RESEARCH PAPER
Future Research in AI & IoT, 2026, 2(1)
Homepage: https://frai.journals.umz.ac.ir/
DOI: 10.22080/frai.2026.30523.1030

Decentralized Trust Management Model to Detect Malicious Nodes in the Internet of Vehicles

Ali Moradi[1]*, Nasser Yazdani[1]

[1] *School of Electrical & Computer Engineering, University of Tehran, Tehran, Iran*

Article Info
Received 14 November 2025
Accepted 27 January 2026
Available online 31 January 2026

Keywords:
Internet of Vehicles;
Graph Neural Networks;
Malicious Node Detection;
Trust Management;
Trust Related Attacks

Abstract:
With the rapid expansion of the Internet of Vehicles, ensuring security and trust among nodes has emerged as a fundamental challenge in this domain. The open, dynamic, and distributed nature of these networks creates an environment conducive to malicious nodes that can compromise communication integrity and overall system security by disseminating false or misleading information. This research presents a hybrid, decentralized trust management model that, through a multilayer approach, can effectively detect and analyze malicious nodes in connected vehicular networks. The proposed framework adopts a two-layer structure: in the first layer, vehicles compute short-term local trust scores of their peers based on interaction data using the proposed LTrustAssess algorithm; while in the second layer, roadside units model the network as a graph and employ the proposed deep learning model, TemporalGATwithLSTM, to predict and update the global and long-term trust scores of nodes over time. Experimental evaluation on a dataset generated from simulated vehicular interaction logs demonstrates that the proposed model achieves higher accuracy and efficiency in the distribution of trust scores and in detecting malicious nodes than existing baseline approaches. Overall, by providing a scalable and adaptive mechanism, the proposed model enhances the security, trust, and efficiency of vehicular networks and represents a significant step toward realizing future intelligent and safe transportation systems.

***Corresponding Author:** ali.moradi7@ut.ac.ir

Supplementary information: Supplementary information for this article is available at https://frai.journals.umz.ac.ir/

Please cite this paper as: Moradi, A., & Yazdani, N. (2026). Decentralized Trust Management Model to Detect Malicious Nodes in Internet of Vehicles. Future Research on AI and IoT, 22-44. DOI: 10.22080/frai.2026.30523.1030

1. Introduction

The rapid advancement of communication technologies has positioned Intelligent Transportation Systems as a key application domain of the Internet of Things and wireless networks. These systems significantly contribute to improving road safety, reducing traffic congestion, and enhancing the quality of life [1][2]. Vehicular Ad Hoc Networks (VANETs) were initially introduced to support such applications, enabling vehicles to exchange real-time information with one another and with roadside infrastructure [3]. However, due to their highly dynamic topologies, limited coverage, and heterogeneous wireless communication environments, VANETs face numerous challenges [3]. To overcome these limitations, the concept of the Internet of Vehicles (IoV) has emerged as an evolution of VANETs,

integrating IoT technologies to enable vehicles to operate as intelligent, connected nodes capable of sensing, processing, and sharing critical traffic and environmental data [3]. Through Vehicle-to-Vehicle (V2V), Vehicle-to-Infrastructure (V2I), and broader Vehicle-to-Everything (V2X) communications, IoV facilitates safer driving, efficient traffic management, and more reliable route planning [2][3][4][5]. Despite these advantages, IoV environments remain highly vulnerable to security and trust issues due to their open, large-scale, highly dynamic and distributed nature [5][6][7]. Malicious nodes can disseminate misinformation, disrupt communication, and jeopardize road safety, potentially leading to severe consequences, including traffic manipulation, chain collisions, and large-scale urban crises [7]. Such characteristics make IoV prone to both external attacks and insider threats, where authenticated nodes may inject falsified messages [8].

Therefore, one of the fundamental challenges in IoV is ensuring reliable communication between vehicles and infrastructure components such as Roadside Units (RSUs) [2][3][5]. The presence of malicious nodes exacerbates this challenge, as they may inject falsified data or deliberately trigger accidents, undermining the safety and stability of the entire transportation system [2][3][4][5]. The highly

ISSN 3092-7552

dynamic topology of vehicular networks, the massive scale of data exchange, and the short-lived interactions between nodes further increase the system's susceptibility to internal threats [8]. Figure 1 shows different types of communications in the Internet of Vehicles.

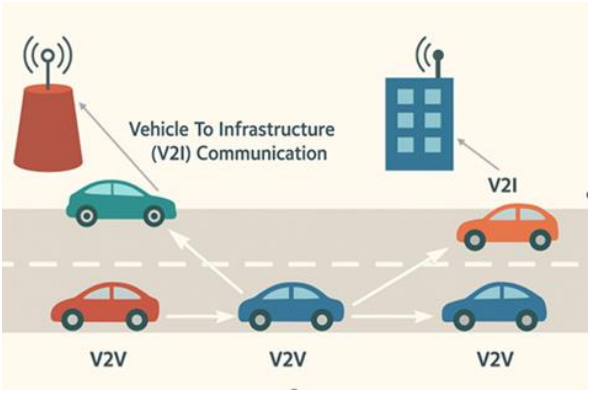

Figure 1. Types of IoV Communications

Traditional security mechanisms, such as encryption and authentication, are effective against external threats but insufficient for addressing internal attacks launched by compromised yet authenticated nodes. This highlights the need for trust management frameworks that can continuously evaluate the reliability of participating entities and detect malicious behaviors [2][3][4][5].

Trust management has emerged as an effective approach to address these issues by continuously evaluating node behavior, assigning trust scores, and isolating malicious participants from the network [2][3][4]. Unlike Misbehavior detection methods that identify misbehavior only at the data level, trust management provides a more comprehensive framework by considering long-term behavioral patterns and collective feedback from multiple entities [8]. However, designing robust and efficient trust management systems for IoV remains a significant open research direction, as existing solutions often lack comprehensive trust attributes, are limited in their adaptability to dynamic environments, or are vulnerable to trust-related attacks.

In response, this research proposes a decentralized, AI-enabled trust management framework that leverages deep neural networks for dynamic trust evaluation. By integrating both data-centric and node-centric perspectives, the proposed model aims to optimize trust score computation, enhance detection of malicious nodes, and strengthen the resilience of IoV communications.

The primary objective of this study is to design and implement a trust management model that enhances the security and reliability of IoV communications. By enabling accurate and timely identification of malicious entities, this model ensures trustworthy data exchange among vehicles, thereby improving network robustness and safety [2][3][4].

The specific contributions of this research are as follows:

- A two-layer trust management model (local and global) is proposed, combining data-centric and node-centric trust evaluation, and relying on consensus among RSUs to support decentralized decision-making.
- A local trust evaluation algorithm (LTrustAssess) is introduced for computing trust scores of vehicles during simulation, enabling local detection of misbehavior & fake relayed events.
- A domain-specific IoV trust dataset is generated by extracting trust-related features from vehicle interactions in a realistic simulation scenario, providing a valuable resource for future research.
- A deep neural network model (TemporalGATwithLSTM) is developed to predict vehicles' global trust scores over time. By leveraging temporal patterns and graph-based feedback, the model improves the accuracy of malicious node detection and strengthens overall network resilience.

By combining distributed trust evaluation with deep learning, this research advances state-of-the-art IoV security solutions, offering a scalable, adaptive, and intelligent framework that addresses both short-term misbehavior detection and long-term trust assessment.

The remainder of this paper is organized as follows: Section 2 introduces the background and fundamental concepts of the Internet of Vehicles (IoV) and trust management in the IoV context. Section 3 provides an overview of related works and existing trust management approaches in vehicular networks. Section 4 presents the proposed decentralized AI-enabled trust management framework, including the local trust evaluation algorithm and the global trust prediction model. Section 5 discusses the experimental setup, dataset generation, and evaluation metrics, followed by the analysis of results & comparison with other works. Finally, Section 6 concludes the paper and outlines potential directions for future research.

2. Background

2.1. Internet of Vehicles

The Internet of Vehicles, as a key component of intelligent transportation systems and autonomous vehicle technologies, enables real-time data exchange and intelligent communication among vehicles, infrastructure, and other entities via wireless networks. Each vehicle, equipped with an On-Board Unit (OBU) and sensors, acts as a smart object that monitors the environment, shares traffic information, and enhances road safety and efficiency [4][5]. The Internet of Vehicles allows vehicles to communicate with each other and with roadside infrastructure by employing specialized wireless communication technologies to ensure low latency, high bandwidth, and reliable message exchange. Key standards

include IEEE 802.11p and the WAVE* (IEEE 1609.x) framework, which enable real-time communication in dynamic vehicular environments, and DSRC† designed for short-range safety-critical applications [4]. The SAE J2735 standard further defines message structures, most notably the Basic Safety Message (BSM), which vehicles broadcast every 100 ms within their communication range to share kinematic data such as position, speed, heading, and acceleration with nearby nodes [8][9]. These messages extend situational awareness and support safety applications such as collision avoidance and cooperative driving. IoV is characterized by numerous dynamic entities that continuously exchange information in real time. The key characteristics of IoV can be summarized as follows [4][5][10][11]:

No geographical restrictions: Vehicles can communicate freely within their transmission range, broadening the scope of potential threats.

High entity density: The number of connected vehicles and other entities is very large.

Massive data exchange: Communication volume in the network is extremely high.

Dynamic topology: Due to vehicle mobility, the network structure and neighboring nodes change rapidly.

Short-lived links: Connections between nodes are transient, often disrupted by rapid movement.

Unreliable wireless channels: Communication is affected by road conditions, relative speed and direction, vehicle types, and environmental obstacles.

Resource constraints: Vehicles have limited capacity to store long-term interactions and lack global, network-wide knowledge.

Scalability challenges: The number of nodes and neighbors can increase significantly over time.

While this capability enables efficient traffic management and intelligent transportation, it also introduces significant security and reliability concerns due to the highly interconnected nature of vehicles. The high volume of communication and rapid changes in network topology make secure and trustworthy interactions more critical than ever [12].

2.2. Trust Management

The concept of trust management entails establishing network communication only between trusted nodes. In these models, nodes are typically assigned a trust score, and are considered malicious if their score falls below a predefined threshold. Trust management provides a framework for evaluating the reliability of network nodes, thereby ensuring secure and dependable interactions. It continuously assesses node behavior, reputation, and protocol adherence using trust scores, with particular emphasis on mitigating internal threats posed by malicious authenticated nodes [1][2][3][4][5]. Therefore, the trust management system is responsible for managing the real-time and long-term trust of network nodes based on the legitimacy of messages received from other nodes or on the legitimacy of the nodes themselves.

In trust management, trust represents a node's reliance on another to behave as expected, encompassing both individual-level trust and overall system reliability [8][13]. Systems comprise two main entities: the trustor (the evaluating node) and the trustee (the evaluated node). Trust evolves over time based on behavior and history and is assessed using mechanisms that evaluate both direct interactions and recommendations from other nodes [2][3][4][5].

In the IoV context, trust management evaluates node reliability (vehicles and RSUs), validates exchanged data, and detects malicious nodes. Effective frameworks enable continuous management of short- and long-term trust based on message legitimacy and node behavior. Attackers may attempt to manipulate trust relationships or provide deceptive information to compromise the system; therefore, establishing a trust management framework to counter such trust-related attacks is essential to strengthening the security posture of IoV [2][3][4][5].

2.3. Trust Management Models

Trust management models in the Internet of Vehicles are generally classified into three categories: data-centric, entity-centric, and hybrid models [2][3][5].

Data-centric approach (what data is provided): These models focus on evaluating the accuracy and reliability of the content of exchanged messages, such as position, speed, or event warnings, to detect misbehavior or attacks related to data, regardless of the source of the data sender. Trust assessment in this approach is usually short-term and does not establish a long-term relationship between vehicles. A major limitation is the dependence on sufficient data for each event, while historical interactions are not utilized. [2][3][4][5].

