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

Ali Moradi[1,*] - Nasser Yazdani[2]
[1] College of Engineering, School of Electrical & Computer engineering, university of Tehran, Tehran, Iran.
[2] College of Engineering, School of Electrical & Computer engineering, university of Tehran, Tehran, Iran.

ABSTRACT

With the rapid expansion of the Internet of Vehicles, ensuring security and trust among nodes has emerged as one of the fundamental challenges in this domain. The open, dynamic, and distributed nature of these networks creates a suitable environment for malicious nodes, which can compromise communication integrity and overall system security by disseminating false or misleading information. This research aims to present a hybrid and 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 trust scores distribution and detecting malicious nodes compared to 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.

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

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 each other and with roadside infrastructures [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 false information, disrupt communication, and jeopardize road safety, potentially leading to severe consequences such as 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 among vehicles and infrastructures like RoadSideUnits (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 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]. Fig. 1, shows different types of communications in Internet of Vehicles.

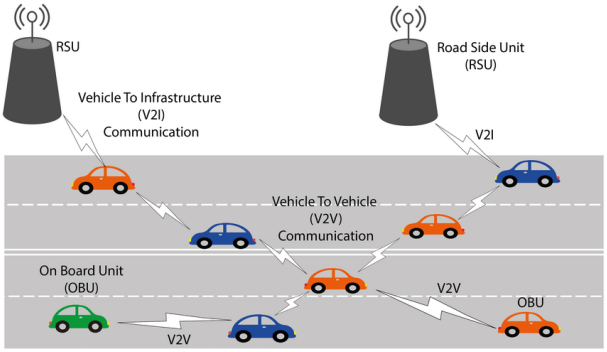

Fig. 1. Types of IoV Communications [1]

Traditional security mechanisms such as encryption and authentication are effective against external threats but insufficient in 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 only identify misbehavior at the data level, trust management provides a more comprehensive framework by considering long-term behavioral patterns and the collective feedback of multiple entities [8]. However, designing robust and efficient trust management systems for IoV remains a significant research open direction, as existing solutions often suffer from lack of comprehensive trust attributes, limited adaptability to dynamic environments, or vulnerability 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 nodes, the proposed 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:

1) 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.

2) A local trust evaluation algorithm (LTrustAssess) is introduced for computing trust scores of vehicles during simulation, enabling local detection of misbehavior & fake relayed events.

3) 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.

4) A deep learning 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 Internet of Vehicles & trust management in the IoV context. Section 3 presents the proposed decentralized AI-enabled trust management framework, including the local trust evaluation algorithm and the global trust prediction model. Section 4 discusses the experimental setup, dataset generation, and evaluation metrics, followed by the analysis of results & comparison with other works. Finally, Section 5 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 related entities through wireless technologies. 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]. Internet of Vehicles allow 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[1](IEEE 1609.x) framework, which enable real-time communication in dynamic vehicular environments, and DSRC[2] 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 a large number of 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, network structure and neighboring nodes change rapidly.*
- ***Short-lived links:** Connections between nodes are temporary, often disrupted by high-speed movements.*
- ***Unreliable wireless channels:** Communication is affected by road conditions, relative speed and direction, vehicle types, and environmental obstacles.*
- ***Resource constraints:** Vehicles have limited capacity for storing 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 massive communication volume 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 means establishing communication in the network only between trusted nodes. In these models, nodes are usually assigned a trust score, and nodes are considered malicious if their trust score is less than a predefined threshold. Trust management provides a framework to evaluate the reliability of nodes in a network, ensuring secure and dependable interactions. It continuously assesses node

[1] *Wireless Access in Vehicular Environments*
[2] *Dedicated Short Range Communication*

behavior, reputation, and adherence to protocols using trust scores, focusing especially on mitigating internal threats from malicious authenticated nodes [1][2][3][4][5]. Therefore, the responsibility of the trust management system is to manage the real-time and long-term trust of nodes in a network based on the legitimacy of the message received from different nodes in the network or the legitimacy of the nodes themselves in the network.

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 involve two main entities: trustor (evaluating node) and trustee (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 try to manipulate trust relationships or provide deceptive information to compromise the system; therefore, establishing a trust management framework to counter these trust-related attacks is essential to strengthen the security posture of IoV [2][3][4][5].

