Information Processing Systems* 14.
81. Zhou H, Zhang D, Xie K (2017) Accurate traffic matrix completion based on multi-Gaussian models. *Computer Communications* 102:165-176.
82. Hou W (2011) K-means vs. GMM & PLSA. Technical note.
83. L. Ding, C. Li, D. Jin, and S. Ding, "Survey of spectral clustering based on graph theory," *Pattern Recognition*, vol. 151, art. no. 110366, 2024.
84. F. Hirschberger, D. Forster, and J. Lücke, "A variational EM acceleration for efficient clustering at very large scales," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 12, pp. 9787-9801, Dec. 2022.
85. A. R. Krishnan, R. Hamid, and M. M. Kasim, "A modified CRITIC method to estimate the objective weights of decision criteria," *Symmetry*, vol. 13, no. 6, art. no. 973, 2021.
86. Krajzewicz D (2010) Traffic simulation with SUMO: simulation of urban mobility. *Fundamentals of Traffic Simulation*, pp 269-293.
87. Liu T, Shi S, Gu X (2020) Naive Bayes classifier based driving habit prediction scheme for VANET stable clustering. *Mobile Networks and Applications* 25:1708-1714.
88. Newman ME (2006) Modularity and community structure in networks. *Proceedings of the National Academy of Sciences* 103(23):8577-8582.
89. Vashishth V, Chhabra A, Sharma DK (2019) GMMR: a Gaussian mixture model based unsupervised machine learning approach for optimal routing in opportunistic IoT networks. *Computer Communications* 134:138-148.
90. Sharma R, Dinkar SK (2022) Selfish node detection by modularized deep NMF autoencoder based incentivized reputation scheme. *Cybernetics and Systems* 1-27.
91. Taherdoost H, Madanchian M (2023) Multi-criteria decision making (MCDM) methods and concepts. *Encyclopedia* 3(1):77-87.
92. T. Majumder, R. K. Mishra, S. S. Singh, P. K. Sahu, and R. Vohnout, "Cognitive-radio-based resource management for smart transportation: a sliding mode control approach," *IEEE Internet of Things Journal*, vol. 10, no. 21, pp. 18622-18632, 2023.
93. K. Kim, J. S. Kim, S. Jeong, J. H. Park, and H. K. Kim, "Cybersecurity for autonomous vehicles: review of attacks and defense," *Computers and Security*, vol. 103, 102150, 2021.

94. A. Alsarhan, A. R. Al-Ghuwairi, I. T. Almalkawi, M. Alauthman, and A. Al-Dubai, "Machine learning-driven optimization for intrusion detection in smart vehicular networks," *Wireless Personal Communications*, vol. 117, pp. 3129-3152, 2021.
95. R. S. Vitalkar, S. S. Thorat, and D. V. Rojatkhar, "Intrusion detection for vehicular ad hoc network based on deep belief network," in *Computer Networks and Inventive Communication Technologies*, Springer Singapore, pp. 853-865, 2022.
96. H. Jin, Z. Zhou, and P. Papadimitratos, "Future-proofing secure V2V communication against clogging DoS attacks," in *Proceedings of the 19th International Conference on Availability, Reliability and Security*, pp. 1-8, 2024.
97. H. Bangui, M. Ge, and B. Buhnova, "A hybrid data-driven model for intrusion detection in VANET," *Procedia Computer Science*, vol. 184, pp. 516-523, 2021.
98. A. V. Turukmane and R. Devendiran, "M-MultiSVM: an efficient feature selection assisted network intrusion detection system using machine learning," *Computers and Security*, vol. 137, 103587, 2024.
99. S. Ercan, M. Ayaida, and N. Messai, "Misbehavior detection for position falsification attacks in VANETs using machine learning," *IEEE Access*, vol. 10, pp. 1893-1904, 2021.
100. D. Durairaj, T. K. Venkatasamy, A. Mehbodniya, S. Umar, and T. Alam, "Intrusion detection and mitigation of attacks in microgrid using enhanced deep belief network," *Energy Sources Part A*, vol. 46, no. 1, pp. 1519-1541, 2024.
101. P. Kumar, G. P. Gupta, R. Tripathi, S. Garg, and M. M. Hassan, "DLTIF: deep learning-driven cyber threat intelligence modeling and identification framework in IoT-enabled maritime transportation systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 2, pp. 2472-2481, 2021.
102. P. Kumar, R. Kumar, G. P. Gupta, and R. Tripathi, "BDEdge: blockchain and deep-learning for secure edge-envisioned green CAVs," *IEEE Transactions on Green Communications and Networking*, vol. 6, no. 3, pp. 1330-1339, 2022.
103. T. Alladi, B. Gera, A. Agrawal, V. Chamola, and F. R. Yu, "Deepadv: a deep neural network framework for anomaly detection in VANETs," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 11, pp. 12013-12023, 2021.
104. J. Shu, L. Zhou, W. Zhang, X. Du, and M. Guizani, "Collaborative intrusion detection for VANETs: a deep learning-based distributed SDN approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 7, pp. 4519-4530, 2020.