Node/Entity-centric approach (who provides the data): These models focus on the reputation and reliability of individual nodes by evaluating their past behavior and interactions. Trust values are typically long-term and reflect the historical performance of nodes. However, in scenarios with limited interactions or short-lived communication links, effective trust evaluation can be challenging [2][3][4][5].

Hybrid Approaches: Hybrid approaches combine both data-centric and entity-centric evaluations to provide a more comprehensive assessment. They consider both the

* Wireless Access in Vehicular Environments

† Dedicated Short Range Communications

accuracy of received messages and the sender node's overall behavior over time. While hybrid models generally offer higher detection accuracy and robustness against a wider range of threats, their trust computation process is inherently more complex [2][4][5].

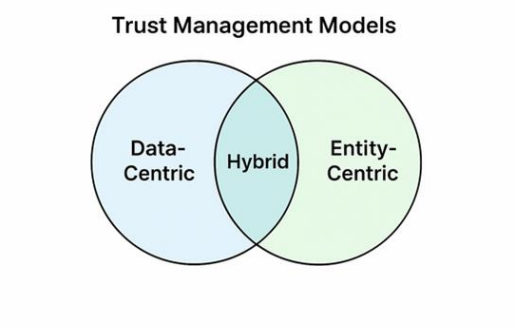

Figure 2. Trust Management Models

2.4. Components of Trust Management Systems

Trust management systems in IoV typically consist of three key components [14]:

Trust Sources: This includes direct and indirect trust [2][4][5].

Direct Trust: Derived from historical direct interactions between nodes. Factors influencing direct trust include packet delivery ratio, similarity between nodes, familiarity, interaction duration and frequency, and timeliness of interactions [2][4][5][15][16].

Indirect Trust: Also known as recommendation-based trust, it is computed from the recommendations of neighboring nodes and considers factors such as confidence in neighbors, positive/negative feedback, and reputation [2][4][5].

Trust Architecture: Trust systems can be categorized as centralized or decentralized [2][5].

Centralized Models: A central trusted server collects, computes, and stores trust values for all vehicles. While simple, these models are less suitable for highly dynamic vehicular environments due to single points of failure and scalability limitations [2][3][4].

Decentralized Models: Multiple nodes collectively manage trust computation, improving scalability and resilience against failures, and better accommodating the dynamic and distributed nature of IoV networks [2][3][4].

Trust Computation Algorithms: Trust computation algorithms can be classified into traditional and learning-based approaches [2][4][5][15][17].

Traditional Algorithms: Include statistical or rule-based methods such as weighted sum, weighted average, fuzzy logic, entropy, and Bayesian inference. These algorithms are computationally simple and fast [5][15][17].

Learning-Based Algorithms: Use machine learning or deep learning techniques to compute more accurate, dynamic trust scores, offering greater accuracy and adaptability than traditional methods [5][15][17].

2.5. Attacks on Trust Management Systems

Due to the inherent characteristics of vehicular networks, IoV trust management systems are vulnerable to a variety of attacks by malicious nodes [18]. Common attack types include:

Bad-Mouthing Attack: Malicious nodes provide false negative feedback about honest nodes to reduce their trust scores.

Ballot-Stuffing (Good-Mouthing) Attack: Colluding malicious nodes provide false positive feedback to increase trust scores of each other.

On-Off (ZigZag) Attack: Nodes alternate between good and malicious behavior to avoid detection.

Selective Misbehavior Attack: Malicious nodes target only specific nodes with false messages, causing inconsistencies in trust evaluation.

Self-Promoting Attack: Nodes attempt to increase their own trust scores by manipulating feedback without necessarily targeting other nodes.

3. Related Works

This section reviews prior research on trust management in vehicular networks, with the aim of establishing baselines for subsequent research. We then categorize prior studies based on our research contributions into two key areas: (i) architectures and approaches for trust management models in vehicular networks, focusing on decentralized and hybrid designs; and (ii) learning-based approaches for computing node trust scores, emphasizing graph-based models.

3.1. Baseline Works

Several baseline trust management models have been proposed in vehicular networks and have since served as reference approaches for subsequent studies. In [19], Xiao et al. (2019) introduced the IWOT-V model, which was inspired by the PageRank algorithm and designed to evaluate trust by constructing an implicit trust graph from dynamic interactions among vehicles. The architecture employed a hybrid centralized–distributed structure in which vehicles computed local trust values (LTVs) via Bayesian inference, roadside units (RSUs) collected these values, and a central system computed global trust values (GTVs) using a Vehicle Rank mechanism. Simulation results in a realistic urban scenario demonstrated high accuracy in distinguishing trustworthy from untrustworthy vehicles, even with up to 20% malicious nodes, which led to its widespread use as a reference baseline. However, the approach relied on a limited set of trust features, lacked mechanisms to counter good-mouthing and bad-mouthing attacks, and assumed fully trustworthy RSUs and central servers. Building on this, in [20] Zhang et al. (2020) proposed the AATMS system, which adopted a Trust Rank-inspired strategy by emphasizing recent interactions while

retaining a memory of past misbehavior. Local trust was computed via Bayesian inference with a beta distribution and an adaptive forgetting factor, whereas global trust was computed by constructing a trust graph and applying a modified Trust Rank algorithm. To improve robustness, seed nodes were selected based on PageRank rankings combined with social factors, and trust propagation employed an adaptive decay factor to slow down sudden trust increases while accelerating decreases. Simulation in a highway scenario showed that AATMS outperformed IWOT-V in resisting specific attacks, including Newcomer, ZigZag, and Colluding. Nonetheless, AATMS still overlooked common attacks such as good and bad mouthing, was highly dependent on sufficient interactions between vehicles (causing delays in sparse networks or for newcomers), and maintained an inherently centralized architecture, assuming fully reliable RSUs and trusted authorities. Together, these baseline approaches highlight the importance of dynamic trust evaluation but also reveal limitations in scalability, feature diversity, and resilience against a broader range of attacks, motivating the need for more adaptive, decentralized, and AI-driven trust management frameworks in the Internet of Vehicles.

3.2. Architecture & Approaches for TM Models

Beyond baseline trust models, several studies have focused on designing specific trust architectures for vehicular networks, often integrating decentralized mechanisms, multi-criteria decision-making, or blockchain- and AI-based approaches. In [21], PuCong (2021) proposed Trust Block MCDM, a decentralized system in which vehicles periodically upload locally computed trust values of message senders to nearby RSUs. RSUs aggregate these inputs using a multi-criteria decision-making framework and encapsulate the resulting reputation values into blocks, which are then competed for inclusion in the blockchain. Simulation results in OMNeT++ indicated improved detection of falsified messages and malicious vehicles; however, reliance on simple statistical methods rather than learning-based models limited adaptability in dynamic environments. In [22], Zhang et al. (2021) proposed a blockchain-assisted AI-driven trust management framework where vehicles use feedforward neural networks to compute local trust values, which are then aggregated by RSUs into global trust levels (GTLs). These GTLs are recorded immutably on the blockchain, with cross-RSU consensus ensuring consistency. While SUMO-based simulations showed improved detection accuracy and recall, the approach suffered from high computational overhead at the vehicle level due to neural network execution, and the use of simple averaging for global trust aggregation raised concerns about adaptability to dynamic IoV scenarios.

In a more advanced direction in [23], Wang et al. (2022) proposed a deep learning–enabled trust management framework coupled with blockchain. In this architecture, RSUs employ deep learning to assess message reliability, while a public blockchain is used to record traffic-related events. A proof-of-trust consensus mechanism further incentivizes vehicles with higher trust scores to participate as block miners, thereby integrating trust management with incentive structures. Although the model showed promising detection rates in SUMO simulations across both dense and sparse network settings, the blockchain component remained largely conceptual, with evaluations focusing primarily on the deep learning module. In [24], Cheong et al. (2024) advanced this line of work with the PBTMS model, which combines entity trust and path trust within a multilayer architecture. By analyzing message paths through RSUs and incorporating mechanisms such as marker trust and dynamically updated thresholds, PBTMS achieved higher accuracy, recall, and F-measure than IWOT-V and demonstrated resilience against MITM, Black Hole, and On-Off attacks. Nevertheless, its dependency on fully trusted RSUs and reliance on basic weighted aggregation limited its scalability in more decentralized IoV environments.

More recently, in [16], Wang et al. (2024) presented TM-IoV, the first multi-label dataset dedicated to trust management in the Internet of Vehicles. TM-IoV consists of 96,707 recorded interactions among 79 vehicles in a realistic simulation of the city of Jinan, China. To capture the dynamic nature of vehicular trust, nine key trust-related parameters were extracted for each trustor–trustee pair: Packet Delivery Ratio, Similarity, External Similarity, Internal Similarity, Familiarity, External Familiarity, Internal Familiarity, Reward and Punishment, and Context. These parameters incorporate both direct and indirect interactions, as well as behavioral history, making the dataset particularly suitable for machine-learning–based trust analysis. Intelligent malicious nodes were also introduced, employing strategies such as On-Off attacks to evade detection. The dataset was generated using a Java-based IoV simulator, but it was not publicly released. Moreover, the authors did not validate the dataset using machine learning or deep learning models to assess its reliability and effectiveness, which is a notable limitation.

Collectively, these works underscore the growing shift toward decentralized, AI-integrated trust architectures. Yet, they also reveal persistent challenges, including scalability, computational overhead on vehicles, and the oversimplification of trust aggregation methods.

3.3. Learning-Based Approaches

With the increasing complexity of IoV and the dynamicity of malicious behavior, learning-based approaches have emerged as a powerful direction for trust management and misbehavior detection in the Internet of Vehicles.

In [25], Eziama et al. (2018) extended this line of research by combining machine learning with deep learning in a trust-oriented detection model. Their hybrid approach modeled trust as a classification process, leveraging Bayesian deep neural networks to capture both probabilistic decision-making and generalization. While effective at identifying malicious nodes, the model still relied solely on exchanged data and did not incorporate historical behavioral records, thereby reducing its robustness against adaptive

attackers. In [26], El-Sayed et al. (2020) introduced an entity-based trust management framework that combines decision-tree classification for rule extraction with artificial neural networks for retraining when trust estimation is insufficient. Their model employed role- and distance-based metrics, such as Euclidean distance, and demonstrated superior performance compared to existing approaches. This work highlighted the potential of hybrid ML–based models but remained preliminary in its validation.

In [27], Siddiqui et al. (2023) addressed two critical challenges: assigning weights to trust features and defining threshold trust values for detecting malicious nodes. Using the CRAWDAD IoT dataset (adapted for IoV), they designed a dynamic machine-learning–based trust-evaluation framework. By combining unsupervised learning for ground-truth generation with supervised methods such as Subspace KNN and Subspace Discriminant, their framework achieved nearly perfect classification results. However, reliance on a non-IoV dataset raises concerns about generalizability to real vehicular scenarios. In [28], Wang et al. (2024) introduced the MESMERIC model, using a machine learning model to assess trust. This model accounts for direct and indirect interactions and includes contextual information, such as vehicle type and operational scenario. This model was evaluated using metrics such as precision, recall, and F1-score and showed high accuracy (up to 100% in urban scenarios) in identifying malicious nodes. In this paper, the authors used trust-related parameters such as direct trust (interaction success rate, familiarity, similarity, reward, and punishment) and indirect trust (feedback from neighbors) to evaluate the model. The machine learning algorithms K-Nearest Neighbor and Random Forest were also used to assess trust. The achievements of this paper include high precision, recall, and F1-Score on the Epinions dataset. However, one limitation of this paper is that the Epinions dataset is used, which is not related to vehicular networks (it concerns trust in social networks). Therefore, this dataset does not cover the dynamic nature of IoV.