2.2.1 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 accuracy of received messages and the overall behavior of the sender node 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]. Fig. 2, shows different types of Trust Management Models.*

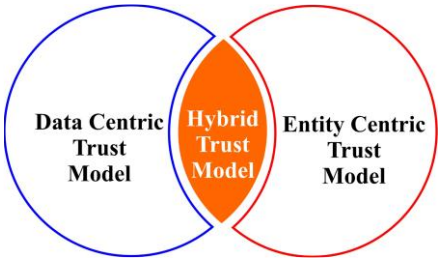

Fig. 1. Trust Management Models [6]

2.2.2 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-point-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: *Algorithms used for trust computation 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 and dynamic trust scores, providing higher accuracy and adaptability compared to traditional methods [5][15][17].*

2.2.3 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. PROPOSED MODEL

In this section we introduce the proposed methodology for trust management in Internet of Vehicles. Connected vehicular networks are inherently dynamic and decentralized, posing significant challenges for trust and security [19]. This work presents a hybrid, decentralized trust management framework that integrates both data-driven (message content-based) and node-centric (behavior-based) evaluations to manage trust at 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 employes an feedback- and consensus-based two-tier approach in order to address 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 proposed LTrustAssess Algorithm. The global layer, implemented on RSUs, aggregates trust related reports from vehicles and determines global trust scores using proposed deep learning model named TemporalGATwithLSTM 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 evidences are transmitted to RSUs which are in vehicles 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.

3.1 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 (analyzing received messages) and node-centric evaluation (assessing sender behavior).

In this model, Vehicles broadcast Basic Safety Messages every second in their communication range. The assumption made in this model is that vehicles are authenticated. That is, all vehicles are pre-registered in the network and enter the network 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 received Basic Safety Message undergoes a set of basic checks on the received message 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 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 using Geometric Mean of the check values, BsmScore will be calculated reflecting message trustworthiness. The Plausibility and consistency checks performed on each received message are listed at Table 1.

***Table 1**. list of Plausibility & Consistency checks performed 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 calculates sender-related trust features, including position similarity, speed similarity, heading similarity, familiarity (interaction frequency and duration of interactions), 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 assigned by the receiver to the sender. Vehicles subsequently use this local trust score to evaluate the reliability of event-driven messages (WSMs) received from other nodes.

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
        TrustScore_θ = 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 following:

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 or 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.*

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

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

$$LocalTrustScore = (LocalTrustScore * Data\_Weight) + (TrustScore_0 * (1 - Data\_Weight))$$

3.2 Interaction Reporting to RSUs

Due to the highly dynamic nature of vehicular networks, 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 containing report of 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, 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 each 30 Second*

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)*
Familiarity	← *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)*

3.3 Global Trust Assessment

Global trust evaluation is performed by RSUs, which receive WSM_Report messages from multiple vehicles within their communication range every 30 seconds. 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 consider temporal variations 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 [21]. 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 receiver perspective about sender. RSUs first predict trust scores for each sender-receiver pair over time based on historical interactions and behavioral changes to compute 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) [21][22][23][24][25]. The global trust score prediction will be done as follows.

3.3.1 Predicting Global Trust Scores

The prediction of Aggregated global trust scores is performed by RSUs using a pre-trained deep learning model, trained on generated dataset from simulation representing vehicle interactions. To effectively model the network as a feedback graph, we employ Graph Attention Networks (GAT) [26] to capture the heterogeneous significance of trust feedback from different 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 [22][24]. 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 [26].