105. W. Wang, X. Du, D. Shan, R. Qin, and N. Wang, "Cloud intrusion detection method based on stacked contractive auto-encoder and support vector machine," *IEEE Transactions on Cloud Computing*, vol. 10, no. 3, pp. 1634-1646, 2020.
106. K. H. Le, M. H. Nguyen, T. D. Tran, and N. D. Tran, "IMIDS: an intelligent intrusion detection system against cyber threats in IoT," *Electronics*, vol. 11, no. 4, 524, 2022.
107. R. H. Dong, H. H. Yan, and Q. Y. Zhang, "An intrusion detection model for wireless sensor network based on information gain ratio and bagging algorithm," *International Journal of Network Security*, vol. 22, no. 2, pp. 218-230, 2020.
108. G. Raja, S. Anbalagan, G. Vijayaraghavan, S. Theerthagiri, S. V. Suryanarayan, and X. W. Wu, "SP-CIDS: secure and private collaborative IDS for VANETs," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 7, pp. 4385-4393, 2020.
109. Y. Gao, H. Wu, B. Song, Y. Jin, X. Luo, and X. Zeng, "A distributed network intrusion detection system for distributed denial of service attacks in vehicular ad hoc network," *IEEE Access*, vol. 7, pp. 154560-154571, 2019.
110. W. W. Lo, S. Layeghy, M. Sarhan, M. Gallagher, and M. Portmann, "E-GraphSAGE: a graph neural network based intrusion detection system for IoT," in *NOMS 2022 IEEE/IFIP Network Operations and Management Symposium*, pp. 1-9, 2022.
111. Y. Wang, Z. Han, Y. Du, J. Li, and X. He, "BS-GAT: a network intrusion detection system based on graph neural network for edge computing," *Cybersecurity*, vol. 8, no. 1, 27, 2025.
112. G. Kaur and D. Kakkar, "Hybrid optimization enabled trust-based secure routing with deep learning-based attack detection in VANET," *Ad Hoc Networks*, vol. 136, 102961, 2022.
113. W. Jiang, X. Lv, and J. Tao, "A secure authentication framework for IoV based on blockchain and ensemble learning," *Vehicular Communications*, 2024.
114. M. K. Pulligilla and C. Vanmathi, "An authentication approach in SDN-VANET architecture with rider-sea lion optimized neural network for intrusion detection," *Internet of Things*, vol. 22, 100723, 2023.
115. R. Zhao, "NSL-KDD," *IEEE Dataport*, 2022.
116. N. Moustafa, B. Turnbull, and K. K. R. Choo, "An ensemble intrusion detection technique based on proposed statistical flow features for protecting network traffic of internet of things," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4815-4830, 2018.

117. N. Moustafa, "A new distributed architecture for evaluating AI-based security systems at the edge: network TON_IoT datasets," *Sustainable Cities and Society*, vol. 72, 102994, 2021.
118. J. Ashraf, M. Keshk, N. Moustafa, M. Abdel-Basset, H. Khurshid, A. D. Bakhshi, and R. R. Mostafa, "IoT-BoT-IDS: a novel statistical learning-enabled botnet detection framework for protecting networks of smart cities," *Sustainable Cities and Society*, vol. 72, 103041, 2021.
119. Y. Zhang, C. Yang, K. Huang, and Y. Li, "Intrusion detection of industrial internet-of-things based on reconstructed graph neural networks," *IEEE Transactions on Network Science and Engineering*, vol. 10, no. 5, pp. 2894-2905, 2022.
120. A. Haydari and Y. Yilmaz, "RSU-based online intrusion detection and mitigation for VANET," *Sensors*, vol. 22, no. 19, 7612, 2022.
121. A. Kumari, S. Kumar, R. S. Raw, and N. Bohra, "Clustering in vehicular ad hoc network: a bibliometric analysis," *SN Computer Science*, vol. 6, no. 2, 2025.
122. A. Kumari, S. Kumar, and R. S. Raw, "Investigating the reliability of vehicle to vehicle and vehicle to infrastructure communication," in *Proc. International Conference on Innovations in Computational Intelligence and Computer Vision*, Singapore: Springer, 2024, pp. 483-500.
123. A. Kumari, S. Kumar, and R. S. Raw, "Advancing VANET stability: enhanced cluster head selection with iTTM and weighted CRITIC," *The Journal of Supercomputing*, vol. 80, pp. 16133-16172, 2024.
124. A. Kumari, S. Kumar, and R. S. Raw, "Modified clustering and incentivized stable CH selection for reliable VANET communication," *Cluster Computing*, vol. 27, pp. 11983-12005, 2024.
125. A. Kumari, S. Kumar, and R. S. Raw, "Incentivized-based Adaptive Neighbour Behaviour Approach for Vehicular Ad hoc Networks," in *Proc. 6th International Conference on Data Science and Applications (ICDSA 2025)*, vol. 1772, July 16-18, 2025.
126. A. Kumari and S. Kumar, "Mitigating intrusion attacks in VANETs through reliable communication with graph-based deep learning approaches," *The Journal of Supercomputing*, vol. 81, 1467, 2025.