In [29], Khan et al. (2024) combined deep neural networks with trust management for intelligent transportation systems. Their framework, trained on 150,000 samples including traffic patterns and sensor data, achieved 90% accuracy in identifying abnormal behavior. Trust scores were computed in the range [0,1], enabling the exclusion of nodes with low trust values. Despite outperforming classical ML algorithms such as Random Forest, SVM, and Naïve Bayes, the dataset used was not IoV-specific and lacked comprehensive trust-related parameters. Finally, in [30], Kushardianto et al. (2024) proposed a two-stage anomaly detection framework for IoV, employing Random Forest, LSTM, GRU, and DBN on two distinct datasets. Their results demonstrated improved detection performance compared to single-stage models, yet the approach remained highly dependent on data quality and introduced computational overhead that may limit real-time applicability. Moreover, trust evaluation was limited to data-level interactions, thereby precluding the formation of long-term or global trust.

Taken together, these studies underscore the growing reliance on machine learning and deep learning for trust management and misbehavior detection in IoV. While such approaches have demonstrated remarkable improvements in accuracy and robustness, their practical deployment is constrained by limitations including limited dataset availability, limited generalizability beyond non-IoV environments, and the integration of long-term behavioral history.

3.4. Graph-Based Models

Graph-based approaches have recently been proposed for modeling trust relationships. By leveraging graph neural networks (GNNs) and related architectures, these approaches aim to capture both the structural and contextual dependencies of trust, moving beyond traditional feature-based methods.

In [31], Jiang et al. (2022) introduced GATrust, a novel framework for pairwise trust evaluation in social networks. While most existing methods relied heavily on graph convolutional networks (GCNs) and largely ignored user-specific contextual features, GATrust combined multifaceted user information—including contextual data, topological structure, and locally formed trust relations—into a unified model. By integrating graph attention networks (GAT) with GCN, the framework assigned adaptive attention weights to different user features and learned latent trust factors between trustor–trustee pairs. Experiments on two real-world social trust datasets demonstrated improved accuracy in predicting trust. Although developed for online social networks, GATrust highlights the potential of attention-based graph models for IoV trust management, where contextual and relational features are equally critical. Building on the direction in [32], Wang et al. (2024) proposed TrustGuard, a graph-based trust evaluation model that incorporates temporal dynamics, attack resilience, and explainability. Operating in a decentralized architecture, TrustGuard treated trust interactions as temporal graphs and introduced a multilayered design consisting of: a snapshot input layer (time-based trust data), a spatial aggregation layer (defense-aware local aggregation resilient to attacks such as fake node injection), a temporal aggregation layer (attention-based learning of trust evolution), and a prediction layer for final trust computation. Experiments on Bitcoin-OTC and Epinions datasets under simulated attacks showed that TrustGuard outperformed state-of-the-art GNN models in both short- and long-term trust prediction, while remaining robust under adversarial conditions. Despite its success, adapting TrustGuard to IoV remains challenging due to the scarcity of real vehicular trust datasets. Most recently in [33], Favour et al. (2025) presented a GNN-based framework for malicious node detection in vehicular networks, marking one of the first attempts to apply deep graph learning directly to IoV trust management. The proposed architecture integrates message-passing layers, attention mechanisms, and readout layers for node-embedding aggregation, supplemented by dropout and normalization to enhance model stability. Temporal aspects

of trust were incorporated through methods such as time encoding and RNN-based integration. While conceptually innovative, the study lacked detailed implementation descriptions and did not provide experimental results, leaving its effectiveness unvalidated.

Together, these graph-based approaches demonstrate the potential of GNNs and attention mechanisms to advance trust management in IoV. They emphasize the need to account for contextual, structural, and temporal dimensions of trust while maintaining robustness against attacks. Nevertheless, their practical applicability in vehicular environments remains constrained by computational overhead and the limited availability of realistic IoV trust datasets.

4. Materials and Methods

In this section, we introduce the proposed methodology for trust management in the Internet of Vehicles. Connected vehicular networks are inherently dynamic and decentralized, posing significant challenges for trust and security [26]. This work presents a hybrid, decentralized trust management framework that integrates data-driven (message-content-based) and node-centric (behavior-based) evaluations to manage trust at both local and global levels. Unlike approaches that rely solely on recent interactions, the model incorporates historical behavior, reducing false classification of honest nodes and penalizing consistently misbehaving nodes. This framework employs a feedback- and consensus-based two-tier approach to address the limitations of centralized methods, including single points of failure and scalability issues, while providing accurate, dynamic, and fully decentralized trust evaluation [8]. The proposed model operates on a two-layer architecture: the local layer, deployed on vehicles, computes trust scores based on direct interactions and extracted relevant parameters such as position, speed, heading, and RSSI using the proposed LTrustAssess Algorithm. The global layer, implemented on RSUs, aggregates trust-related reports from vehicles and computes global trust scores using a proposed deep learning model, TemporalGAT with LSTM, combined with a consensus mechanism. So the proposed model is organized into three stages:

(i) Local Trust Assessment, where vehicles evaluate peers based on direct interactions;

(ii) Trust Reporting, where trust-related evidence is transmitted to RSUs that are in the vehicle's communication range;

(iii) Global Trust Assessment, where RSUs aggregate reports and compute final trust scores of the nodes.

The framework enhances both the reliability and security of connected vehicular networks, supporting robust decision-making in highly dynamic environments.

4.1. Local Trust Assessment

Local trust assessment is performed periodically by vehicles every 30 seconds, focusing on interactions that occurred within the preceding interval. The process adopts a hybrid approach, combining data-driven evaluation (analysis of received messages) and node-centric evaluation (assessment of sender behavior).

In this model, Vehicles broadcast Basic Safety Messages every second in their communication range. This model assumes that vehicles are authenticated. That is, all vehicles are pre-registered in the network and join it using a certificate issued by a Certificate Authority (CA). This means that only authorized OBUs can send/receive safety messages.

For data-driven evaluation, each Basic Safety Message received undergoes a set of basic checks by the receiver node, based on the F2MD [8] framework. These include both basic Plausibility (e.g., acceptable range, position, speed, heading, acceleration) and consistency check (e.g., position, speed, heading, position-speed correlation), computed over the last five messages received from the sender within the last 15 seconds. Anomalous behaviors, such as sudden appearance or abnormal message frequency, are also detected. Each check assigns a continuous score between 0 and 1 to the message, and the geometric mean of the check scores is used to calculate BsmScore, reflecting message trustworthiness. Different types of checks (plausibility & consistency) applied to each received message are listed in Table 1.

Table 1. List of Plausibility & Consistency checks applied on the received messages

Checks		
PlausabilityCheck	**Consistency Check**	**Kalman Filter-Based Anomaly Detection Check**
Proximity Plausibility	Position Consistency	Kalman Position Consistency
Range Plausibility	Speed Consistency	Kalman Speed Consistency
Position Plausibility	Position Speed Consistency	Kalman Position Speed Consistency
Speed Plausibility	Position Speed Max Consistency	KalmanPositionAcc Consistency
	Position Heading Consistency	

In parallel, node-centric evaluation computes sender-related trust features, including position similarity, speed similarity, heading similarity, familiarity (interaction frequency and interaction duration), packet delivery ratio, and event contribution. These features capture the sender's behavioral consistency and reliability over time.

The proposed LTrustAssess algorithm integrates these factors into a weighted scoring system with five primary components: Misbehavior Factor, Context Factor, Interaction Factor, Quality Factor, and Event Factor. The resulting score represents the local trust the receiver assigns

to the sender. Vehicles subsequently use this local trust score to evaluate the reliability of event-driven messages (WSMs) received from other node

Algorithm 1: LTrustAssessAlgorithm

Pseudocode for Local Trust Assessment Algorithm

```
Input: Data Related Features + Node Related Features (All features are between 0 - 1)
Output: LocalTrustScore  (0-1)
 Initialize
        TrustScore₀ = 0.7
        W1, W2, W3, W4, W5 = (0.35, 0.30, 0.15, 0.10, 0.10)
    Data-Related Features
          AvgBSMScore ← Avg(BSMScores)
         AvgNormalizedRSSI ← Avg(RSSINoramalized)
    Node-Related Features
          TotalSimilarity ← Avg(PosSimilarity, SpeedSimilarity, HeadingSimilarity)
          Familiarity ← f(duration, ReceivedBSMCount)
          PDR ← f(Expected Messages, ReceivedBSMCount)
         EventCoopScore ← f(EventCoopScore, NotFakeEventRatio)
        PDR_RSSI_Combined ←  Avg(PDR, AvgNormalizedRSSI)
    Execute
       MisbehaviorFactor (MF) ← AvgBSMScore
       ContextFactor (CF) ← TotalSimilarity
       InteractionFactor  (IF) ← Familiarity
       QualityFactor  (QF) ← PDR_RSSI_Combined
       EventFactor (EF)  ← EventContributionScore
       Calculate Sender Local Trust Score
       LocalTrustScore ← (W1 * MF) + (W2 * CF) + (W3 * IF) + (W4 * QF) + (W5 * EF)
       LocalTrustScore ← f(LocalTrustScore, dataWeight)
    END
```

Features used in LTrustAssess and their computations are as follows:

AvgBSMScore: The average score of all BSM messages received from a sender in the last 30 seconds. Each BSM is evaluated for plausibility and consistency using the F2MD framework [8], identifying abnormal behaviors such as sudden appearance, irregular frequency, or message modification.

Normalized RSSI: Normalized Received Signal Strength Indicator, representing the communication link quality. Higher RSSI indicates stronger connectivity and increases trustworthiness.

$$RSSI_{Normalized} = \frac{1}{1 + e^{-0.15(rssi+80)}}$$

Total Similarity: Average similarity between the sender and receiver in terms of position, speed, and heading during the last 30 seconds.

$$Pos_{Similarity} = Max(0, \frac{1-Pos\ difference}{Max\ Plausible\ Range})$$

$$Speed_{Similarity} = Max(0, \frac{1-Speed\ difference}{Max\ Plausible\ Speed})$$

$$Heading_{Similarity} = Max(0, \frac{1-Heading\ Differrnce}{180})$$

Familiarity: Measures the degree of prior interactions between sender and receiver, considering the number of messages exchanged and the duration of interaction.

$$MessageCount_{Normalized} = \frac{Log(1 + ReceivedCount)}{Log(1 + MaxReceived)}$$

$$Duation_{Normalized} = \frac{Log(1 + Duration)}{Log(1 + MaxDuration)}$$

$$Familiarity = 0.3 * MessageCount_{Normalized} + 0.7 * Duration_{Noramalized}$$

Packet Delivery Ratio (PDR): Indicates communication reliability by comparing expected versus received messages over the interaction duration.

$$ExpectedMessages = duration * MessageRate + 1$$

$$PDR = \frac{MessageReceivedCount}{ExpectedMessages}$$

Event Contribution (EventFactor): Assesses the sender's cooperation and accuracy in reporting events. Includes a reward/punishment mechanism that

penalizes incorrect event reports and rewards accurate contributions.

$$ExpectedEvents = duration * EventFrequency$$

$$EventRatio = \frac{EventReceivedCount}{ExpectedEvents}$$

$$NotFakeEventRatio = 1 - \frac{EventDetectedAsFake}{EventReceivedCount}$$

$$CooperationScore = X + 0.5 * EventRatio\ ()$$

{Where- if eventRatio < 1 then X = 0.5 and if eventRatio ≥ 1 then X = 1}

$$EventFactor = \frac{(Avg(CooperationScore, NotFakeEventRatio) - 0.25)}{1}$$

Data weight: To avoid premature judgments when limited observations exist, a Data weight factor (based on ReceivedBSMCount) moderates the influence of freshly computed scores versus the initial/default trust (0.7). With fewer messages, the algorithm favors cautious trust.

$$Data\_Weight = \min(1, \frac{MessageReceivedCount}{10})$$

The output LocalTrustScore ranges between 0 and 1, reflecting the sender node's trustworthiness from the receiver's perspective. These scores are then used to evaluate the reliability of event-driven messages (WSMs) and are reported to RSUs in the WSM_report message for subsequent global trust assessment.

$$LocalTrustScore = (LocalTrustScore * Data_{Weight}) + (TrustScore_0 * (1 - Data\_Weight))$$

4.2. Interaction Reporting to RSUs

Given the rapidly changing characteristics of IoV, in which neighboring nodes frequently change, and the number of neighbors varies significantly [5], and considering the limited computational resources available in vehicles [4], trust assessment cannot be efficiently performed solely at the vehicle level. To address this, each receiving vehicle maintains interaction records with other nodes over a 30-second interval. After computing the local trust score for each sender node, the vehicle generates a WSM_Report message reporting interactions during the interval and sends it to the nearest RSU(s) within its communication range.