For this, RSUs first transform received interaction reports into sender-receiver pairs. The model then considers temporal sequences of interactions between each pair by considering all previous interactions at previous time intervals from the point of view of different receivers based on the history of the sender interactions, predicting 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. For this purpose, to incorporate both temporal dynamics and network structure, we propose a TemporalGATWithLSTM model, which combines graph attention with LSTM layers to capture both 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

3.3.2 TemporalGATwithLSTM

The proposed TemporalGATWithLSTM model is a hybrid deep learning architecture that combines Graph Attention Networks (GAT) and Long Short-Term Memory (LSTM) networks to predict the aggregated global trust scores of vehicles in highly dynamic vehicular networks. This model is capable of capturing both the graph-structured interactions between 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, PDR_RSSI_Combined over time, generates 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), with more duration time and more recent occurrences are weighted higher. ensuring that critical interactions have greater influence on edge-level trust scores. These edge level trust scores are called Receiver Feedback on the sender node trust in this work. These values are then used in the GAT layer for node-level aggregation.*

***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 feedbacks from different receivers don't have the same weights for the sender, this layer also uses 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 reflects 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.*

Since edge weights depend on the trustworthiness of receiver nodes, which may themselves act as senders in other edges, an iterative procedure is need 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 called RSU feedback on the sender node trust. Figure3 shows the structure of the proposed TemporalGATwithLSTM model.

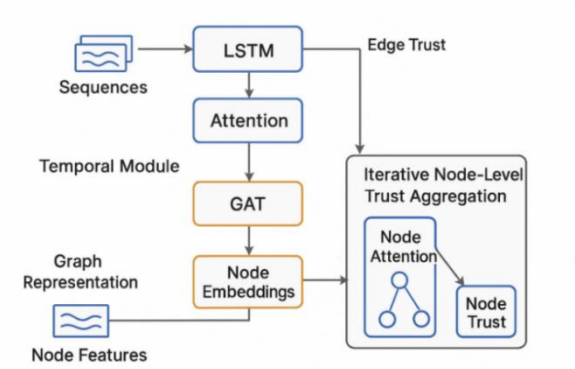

Fig. 3. Structure of the proposed TemporalGATwithLSTM

3.3.3 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.

3.3.4 State Update and Malicious Node Detection

Once the aggregated global trust score for each sender node is stablished 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*
- $\boldsymbol{cT_N} = \textit{Calculated Global Trust at state N}$
- $\boldsymbol{fT_{N-1}}$ *= 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 detection of malicious nodes based on the aggregated global trust scores. After predicting trust values using TemporalGATWithLSTM, a threshold-based

mechanism is applied. Various thresholds between 0.4 and 0.7 (increment 0.05) were evaluated, and the optimal threshold is selected to maximize detection accuracy. A node is classified as malicious if its global trust score falls below the threshold and genuine otherwise:

$$\text{Node Label}(i) = \begin{cases} \text{Malicious}, & \text{if } Trust(i) < \text{threshold } (0.65) \\ \text{Genuine}, & \text{if } Trust(i) \geq \text{threshold } (0.65) \end{cases}$$

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

4. *PERFORMANCE EVALUATION*

In order to evaluate the performance of the proposed model, extensive simulation has been done to assess the LTrustAssess algorithm and generate the dataset needed for evaluation of TemporalGATwithLSTM model. So 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 dataset needed for further analysis, and (ii) implementation of the deep learning trust model for global trust prediction and malicious node detection.

4.1 Vehicular Network Simulation

Due to the complexity and cost of real-world deployment of 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 [27], which combines OMNeT++ (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 vehicles.

For V2I WSM_Report messages, Four RSUs are deployed along roads in the simulation map. their coordinates are chosen to provide uniform coverage so that the entire simulation area falls under RSU coverage. Each RSU receives 30-second trust reports from vehicles currently inside its communication range and use 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 [27][28]. 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.*

[3] *VEhicles In Network Simulation*
[4] *Simulator of Urban MObility*
[5] *Framework for Misbehavior Detection*

- ***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 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 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. Concretely, the malicious reporter reduces computed sender node trust score probabilistically 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 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 realistic as possible, The ZigZag (On/Off) attack is included for both data-layer and feedback attacks. In this situation, a malicious node behaves honestly for some times but injects dishonest behavior in some 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, representing an urban environment with high vehicular mobility. In this real-world scenario, the interactions between vehicles are logged for a continuous 24-hour time period and are made available in the F2MD framework [8]. Then 30% of vehicles are randomly selected as attackers when these attackers in 70% of times inject data-level attacks and in different percentage of times make 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 later used for dataset generation. Figure 4 illustrates the simulation environment where also table 2 shows the simulation details.