Given the short-lived and highly dynamic interactions between sender and receiver nodes, vehicles discard these interaction records after reporting. This design reduces computational overhead on vehicles and focuses trust evaluation on short-term interactions. Consequently, a global trust assessment must be performed at the RSU level.

The WSM_Report message sent from vehicles to RSUs every 30 seconds includes the following trust-related parameters, which RSUs use to compute global trust scores:

Message WSM_Report: Message Sent from Vehicles to RSUs every 30 seconds

- ***ReceiverPseudoId*** ← *Pseudo-ID of BSM message receiver(Reprter to RSU)*
- ***SenderPseudoId*** ← *Pseudo-ID of BSM message sender*
- ***ReceivedBSMCount*** ← *Number of received BSMs*
- ***AvgBsmScore*** ← *Average(BSMScores)*
- ***Familiarit*** ← *InteractionHistory(Receiver, Sender)*
- ***AvgPosSimilarity*** ← *Average position similarity between (R,S) in BSMs*
- ***AvgSpeedSimilarity*** ← *Average speed similarity between (R,S) in BSMs*
- ***AvgHeadingSimilarity*** ← *Average heading similarity between (R,S) in BSMs*
- ***FirstInteractionTime*** ← *Timestamp of first interaction*
- ***LastInteractionTime*** ← *Timestamp of last interaction*
- ***Duration*** ← *LastInteractionTime - FirstInteractionTime*
- ***AvgRSSI*** ← *Average RSSI of received BSMs*
- ***PDR*** ← *PacketDeliveryRatio (communication quality)*
- ***EventFactor*** ← *Event interaction score*
- ***CalculatedTrustScore*** ← *Local trust score (from LTrustAssess formula)*
- ***ExitTime*** ← *Report submission time (end of 30s window or node exit)*

4.3. Global Trust Assessment

Global trust evaluation is performed by RSUs, which receive WSM_Report messages from multiple vehicles within their communication range at 30-second intervals. These messages contain trust-related feedback from receiving vehicles regarding sender nodes, summarizing interactions during the reporting interval. RSUs aggregate these reports and, after a network-dependent interval (shorter for high-density networks and longer for low-density networks; 5000 s in our simulation), compute global trust scores for all vehicles that were active in their coverage area. To provide a comprehensive analysis, RSUs must account for temporal variation and repeated interactions when evaluating each node's behavior.

To model these interactions, RSUs construct a feedback graph in which nodes represent vehicles and edges represent trust feedback between sender-receiver pairs [34]. In this graph, edges are directed from receivers to senders, and the value of each edge corresponds to the predicted trust score for that pair from the receiver's perspective. RSUs first predict trust scores for each sender-receiver pair over time based on historical interactions and behavioral changes to compute the edge score between each pair, and then aggregate the edge scores from multiple receivers to compute the overall global trust score for each sender node (graph vertex) [34][35][36][37][38]. The global trust score prediction will be done as follows.

4.3.1. Predicting Global Trust Scores

Aggregated global trust scores are predicted by RSUs using a pre-trained deep learning model trained on a simulation-generated dataset of vehicle interactions. To effectively model the network as a feedback graph, we employ Graph Attention Networks (GAT) [39] to capture the heterogeneous importance of trust feedback across vehicles.

In this model, each vehicle is represented as a node and trust feedback between sender-receiver pairs as directed edges, with associated node and edge features. GAT learn node embeddings by aggregating behavioral patterns from neighboring nodes through a message-passing process [35][37]. Also, by using attention mechanisms to assign different weights to neighbors, reflecting their relative importance. In the context of trust, these weights are learned based on trust-related parameters [39].

For this, RSUs first transform received interaction reports into sender-receiver pairs. The model then considers temporal sequences of interactions between each pair, incorporating all prior interactions at earlier time intervals from the perspective of different receivers, based on the history of the sender's interactions, to predict a pairwise trust score over time (edge trust). These predicted edge-level trust scores are aggregated across all receivers to compute the final global trust score for each sender node. To incorporate both temporal dynamics and network structure, we propose a TemporalGATWithLSTM model that combines graph attention with LSTM layers to capture spatial and temporal patterns in vehicle interactions. This approach enables RSUs to accurately predict dynamic trust levels for each vehicle, ensuring robust and scalable trust evaluation in highly dynamic vehicular networks.

4.3.2. TemporalGATwith LSTM

The proposed TemporalGATWithLSTM model is a hybrid deep learning architecture that integrates Graph Attention Networks (GAT) & Long Short-Term Memory (LSTM) networks to predict the aggregated global trust scores of vehicles in highly dynamic vehicular networks. This model can capture both graph-structured interactions among vehicles and the temporal evolution of interaction features.

Model Architecture: The proposed model is a hybrid model as follows:

LSTM layer: to model the temporal sequence of interactions between sender-receiver vehicle pairs. This layer processes the changes of input features such as AvgBSMScore, AvgSpeedSimilarity, AvgPosSimilarity, AvgHeadingSimilarity, Familiarity, Event Contribution, and PDR_RSSI_Combined over time, generating hidden states that capture historical interactions and contextual information for each pair. For each sender-receiver pair, multiple interactions may occur over time. The LSTM layer processes these temporal sequences and predicts the edge-level trust score for the latest interaction. Attention mechanisms ensure that interactions with **more messages exchanged** (high quality), longer durations, and more recent occurrences are weighted more heavily. ensuring that critical interactions have greater influence on edge-level trust scores. These edge-level trust scores are referred to as Receiver Feedback on the sender node's trust in this work. These values are then used in the GAT layer for node-level aggregation.

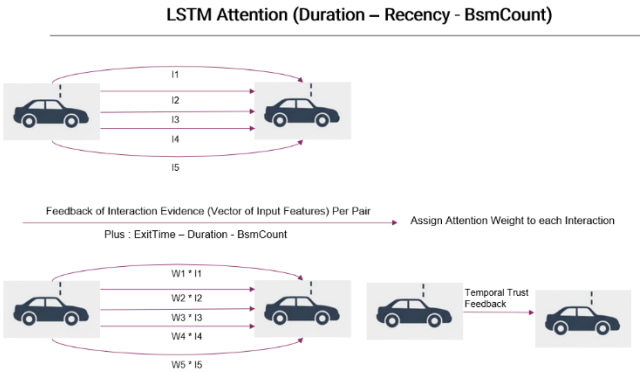

Figure 3. LSTM Attention

GAT layer: to represent the vehicle trust network as a graph of interactions between the sender and receiver vehicles, using the output of the LSTM and Attention Layer, where nodes are vehicles and incoming edges represent trust feedback from different receivers to senders. Since different feedback from different receivers doesn't have the same weight for the sender, this layer also uses an attention mechanism to assign different weights to different edge trust scores according to the importance of each receiver node. Then, Aggregates weighted edge-level trust scores (feedbacks) from different receiver nodes to compute node-level (aggregated) trust scores that reflect a comprehensive,

network-wide perspective. Weights are assigned based on multiple factors, including:

Source Node Trust: Trustworthiness of the receiver node providing feedback.

Source Node Recency: Recency of the interactions of the node providing feedback.

Source Node Degree: Degree of the receiver node (number of interactions in the network).

Feedback freshness: Feedback freshness based on the time elapsed since the last interaction.

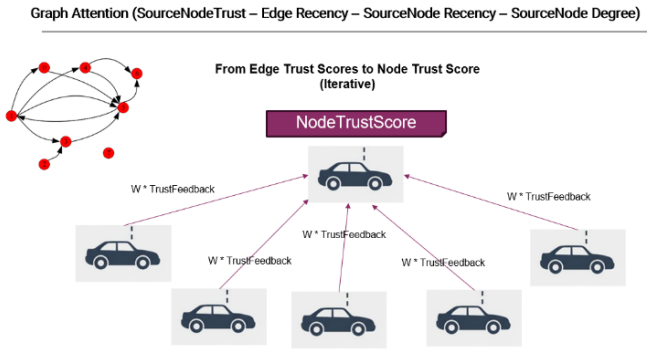

Figure 4. Graph Attention

Since edge weights depend on the trustworthiness of receiver nodes, which may themselves act as senders in other edges, an iterative procedure is needed for predicting the nodes' trust scores, which will be done as follows:

- Initial trust scores for all receiver nodes are set to zero.
- Node-level trust scores are computed based on Source Node Recency, Source Node Degree & Feedback freshness.
- Updated node-level trust scores are used to recalculate attention weights and refine trust aggregation.
- Iteration continues until convergence (typically 3–5 iterations).

Therefore, from the trust scores of different edges incoming to a node, we arrive at a node trust score. This trust score is referred to as RSU feedback on the sender node's trust. Figure 5 shows the structure of the proposed TemporalGATwithLSTM model.

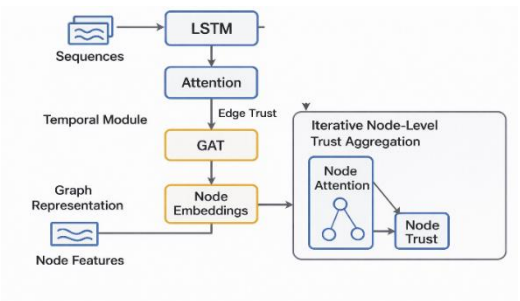

Figure 5. Structure of the proposed TemporalGATwithLSTM

4.4. Decentralized Consensus

To maintain decentralization, RSUs share their predicted aggregated trust scores with neighboring RSUs and use a consensus mechanism to determine the final global trust score for each vehicle, which is called the network feedback on the sender node trust. This approach ensures that predicted global trust is consistent and globally validated across the network, providing a robust, accurate, and decentralized view of trust.

4.5. State Update and Malicious Node Detection

Once the aggregated global trust score for each sender node is established after consensus, the network completes the current state (State N). To maintain temporal continuity in trust evaluation, the final global trust score of each node in the current state is calculated as the average of the newly computed global trust score and the final global trust score from the previous state:

$$fT_N = \frac{cT_N + fT_{N-1}}{2}$$

- $\boldsymbol{fT_N}$ = *Final Global Trust at state N*
- $\boldsymbol{cT_N}$ = *Calculated Global Trust at state N*
- $\boldsymbol{fT_{N-1}}$ = *Final Global Trust from previous state*

Note: Newly joined nodes start at State 0, with an initial global trust score of 0.7. At the end of each state, after RSU consensus, the global trust scores are updated across the network according to the above formula, ensuring that trust evaluation reflects both current behavior and historical performance.

The proposed trust management model enables the detection of malicious nodes using aggregated global trust scores. After predicting trust values using TemporalGATWithLSTM, a threshold-based mechanism is applied. Various thresholds between 0.4 and 0.7 (in increments of 0.05) were evaluated, and the optimal threshold was selected to maximize detection accuracy. A node is classified as malicious if its global trust score falls below the threshold; otherwise, it is classified as genuine.

Malicious, if Trust(i) < threshold (0.65)

Genuine, if Trust(i) ≥ threshold (0.65)

In this study, a threshold of 0.65 was determined to be optimal for distinguishing malicious nodes, enabling the network to punish or isolate such vehicles accordingly.

5. Performance Evaluation

To evaluate the performance of the proposed model, extensive simulations were conducted to assess the LTrustAssess algorithm and to generate the dataset

required for evaluating the TemporalGATwithLSTM model. So, the evaluation of the proposed model is organized into two main components: (i) vehicular network simulation for generating realistic interaction data and modeling malicious behavior for generating a dataset needed for further analysis, and (ii) implementation of the deep learning trust model for global trust prediction and malicious node detection.

5.1. Vehicular Network Simulation

Given the complexity and cost of deploying trust management models in vehicular networks, the proposed model is evaluated through simulation. Vehicular interactions are simulated to generate datasets for local trust computation, RSU-level trust reporting, and subsequent deep learning-based trust aggregation.

The simulation is implemented using the open-source VEINS[3] framework [40], which combines OMNeT++ (a discrete event network simulator) and SUMO[4] (microscopic traffic simulator). VEINS supports realistic V2X scenarios by integrating road traffic patterns from SUMO with wireless communication components, including an IEEE 802.11p MAC/PHY model in OMNeT++. The simulator therefore extracts and maintains the following per-message items during each 30-second local window from vehicle-to-vehicle interaction logs: BSMScore (per-message plausibility/consistency score), Familiarity, Position/Speed/Heading Similarities, AvgRSSI, Packet Delivery Ratio (PDR), event-related metrics (e.g., EventContribution, NotFakeEventRatio), and other intermediate values required by the LTrustAssess pseudocode described earlier to compute the local trust scores of each vehicle.

For V2I WSM_Report messages, four RSUs are deployed along roads in the simulation map. Their coordinates are chosen to ensure uniform coverage, such that the entire simulation area is within RSU coverage. Each RSU receives 30-second trust reports from vehicles currently inside its communication range and uses them for the RSU-level aggregation/learning components.

To incorporate malicious behaviors, the simulation is extended using the F2MD[5] framework [8], an open-source extension of VEINS designed for modeling attacks and misbehavior in vehicular networks. F2MD enables the injection of various malicious data-level actions, particularly at the BSM level, such as false message injection, data manipulation, and intentional delays [40][41]. The implemented data-level attacks are as follows:

ConstPos — broadcasts the same (constant) position in every BSM.

RandomPos — broadcasts a random position sampled from the simulation area.

Const PosOffset — broadcasts the same (constant) position plus a bounded random offset.

RandomPosOffset — broadcasts the real position plus a bounded random offset.

ConstSpeed — broadcasts a constant speed in all BSMs.

ConstSpeedOffset — broadcasts the same (constant) speed plus a bounded random offset.

RandomSpeed — broadcasts a random speed with a specified upper bound.

RandomSpeedOffset — broadcasts the real speed plus a bounded random offset.

StaleMessages — transmits authentic-looking but delayed (stale) BSMs (fixed delay before broadcast).

DoS, DoS_Random, DoS_Disruptive — increases BSM transmission frequency to flood the wireless channel (can be targeted or random) to disrupt communication availability.

Disruptive — repeatedly retransmits an older BSM from its history to confuse neighbors (replay of previously valid messages).

EventualStop — broadcast the speed = 0 in order to inject eventualStop.

DataReplay — selects a target and replays that target's past messages with a delay, creating an apparent tailing behavior (two vehicles appearing to travel closely).

Plus, these data-related attacks, some additional trust-feedback related attacks were developed and integrated into the F2MD framework for implementing trust-related attacks. The implemented attacks are:

Bad-Mouthing (False Negative feedback) — the malicious reporter lowers the reported local trust for honest nodes. Specifically, the malicious reporter probabilistically reduces the computed sender-node trust score within a bounded range before sending it to the RSU. The goal is to cause false accusations and isolate honest nodes.

Good-Mouthing (Collusion / False Positive feedback) — the malicious reporter increases reported trust within a bounded range for colluding malicious

[3] VEhicles In Network Simulation

[4] Simulator of Urban MObility

[5] Framework for Misbehavior Detection

peers to protect malicious nodes and bias the global aggregation.

Selective Misbehavior Attack — Malicious node attacks only some honest nodes and displays completely honest behavior towards some other nodes. These attacks are pairwise oriented and are defined as related to the edges in the graph.

For implementing these attacks as realistically as possible, the ZigZag (On/Off) attack is included for both data-layer and feedback attacks. In this situation, a malicious node initially behaves honestly for some time but subsequently injects dishonest behavior in certain interactions.

In our experiments, the Ulm traffic scenario is employed, which is a realistic vehicular traffic scenario for IoV networks. The mobility traces of this scenario are generated using SUMO, while integration with OMNeT++ is performed through the VEINS framework. The corresponding map of Ulm is extracted from OpenStreetMap and depicts an urban environment with high vehicular mobility. In this real-world scenario, vehicle interactions are logged continuously over a 24-hour period and made available within the F2MD framework [8].

Then, 30% of vehicles are randomly selected as attackers; in 70% of cases, they inject data-level attacks, and in the remaining cases, they cause feedback corruption. The simulation of the Ulm scenario runs for 5000 seconds, where more than 550 vehicles interact with each other. The interaction logs are stored and subsequently used to generate the dataset. Figure 6 illustrates the simulation environment, and Table 2 shows the simulation details.

During simulation runs, WSM_Report messages and ground-truth attacker labels are logged per interval to generate the IOV_DS dataset. The generated dataset is then used to train and test the proposed RSU-level deep learning TemporalGATWithLSTM model for global trust prediction and malicious node detection. Figure 7 shows the sample rows of the IOV_DS dataset generated from the simulation, where Table 3 shows the parameters used in this dataset.

Table 2. Simulation Details

Parameter	Value
Network simulator	OMNeT++ 5.0
V2X Traffic Simulator	SUMO 0.25.0
Framework	VEINS 4.4
Malicious Nodes Framework	F2MD
Simulation arena (urban)	6899 M * 5889 M
Simulation Time	5000 Seconds
Event start time	75 Seconds
Number of Vehicles	+550
Percentage of Malicious Nodes	30 %
MAC Protocol	IEEE 802.11p
Radio Propagation Model	Simple Path Loss
Data length	1024 bit
Header Length	256 bit
Initial Trust Score	0.7
11p Specific Parameters (NIC-Settings)	
Tx Power	20 mW
Bit Rate	6 Mbps
MinPowerLevel	-89 dbm
Noise Floor	-98 dbm
Antenna Offset Y	0 Meter
App Layer Header Length	80 bit
Beacon Interval	1 seconds
Frequency Band	5.9 GHz
Max Interference Distance	1000 Meters

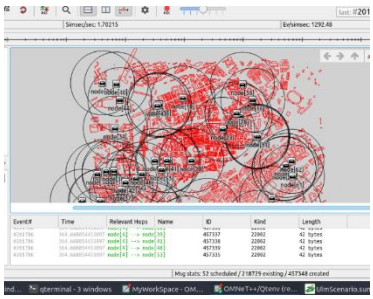

Figure 6. Simulation of the proposed model

Figure 7. sample rows of the IOV_DS generated dataset

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
1	receiver Pseudold	sender Pseudold	sender MbType	Received BSMCount	avgBsm Score	BSMScores	Familiarity	avgPos Similarity	avgSpeed Similarity	avgHeading Similarity	Total Similarity	First Interaction Time	Last Interaction Time	Duration	AvgRSSI	PDR	fTrust	calculated TrustScore	ExitTime	Event Factor
2	10450	10510	Genuine	30	0.96102	59361,0.958(	0.825047	0.727475	0.993648	0.978061	0.899728	0.46835676	29.4683524	29	-79.9415	1	0.87819	0.878194	30.1	0.5
3	10450	10690	Genuine	7	0.96715	68,0.965228,	0.480248	0.0945845	0.98285	0.979395	0.68561	20.7789978	26.7789971	6	-88.8336	1	0.73506	0.735056	30.1	0.5
4	10450	101052	Genuine	28	0.96181	59569,0.9555	0.808523	0.963281	0.9897	0.936503	0.963161	2.69620703	29.6962104	27	-60.9869	1	0.91722	0.917224	30.1	0.5
5	10510	10450	Genuine	30	0.94832	44908,0.9471	0.833507	0.727408	0.990805	0.977918	0.89871	0.17114338	29.171152	29	-79.9433	1	0.87407	0.874074	30.1	0.5
6	10510	10690	Genuine	3	0.97683	4,0.971999,0	0.289371	0.271934	0.99343	0.736359	0.667241	0.77898741	2.77898896	2	-88.8801	1	0.70593	0.705925	30.1	0.5
7	10510	101052	Genuine	28	0.96152	5926,0.9553(	0.816815	0.747682	0.979473	0.920732	0.882629	2.69620736	29.6962107	27	-79.2363	1	0.87335	0.873351	30.1	0.5
8	10690	10450	Genuine	6	0.95939	51029,0.9552	0.658017	0.0950684	0.976909	0.980882	0.684287	21.1711511	26.1711462	5	-88.8295	1	0.74395	0.743948	30.1	0.5
9	10690	10510	Genuine	3	0.97823	4,0.973342,0	0.422806	0.273602	0.995505	0.82018	0.696429	0.46835739	2.46835893	2	-88.8658	1	0.71473	0.714731	30.1	0.5
10	10690	101052	Genuine	10	0.96907	62764,0.962	0.835323	0.126584	0.977338	0.990366	0.698096	17.6962082	26.6962048	9	-88.5867	1	0.80816	0.808163	30.1	0.5
11	101052	10450	Genuine	30	0.94576	945222,0.941	0.807912	0.963331	0.990113	0.964334	0.972592	3.17114306	32.171149	29	-61.0888	1	0.9135	0.9135	32.6	0.5
12	101052	10510	Genuine	30	0.95958	955468,0.954	0.807912	0.748527	0.986656	0.951207	0.895463	3.46835676	32.4683496	29	-79.2093	1	0.87514	0.875138	32.6	0.5
13	101052	10690	Genuine	10	0.96176	51741,0.9501	0.552096	0.12689	0.980769	0.990291	0.699316	17.7789874	26.778997	9.00001	-88.5815	1	0.76313	0.76313	32.6	0.5
14	10330	10390	Genuine	12	0.95053	51082,0.949	1	0.490884	0.981613	0.304753	0.592417	46.6681584	59.6681591	13	-83.3857	0.85714	0.79458	0.794578	60.1	0.5
15	10390	10330	Genuine	12	0.94302	3,0.922503,0	1	0.485528	0.979835	0.309189	0.591517	46.5777742	59.5777759	13	-83.5177	0.85714	0.79107	0.791073	60.1	0.5
16	10450	10510	Genuine	21	0.96315	5,0.958864,0	0.81858	0.789702	0.99265	0.839596	0.873983	30.4683471	59.4683562	29	-76.8444	0.7	0.86105	0.861049	60.1	0.5
17	10450	101052	Genuine	30	0.95508	5247,0.9580	0.844618	0.924573	0.986933	0.718293	0.8766	30.6962052	59.6962079	29	-65.1632	1	0.89183	0.891831	60.1	0.5
18	10510	10450	Genuine	19	0.9458	,0.948411,0	0.832501	0.788289	0.988222	0.846361	0.874291	30.1711468	59.1711431	29	-76.8944	0.63333	0.85287	0.85287	60.1	0.5
19	10510	101052	Genuine	20	0.95926	.9532,0.956(	0.830599	0.747553	0.979124	0.885669	0.870782	30.6962055	58.6962071	28	-78.7537	0.68966	0.85634	0.856342	60.1	0.5
20	101052	10450	Genuine	28	0.93752	946643,0.948	0.857281	0.915853	0.984289	0.687838	0.86266	33.1711437	60.1711443	27	-65.8485	1	0.88205	0.882054	62.6	0.5
21	101052	10510	Genuine	15	0.96409	8,0.956246,0	0.797005	0.744307	0.981826	0.856155	0.860766	33.4683574	58.4683486	25	-79.7914	0.57692	0.84454	0.844536	62.6	0.5

IoV_DS_RSU_0

Table 3. Columns used in Dataset

Columns			
Receiver PseudoId	**Sender PseudoId**	**Sender MbType (Ground Truth)**	**Receiver MbType (Ground Truth)**
Received BsmCount	**Avg BsmScore**	**Familiarity**	**AvgRSSI**
AvgPosSimilarity	**AvgSpeedSimilarity**	**AvgHeadingSimilarity**	**AvgTotalSimilarity**
First Interaction Time	**Last Interaction Time**	**Duration**	**PDR**
fTrust (Ground Truth)	**Calculated TrustScore (Feedback Trust)**	**EventFactor**	**ExitTime**

5.2. Implementing Proposed TemporalGATwithLSTM deep learning model

In this section, we describe the implementation of the proposed deep learning–based trust management model, TemporalGATwithLSTM, and its training on the IoVDS dataset generated from the realistic Ulm simulation. The implementation is carried out in PyTorch. The implementation pipeline consists of five main stages: (i) feature definition, (ii) data preprocessing, (iii) sequence preparation, (iv) model design and training, and (v) model evaluation.

Feature definition: The proposed model uses a set of trust-related features, including AvgBSMScore, Familiarity, AvgPosSimilarity, AvgSpeedSimilarity, AvgHeadingSimilarity, AvgRSSI, PDR, and EventFactor, as input features. Additional features (e.g., SenderPseudoId, ReceiverPseudoId, ReceivedBSMCount, Duration, ExitTime) are also employed for node pairing and temporal modelling... interaction structuring. Labels include fTrust and senderMbType and are used for supervised training and evaluation.

Data Preprocessing: The dataset undergoes several preprocessing steps:

- **Cleaning:** Removing missing values in critical fields (ExitTime, fTrust, AvgRSSI) and replacing invalid numeric values (e.g., negative ReceivedBSMCount) with zero.
- **Feature Engineering:** A combined feature, PDR_RSSI_Combined, is created to represent link quality more comprehensively.
- **Standardization:** Features are standardized using zero mean and unit variance for faster convergence.
- **Class Weighting:** Since trust values are unevenly distributed, target values are divided into ten bins, and inverse-frequency weights are computed. Lower trust bins are further emphasized to improve the detection of malicious nodes.

Sequence Preparation: Since trust changes temporally, the dataset is grouped by sender–receiver pairs and sorted into sequences. Variable-length sequences are padded to a fixed max sequence length, enabling batch processing with the LSTM component.

Models' definition and training:

TemporalGATwithLSTM: The main model integrates temporal learning and graph attention mechanisms. By integrating these mechanisms, the model captures both the temporal dynamics and the structural dependencies of trust in IoV networks.

Architecture: This model consists of 2 main components.

RNN Component (Temporal Module): A bidirectional LSTM processes sequences of interactions, with attention mechanisms emphasizing more recent interactions, longer ones, and those with higher BSM counts within a sequence (LSTM Attention).

Graph Component (Spatial Module): Vehicle interactions are represented as a directed graph, with nodes denoting vehicles and edges denoting trust feedback. Node features include degree and average recency. Two GAT layers with multi-head attention (8 heads) learn structural dependencies, prioritizing feedback (edges) from neighbors with stronger influence, based on recency, degree, and sender trustworthiness. Ensuring that the final node trust score is computed as a weighted aggregation of multiple trust feedbacks (Graph Attention). After each GAT layer, there is a LayerNormalization, ReLU Activation & dropout set to 0.2. The outputs of the Dropout layers preceding each GAT layer are combined using the Residual_weight. Finally, using an iterative method, the trust scores of the sender nodes are predicted using the GraphAttention mechanism.

Training: To train the proposed model, the dataset is split into training and validation sets at an 80-20 node-pair ratio. Then, the model is trained using the class weights computed during preprocessing and a combined weighted loss function comprising Weighted MSELoss (with a hinge term to penalize high trust scores for malicious nodes) for the regression component and BinaryCrossEntropyLoss for the

classification component. These loss functions are balanced with coefficients α = 0.3 and β = 0.7 to prioritize classification over regression. To train the model, the AdamW optimizer with an initial learning rate of 0.0025 is used, and weight_decay=0.0001 is also applied to reduce the risk of overfitting. Additionally, a learning rate planner dynamically adjusts the optimizer's learning rate by a factor of 0.5 based on model performance to improve convergence. An Early Stopping mechanism with patience 20 is also defined, indicating the number of epochs the planner waits for the validation loss to recover. The model is trained for a maximum of 300 Epochs, and in each epoch it produces edge- and node-level trust-score predictions and classification probabilities. This process is repeated, updating the output until stable changes in trust values are achieved. The model with the best checkpoint (i.e., the lowest validation loss) is saved. The model is trained to predict TemporalTrust as a proxy label, defined as the exponentially weighted average of historical trust scores (fTrust) across interactions of a pair. The pseudocode for calculating TemporalTrust is shown below.

Algorithm 2: ComputeTemporalTrustScore

Pseudocode for Computing TemporalTrustScore as ProxyLabel

```
Input: InteractionLevelTrustScores
Output: TemporalTrustScore
 Initialize
Decay_rate = 0.3
finalTrustScore = Interaction Level TrustScore
ExponentialDecayWeights = exp(-decay_rate *
(NumberOfInteractions - 1))
 MessageQualityWeights = Log(BSMCount where
Capped to max 30) / Log(30)
DurationWeights = Log(duration) /
Log(max(duration))
OverallWeights = ExponentialDecayWeight *
MessageQualityWeights * durationWeights
OverallWeigths = OverallWeights /
SUM(OverallWeights)
TemporalTrust = (fTrust * OverallWeights) – for
each Interaction in senderReceiverPairSequence
END
```

5.3. Results and Discussion

In this study, to evaluate the performance of the proposed model to assign trust scores and detect malicious nodes, the criteria defined in the confusion matrix have been used to calculate the True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) parameters. Then, based on these values, the following key evaluation criteria, including Precision, recall, accuracy, F1-score, true positive rate, and true negative rate, have been calculated, which are mainly used to examine the feasibility of the proposed trust model [42]. This section analyzes the performance of the proposed trust management framework through two evaluation levels: (i) vehicle-level (local trust computation) and (ii) RSU-level (global trust aggregation using deep learning). The evaluation focuses on two key aspects of trust management models: detection capability (the ability to distinguish between genuine and malicious nodes) and resilience against trust-related attacks.

5.3.1. Vehicle level evaluation

At the vehicle level, the LTrustAssess algorithm computes local trust scores in real time, based on plausibility & consistency checks over BSM data. A node is classified as malicious if its trust score falls below a predefined threshold. To select an optimal threshold, multiple candidates between 0.50 and 0.70 (step size 0.05) were evaluated, and the classification performance was assessed using a confusion matrix. For this purpose, the algorithm's performance is examined in two different situations: (i) without trust feedback related attacks, such as goodMouthing, BadMouthing. (ii): with 20% malicious trust feedback in the whole network. The results are shown in the following:

Table 4. LTrustAssess without Trust feedback attacks

Algorithm	Threshold	Accuracy	Precision	Recall	F1-Score
LTrustAssess InteractionLevel	**0.65**	**0.9515**	**0.9552**	**0.9811**	**0.9679**
LTrustAssess NodeLevel	**0.65**	**0.9786**	**0.9767**	**0.9952**	**0.9859**

Table 5. LTrustAssess with 30% Trust feedback attacks

Algorithm	Threshold	Accuracy	Precision	Recall	F1-Score
LTrustAssess InteractionLevel	**0.65**	**0.7952**	**0.8984**	**0.8180**	**0.8564**
LTrustAssess NodeLevel	**0.65**	**0.8446**	**0.8778**	**0.9216**	**0.8992**

The results show that the local trust assessment method achieves high accuracy in detecting data-level attacks such as DoS, message modification, and data replay, due to effective plausibility and consistency checks. However, since LTrustAssess operates instantaneously without considering temporal history and changes in node behavior over time, it is less effective against time-varying and trust-related attacks such as ZigZag (On/Off) and good-mouthing and bad-mouthing attacks, where malicious nodes alternate between honest and dishonest behaviors. Therefore, this detection needs to be performed at the RSU level and globally.

5.3.2. RSU level evaluation

At the RSU level, the proposed TemporalGATwithLSTM model aggregates trust over time and across multiple receivers. The model is trained on the IoVDS dataset and evaluated on the test set after applying the same preprocessing steps as in the training phase. In examining global trust scores, we will analyze both the trust scores between pairs (edge trust scores) and the trust scores of each sending entity (node trust scores). Edge trust scores are used to assess the model's ability to detect Selective Misbehavior Attacks, in which malicious nodes send only fake messages to some nodes while behaving normally toward others. Therefore, both the predicted edge trust score and the predicted node trust score are examined in the performance evaluation. The model-predicted trust scores are then used to classify nodes as genuine or malicious, with thresholds between 0.50 and 0.70. As in the local evaluation, a threshold of 0.65 yielded the best results. For this purpose, the model's performance is examined in two different situations: (i) without trust feedback related attacks, such as goodMouthing, BadMouthing. (ii): with 30% malicious trust feedback in the whole network. The results are shown in the following tables.

Table 6. DL Model without Trust feedback attacks

Algorithm	Threshold	Accuracy	Precision	Recall	F1-Score
Deep Learning Edge Level	**0.65**	**0.9683**	**0.9658**	**0.9919**	**0.9786**
Deep Learning Node Level	**0.65**	**0.9857**	**0.9814**	**1.0000**	**0.9906**

Table 7. DL Model with 30% Trust feedback attacks

Algorithm	Threshold	Accuracy	Precision	Recall	F1-Score
Deep Learning Edge Level	**0.65**	**0.9187**	**0.9311**	**0.9600**	**0.9453**
Deep Learning Node Level	**0.65**	**0.9732**	**0.9743**	**0.9905**	**0.9823**

These results shows that the deep learning model in an effective manner detects malicious nodes at the end of each state by leveraging both temporal trust dynamics and graph-based structural dependencies. The attention mechanism further enhances resilience against ballot-stuffing (good-mouthing) and bad-mouthing attacks, as RSUs assign higher weights to consistent feedback from trusted nodes and down-weight anomalous feedback. Furthermore, the model shows strong resistance to ZigZag attacks: even when malicious nodes intermittently behave honestly, their cumulative trust score gradually decreases due to temporal aggregation, enabling detection over time. In addition, the calculated edge trust score between each pair allows the impact of the Selective Misbehavior Attack to be minimized. The trust score between each pair indicates the feedback from each node to other nodes and enables the detection of whether a malicious node has targeted only some nodes in the network.

5.4. Comparison

In this section, the proposed trust management model is compared with existing approaches from two different perspectives, aligned with the contributions of this research. The first comparison focuses on dataset generation, while the second addresses the performance of malicious node detection models.

- Comparison of the IoVDS-generated dataset with existing datasets,
- Comparison of the proposed detection model with baseline methods.

5.4.1. Comparison of the IoVDS Dataset

One of the contributions of this study is the development of IoVDS, a dataset specifically designed for trust management in Internet of Vehicles. To the best of our knowledge, the closest effort toward creating a trust-related dataset for vehicular networks is the work of Wang et al. [16], who introduced the TM-IoV dataset. In their work, the authors emphasized that no public dataset for trust management in IoV was available, and thus their dataset represented an initial contribution to this area. However, their dataset suffers from several limitations that IoVDS addresses:

Temporal dimension: TM-IoV is a static dataset in which trust parameters are only computed once at the end of the simulation, making it unsuitable for analyzing the temporal evolution of trust. In contrast, IoVDS captures trust-related parameters at multiple time intervals, enabling time-series analysis of trust dynamics.

Scope of features: TM-IoV is node-centric, considering only entity-related parameters and ignoring data-level trust factors. IoVDS is hybrid, incorporating both entity-level features (e.g., familiarity, PDR, Similarity) and data-centric features (e.g., avgBSMScore, RSSI).

Feature richness: IoVDS includes additional contextual parameters such as number of exchanged messages, duration of interactions, RSSI values, and event-related factors, which enrich the dataset and allow extraction of more meaningful dependencies.

Attack coverage: TM-IoV only considers the ZigZag (On–Off) attack. IoVDS includes a wider variety of trust-related attacks such as ZigZag, Bad-Mouthing, and Good-Mouthing, which enhances the dataset's realism and robustness.

Table 8. comparison of the proposed dataset with TM-IoV dataset

Category	Proposed Dataset (IoVDS)	Dataset [16] TM-IoV
Dataset Size	Interaction logs between 582 vehicles (over ~17,000 interactions)	Interaction logs between 79 vehicles (~9,700 interactions)
Temporal Dimension	Dynamic (trust-related parameter values for each trustor–trustee pair at different time intervals during simulation) (sequence of interactions)	Static (trust-related parameter values for each trustor–trustee pair at the end of the simulation)
Approach	Hybrid (entity- and data-focused)	Entity-focused
Dataset Structure	Pair-based	Pair-based
Parameters	BSMScores, SpeedSimilarity, HeadingSimilarity, PositionSimilarity, Familiarity, ReceivedBSMCount, Duration, PDR, EventFactor, RSSI, ReportTime, SenderMbType, fTrust, CalculatedTrustScore	PDR, Similarity, Familiarity, Context, Reward/Punishment
Simulator	Veins framework in OMNeT++ and Ulm Scenario (a real-world scenario, highly relevant to vehicular networks)	Java-based IoV simulator (no further details provided)
Implemented trust Attacks	ZigZag attack / Badmouthing attack / Ballot-stuffing attack + some data level attacks and misbehaviors	ZigZag attack
Applied Model	TemporalGAT with LSTM, trained on the generated dataset	No learning model introduced
Availability	Public	Not Public

5.4.2. Comparison of Proposed DL Model

It is important to note that a direct one-to-one comparison of trust management models is often not feasible due to differences in trust metrics, simulation environments, and datasets [26]. Nevertheless, to evaluate the effectiveness of the proposed model, it was compared with existing methods by implementing prominent approaches from similar studies on the generated dataset, using the trust parameters defined in this research. In this process, common evaluation metrics, including True Negative Rate, Accuracy, Recall (True Positive Rate), Precision, and F1-Score, have been utilized to evaluate the detection rate of malicious nodes, enabling a comparative analysis and clarifying the relative standing of the proposed model compared to other methods. The following table compares the proposed deep neural network-based method with similar works by evaluating several common performance metrics defined in those studies. To this end, comparisons have been made with baseline methods, such as Weighted Voting and the Random Forest-based approach presented in [28].

Table 9. comparison of approaches without malicious attacks

Approach	scope	Accuracy	Precision	Recall	F1-Score
Baseline Weighted voting	Edge	84.95	90.33	88.97	89.65
	Node	91.61	91.93	97.39	94.58
Random Forest	Edge	90.88	98.67	88.74	93.44
	Node	96.96	99.27	96.67	97.95
KNN Regressor	Edge	95.13	98.06	95.24	96.63
	Node	97.5	99.04	97.62	98.33
Proposed Model TemporalGATwithLSTM	Edge	97.65	97.55	99.29	98.41
	Node	98.57	98.36	99.76	99.06

Table 10. Comparison of approaches with 30% malicious attacks

Approach	scope	Accuracy	Precision	Recall	F1-Score
Baseline Weighted voting	Edge	78.34	87.26	82.48	84.80
	Node	89.46	90.22	96.44	93.23
Random Forest	Edge	82.50	95.38	79.98	87.01
	Node	92.86	98.97	91.45	95.06
KNN Regressor	Edge	84.57	93.83	84.49	88.92
	Node	93.21	97.99	92.87	95.37
Proposed Model TemporalGATwithLSTM	Edge	91.87	93.11	96	94.53
	Node	97.32	97.43	99.05	98.23

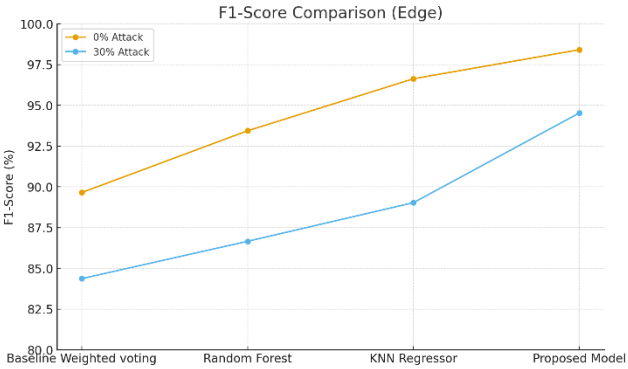

Figure 8. Comparison of Edge Trust in different approaches without and with 30 % malicious trust attacks

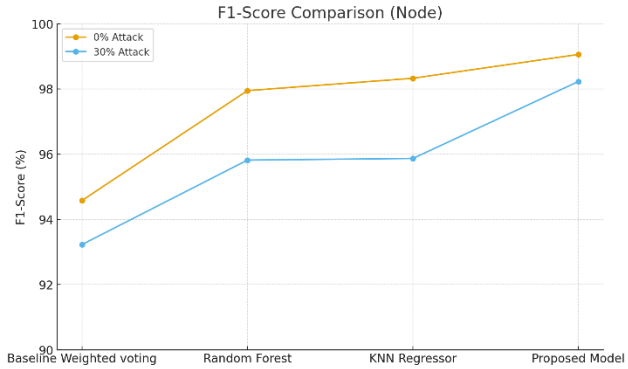

Figure 9. Comparison of Node Trust in different approaches without and with 30 % malicious trust attacks

As shown by the results, when the rate of good- and bad-mouthing attacks in the network is low, the proposed method remains superior to the compared methods; however, the other methods also identify malicious nodes with a high percentage. However, with the increase in the rate of good-mouthing and bad-mouthing attacks in the feedback of malicious nodes regarding the trust score of other entities in the network, our model remains capable of malicious node detection with a high percentage, which indicates the appropriate efficiency of the proposed model in identifying malicious nodes in a high percentage of trust-related attacks compared to other basic and machine learning-based methods.

The primary advantage of the proposed approach over the evaluated methods lies in its use of deep neural networks to compute trust scores and its consideration of dynamic behavioral and temporal patterns. By analyzing the historical performance of nodes over time and accounting for their interactions across the entire network based on a graph-based representation, this model enables a more comprehensive trust evaluation for each node. This feature has led the proposed model to achieve, according to comparative results, a higher detection rate and better performance compared to the evaluated models on the dataset generated from the simulation of a real-world ULM scenario using the Veins framework and the defined trust parameters. Consequently, it emerges as an efficient approach for trust management and the identification of malicious nodes in vehicular networks.

5.5. Complexity Analysis and Execution Time of the Proposed Method

When analyzing the computational complexity and execution time of the proposed method, the following points are noteworthy:

Global Trust Score Computation Window: In the current model simulation, a 5000-second time window is used to compute the global trust score. Given that the simulation scenario is an Urban Local Mobility (ULM) scenario, this duration is appropriate because, in less than two hours of urban mobility, vehicles do not travel far enough to pass through the communication range of many RSUs. Consequently, even in the most extreme case, the number of RSUs that must participate in consensus over the trust score remains limited. This allows the adoption of a lightweight, location-aware consensus protocol for trust aggregation among RSUs. Depending on the network type and density, this duration may be adjusted (increased or decreased).

On-Vehicle Computation: Vehicle-to-vehicle communication is performed using Basic Safety Messages (BSMs), whose format is standardized and readily processable at the vehicle level. Each vehicle performs a simple trust computation every 30 seconds to derive its local trust score. Since this calculation is based on a weighted sum, the computational complexity remains O(1). This indicates that the model is well-suited to optimizing energy consumption in vehicles within IoV environments [43][44].

Deep Learning Component at the RSU Layer: The deep learning component for RSUs was implemented and evaluated in Google Colab. The following execution times were obtained:

Model training: Training the deep learning model for 300 epochs took approximately 16 minutes. Since the model is pre-trained and only needs to be loaded during evaluation, training time does not pose any operational concerns.

Model evaluation: The end-to-end evaluation pipeline—transforming the raw simulation data into final trust outputs—took 45 seconds for data corresponding to a 5000-second simulation interval. This duration includes interaction pairing, sequence construction, trust prediction using an LSTM layer, graph-based representation of interactions, and

computation of link- and node-level trust scores over a 5000-second history. After incorporating the time required for RSU consensus via a lightweight consensus protocol, the global trust score for all nodes can be computed in under 2 minutes. Compared to the 5000-second interval between defined simulation states, these results demonstrate that execution time is well within acceptable limits.

Suitability for RSU Deployment: RSUs typically have fixed locations, larger communication ranges, and higher computational capacity than vehicles. Given the small size and low computational overhead of the proposed deep learning model, it can be easily deployed at the RSU level without operational constraints.

6. Discussion and Conclusion

In this research, we introduced a trust management model for detecting malicious nodes in the Internet of Vehicles. The proposed model adopts a two-layer architecture operating at both the vehicle and RSU levels and combines data-centric and node-centric approaches to detect malicious behaviors at both the data and node levels. This design enables a decentralized and efficient trust management mechanism for vehicular networks.

The proposed model was implemented through simulation and evaluated using standard performance metrics. The evaluation results, along with comparisons with baseline methods, demonstrate that the proposed approach effectively manages trust by assigning dynamic trust scores to vehicles. Moreover, it achieves higher detection rates for malicious nodes and demonstrates greater resilience to trust-related attacks than baseline models. The advantages of this work can be summarized as follows:

- To address the lack of publicly available trust datasets, we generated a dataset (IoVDS) through the simulation of a real-world urban scenario, which can serve as a foundation for future studies.
- By leveraging graph-based feedback aggregation and an attention mechanism, the model enables dynamic weighting of feedback from neighboring nodes, overcoming limitations in traditional trust aggregation schemes.
- The integration of LSTM enables the model to incorporate temporal dynamics, aligning with the principle that trust is developed over time.
- To tackle the scalability challenge, the model relies on a decentralized consensus mechanism among RSUs, ensuring distributed trust management across the network.

These characteristics position the proposed framework as a promising candidate for deployment in ITS, enhancing both the security and performance of vehicular networks.

6.1. Open Research Directions

While the proposed model shows strong performance, several open research directions remain for future investigation:

Weighted RSU feedback – The current model weights node feedback dynamically; it could be extended to weight RSU contributions based on their historical reliability during the consensus process.

Consensus mechanism optimization: Future work should investigate lightweight, scalable consensus mechanisms for RSUs, potentially employing localized or cluster-based consensus to reduce latency and better adapt to vehicle mobility.

Blockchain integration – The final trust scores, once agreed upon by RSUs, could be stored on a blockchain, ensuring immutability, transparency, and availability of trust information across all RSUs.

Data retention policies: Currently, only the most recent global trust scores are retained for each state, whereas previous interaction data is discarded. Optimizing data storage and retention strategies could improve long-term trust assessment.

Privacy Consideration – Real-world vehicle networks use different pseudo-IDs to maintain privacy in various interactions, and their real ID is only registered in the network. In the current study, vehicles also use pseudo-IDs to exchange information among themselves; however, these pseudo-IDs are defined as fixed and immutable. Future research in this area can address the modification of pseudo-IDs across vehicles' interactions with other nodes to maintain privacy.

Event Confidence evaluation – A promising research direction is the application of calculated trust scores to event message validation. When a vehicle broadcasts an event (e.g., an accident or hazard warning), the trust scores of both the sender and relay nodes could be incorporated, along with factors such as hop count and message redundancy, to assess event credibility.

By addressing these challenges, future research can further strengthen the robustness, scalability, and applicability of trust management systems in IoV, ultimately contributing to the development of secure, reliable, and intelligent transportation networks.

7. Author Contributions

This Manuscript is extracted from the master's thesis of the corresponding author (Ali Moradi) with the supervision of the proofreader. Dr Nasser Yazdani.

8. Funding

This research received no external funding.

9. Data Availability Statement

The IoVDS dataset presented within this manuscript is publicly available here.

10. AI-Assisted Technology Declaration

AI-assisted tools were used in this study solely to generate and preliminarily visualize certain figures. These tools were employed to enhance graphical clarity. The authors reviewed, edited, and validated all generated images, and take full responsibility for the accuracy, integrity, and originality of the content. No AI-assisted tools were used for data analysis, result interpretation, or scientific decision-making.

11. Conflicts of Interest

The authors declare no conflicts of interest.

12. References

[1] Najafi, M., Khoukhi, L., & Lemercier, M. (2022). Decentralized prediction and reputation approach in vehicular networks. Transactions on Emerging Telecommunications Technologies, *33*(7), e4456. DOI:10.1002/ett.4456

[2] Siddiqui, S. A., Mahmood, A., Sheng, Q. Z., Zhang, W. E., & Aljubairy, A. (2021). A survey of trust management in the internet of vehicles. Electronics, *10*(18), 2223. DOI:10.3390/electronics10182223

[3] Mahmood, A., Ullah, F., Sheng, Q. Z., Siddiqui, S. A., & Aljubairy, A. (2021). When trust meets the internet of vehicles: Opportunities, challenges, and future prospects. In 2021 IEEE 7th International Conference on Collaboration and Internet Computing (CIC) (pp. 60–69). IEEE.

[4] Hbaieb, A., Ayed, S., & Chaari, L. (2022). A survey of trust management in the Internet of Vehicles. Computer Networks, *203*, 108558. DOI:10.1016/j.comnet.2021.108558

[5] Alalwany, E., & Mahgoub, I. (2024). Security and Trust Management in the Internet of Vehicles (IoV): Challenges and Machine Learning Solutions. Sensors, *24*(2), 368. DOI:10.3390/s24020368

[6] Rehman, A., Paul, A., Din, S., & Jeon, G. (2020). State-of-the-art IoV trust management a meta-synthesis systematic literature review (SLR). PeerJ Computer Science, *6*, e334. DOI:10.7717/peerj-cs.334

[7] El-Sayed, H., Chaqfeh, M., & Lakas, A. (2019). Trust enforcement in vehicular networks: challenges and opportunities. IET Wireless Sensor Systems, *9*(5), 237-246. DOI:10.1049/iet-wss.2018.5211

[8] Kamel, J., Ansari, M. R., Petit, J., Kaiser, A., Jemaa, I. B., & Urien, P. (2020). Simulation framework for misbehavior detection in vehicular networks. IEEE Transactions on Vehicular Technology, *69*(6), 6631-6643. DOI:10.1109/TVT.2020.2984203

[9] Ababsa, M., Bounabat, B., & Maachaoui, M. (2025). Deep Multimodal Learning for Real-Time DDoS Attacks Detection in Internet of Vehicles. arXiv preprint arXiv:2501.15252. DOI:10.48550/arXiv.2501.15252

[10] Siddiqui, S. A., Mahmood, A., Zhang, W. E., & Sheng, Q. Z. (2023). Trust in vehicles: toward context-aware trust and attack resistance for the internet of vehicles. IEEE Transactions on Intelligent Transportation Systems, *24*(9), 9546–9560. DOI:10.1109/TITS.2022.3173629

[11] Rehman, A., Paul, A., & Ahmad, A. (2022). CTMF: Context-aware trust management framework for internet of vehicles. IEEE Access, *10*, 73685–73701. DOI:10.1109/ACCESS.2022.3190513

[12] Elsayed, M. A., & Zincir-Heywood, N. (2022). BoostGuard: interpretable misbehavior detection in vehicular communication networks. In *NOMS 2022-2022 IEEE/IFIP Network Operations and Management Symposium* (pp. 1–7). IEEE.

[13] Guleng, S., Wu, C., Chen, X., Wang, X., & Yoshinaga, T. (2019). Decentralized trust evaluation in vehicular Internet of Things. IEEE Access, *7*, 15980–15988. DOI:10.1109/ACCESS.2019.2893268

[14] Najib, W., & Sulistyo, S. (2019). Survey on trust calculation methods in Internet of Things. Procedia Computer Science, *161*, 1300–1307. DOI:10.1016/j.procs.2019.11.245

[15] Xu, Q., Zhang, L., & Liu, Y. (2025). Enhancing Trust Management System for Connected Autonomous Vehicles Using Machine Learning Methods: A Survey. arXiv preprint arXiv:2505.07882.

[16] Wang, Y., Mahmood, A., Sabri, M. F. M., & Zen, H. (2024). TM–IoV: A First-of-Its-Kind Multilabeled Trust Parameter Dataset for Evaluating Trust in the Internet of Vehicles. Data, *9*(9), 103. DOI:10.3390/data9090103

[17] Wang, J., Jing, X., Yan, Z., Fu, Y., Pedrycz, W., & Yang, L. T. (2020). A survey on trust evaluation based on machine learning. ACM Computing Surveys (CSUR), *53*(5), 1–36. DOI:10.1145/3408292

[18] Wang, Y., Mahmood, A., Sabri, M. F. M., & Zen, H. (2022). Towards Strengthening the Resilience of IoV Networks—A Trust Management Perspective. Future Internet, *14*(7), 202. DOI:10.3390/fi14070202

[19] Xiao, Y., & Liu, Y. (2019). BayesTrust and VehicleRank: Constructing an implicit Web of trust in VANET. IEEE Transactions on Vehicular Technology, *68*(3), 2850–2864. DOI:10.1109/TVT.2019.2892141

[20] Zhang, J., Zheng, K., Zhang, D., & Yan, B. (2020). AATMS: An anti-attack trust management scheme in VANET. IEEE Access, *8*, 21077–21090. DOI:10.1109/ACCESS.2020.2968469

[21] Pu, C. (2021). A novel blockchain-based trust management scheme for vehicular networks. In 2021 Wireless Telecommunications Symposium (WTS) (pp. 1–7). IEEE.

[22] Zhang, C., Zhu, L., Xu, C., Lu, R., & Zhang, X. (2020). AIT: An AI-enabled trust management system for vehicular networks using blockchain technology. IEEE Internet of Things Journal, *8*(5), 3157–3169. DOI:10.1109/JIOT.2020.3008041

[23] Wang, S., Hu, Y., & Qi, G. (2022). Blockchain and deep learning based trust management for Internet of Vehicles. Simulation Modelling Practice and Theory, *120*, 102627. DOI:10.1016/j.simpat.2022.102627

[24] Cheong, C., Wang, Y., Lee, V. C. S., & Zhang, M. (2024). A Path-Backtracking-Based Trust Management Scheme for VANETs. In *2024 IEEE 99th Vehicular Technology Conference (VTC2024-Spring)* (pp. 1–6). IEEE.

[25] Eziama, E., Tepe, K., Balador, A., Nwizege, K. S., & Jaimes, L. M. (2018). Malicious node detection in vehicular ad-hoc network using machine learning and deep learning. In 2018 IEEE Globecom Workshops (GC Wkshps) (pp. 1–6). IEEE.

[26] El-Sayed, H., Chaqfeh, M., & Lakas, A. (2020). Machine learning based trust management framework for vehicular networks. Vehicular Communications, *25*, 100256. DOI:10.1016/j.vehcom.2020.100256

[27] Siddiqui, S. A., Mahmood, A., Zhang, W. E., & Sheng, Q. Z. (2023). Towards a Machine Learning Driven Trust Management Heuristic for the Internet of Vehicles. Sensors, *23*(4), 2325. DOI:10.3390/s23042325

[28] Wang, Y., Mahmood, A., Sabri, M. F. M., & Zen, H. (2024). Mesmeric: machine learning-based trust management mechanism for the internet of vehicles. Sensors, *24*(3), 863. DOI:10.3390/s24030863

[29] Khan, S., Khan, S., Sulaiman, A., Al Reshan, M. S., Alshahrani, H., & Shaikh, A. (2024). Deep neural network and trust management approach to secure smart transportation data in sustainable smart cities. ICT Express, *10*(5), 1059–1065. DOI:10.1016/j.icte.2024.08.001

[30] Kushardianto, N. C., Rahim, R., & Ramli, K. (2024). Vehicular network anomaly detection based on 2-step deep learning framework. Vehicular Communications, *49*, 100802. DOI:10.1016/j.vehcom.2024.100802

[31] Jiang, N., Liu, X., Zheng, H., Wang, D., & Zhou, L. (2022). Gatrust: A multi-aspect graph attention network model for trust assessment in osns. IEEE Transactions on Knowledge and Data Engineering, *35*(6), 5865–5878. DOI:10.1109/TKDE.2022.3150246

[32] Wang, J., Zhang, G., Xu, K., & Zhong, C. (2024). TrustGuard: GNN-based robust and explainable trust evaluation with dynamicity support. IEEE Transactions on Dependable and Secure Computing, *21*(5), 4433–4450. DOI:10.1109/TDSC.2023.3338232

[33] Akintan, F. A., Zhang, Y., Georgalas, N., & Merabti, M. (2025). Graph Neural Networks for Malicious Node Detection in Dynamic Vehicular Networks. In Proceedings of the 2025 ACM International Conference on Future Networks and Distributed Systems (pp. 1–10).

[34] Ou, X., Mi, B., & Zou, H. (2025). Trust Evaluation of Intelligent Connected Vehicles Based on GNN: Multi-feature Fusion and Temporal Modeling. In 2025 IEEE 14th Data Driven Control and Learning Systems (DDCLS) (pp. 1–6). IEEE.

[35] Luo, T., Pan, S., & Wang, H. (2025). Graph neural networks for trust evaluation: Criteria, state-of-the-art, and future directions. IEEE Network, *39*(1), 207–215. DOI:10.1109/MNET.2024.3421578

[36] Yang, L., & Li, H. (2019). Vehicle-to-vehicle communication based on a peer-to-peer network with graph theory and consensus algorithm. IET Intelligent

Transport Systems, *13*(2), 280–285. DOI:10.1049/iet-its.2018.5014

[37] Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., & Yu, P. S. (2020). A comprehensive survey on graph neural networks. IEEE transactions on neural networks and learning systems, *32*(1), 4–24. DOI:10.1109/TNNLS.2020.2978386

[38] Zhang, S., Tong, H., Xu, J., & Maciejewski, R. (2019). Graph convolutional networks: a comprehensive review. Computational Social Networks, *6*(1), 1–23. DOI:10.1186/s40649-019-0069-y

[39] Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., & Bengio, Y. (2017). Graph attention networks. stat, *1050*(20), 10-48550.

[40] Sommer, C., Joerer, S., & Dressler, F. (2019). Veins: The open source vehicular network simulation framework. In Recent advances in network simulation: the OMNeT++ environment and its ecosystem (pp. 215–252). Springer International Publishing. DOI:10.1007/978-3-030-12842-5_6

[41] Ruan, W., Zhang, X., Liang, R., Li, B., & Xiong, N. N. (2023). A double-layer blockchain based trust management model for secure internet of vehicles. Sensors, *23*(10), 4699. DOI:10.3390/s23104699

[42] Gyawali, S., Qian, Y., & Hu, R. Q. (2020). Machine learning and reputation based misbehavior detection in vehicular communication networks. IEEE Transactions on Vehicular Technology, *69*(8), 8871–8885. DOI:10.1109/TVT.2020.2999632

[43] Firouzjah, K. G., & Ghasemi, J. (2025). An Efficient Computational Method for Network Analysis Using Clustering of EV Charging Pattern in Parking. Sustainable Energy, Grids and Networks, *40*, 101945. DOI:10.1016/j.segan.2025.101945

[44] Roudbari, N., Firouzjah, K. G., & Ghasemi, J. (2025). Scenario-based sizing and siting of battery swapping stations for electric buses using realistic demand modeling on distribution network. Energy, *311*, 139378. DOI:10.1016/j.energy.2025.139378