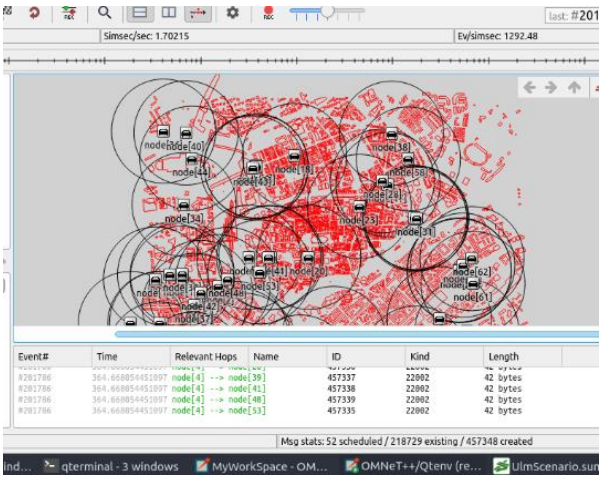

Fig. 4. simulation of the proposed model

***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

During simulation runs WSM_Report messages and ground-truth attacker labels are logged per interval to generate IOV_DS dataset. The generated dataset is then used for training and testing the proposed RSU-level deep learning TemporalGATWithLSTM model for global trust prediction and malicious node detection . figure 5 shows the sample rows of the IOV_DS dataset generated from simulation where table 3 shows the parameters used in this 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	34908,0.9471	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	962764,0.962	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	51741,0.950(	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	0.96408	8,0.956246,0	0.797005	0.744207	0.981826	0.956165	0.860766	33.4683574	59.4683496	25	-79.7814	0.57692	0.84454	0.844536	62.6	0.5

IoV_DS_RSU_0

Fig. 5. sample rows of the IOV_DS generated dataset

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**

4.2 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, 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 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 model integrates temporal learning and graph attention mechanisms. The combination of these mechanisms allows the model to capture both temporal dynamics and 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, more longer ones and those with higher BSM counts withing a sequence (LSTM Attention).*
  - *Graph Component (Spatial Module): Vehicle interactions are represented as a directed graph, where nodes represent vehicles and edges represent trust feedback. Node features include degree and average recency. Two GAT layers with multi-head attention (8 heads) learn structural dependencies, Prioritizes feedbacks (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 an LayerNormalization, ReLU Activation & dropout set to 0.2. The outputs from the Dropout layers leading from each GAT layer are combined with each other by considering the Residual_weight. Finally, using an iterative method, the trust score of the sender nodes is predicted by considering the GraphAttention mechanism.*

Training: To train the proposed model, the dataset is divided into two parts, training and validation, with a percentage of 80-20 based on node pairs. Then, the model is trained using the class weights calculated in the preprocessing process and a combined weighted loss function including Weighted MSELoss (with hinge loss to penalize high trust scores for malicious nodes) for the regression part and BinaryCrossEntropyLoss for the classification part. These loss functions are balanced with coefficients alpha=0.3 and beta=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. Also, a learning rate planner is used, which dynamically adjusts the optimizer learning rate by a factor of 0.5 based on the model performance and is used to improve convergence. An Early Stopping mechanism with patience 20 is also defined, which indicates the number of Epochs that 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 and updates the output until stable changes in trust values are achieved. Model with the best checkpoint (lowest val 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
```

4.3 Results & 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, 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. 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 (ability to distinguish between genuine and malicious nodes) and resilience against trust-related attacks.

4.3.1 Vehicle Level Evaluation

At the vehicle level, the LTrustAssess algorithm computes local trust scores in real time, based on plausibility and 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 ranging from 0.50 to 0.70 (step size 0.05) were tested, and the classification outcomes were validated using a confusion matrix. For this purpose, algorithm's performance is examined in two different situations: (i): without trust feedback related attacks such as goodMouthing, BadMouthing. (ii): with 20% malicious trust feedbacks in whole network. The results are shown in 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 indicate 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.

4.3.2 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 test set after applying the same preprocessing steps as in the training phase. In examining the global trust scores, we will examine both the trust scores between each pair (edge trust score) and the trust scores of each sending entity (node trust score). Edge trust scores are used to examine the model's ability to detect Selective Misbehavior Attacks, in which malicious nodes only send fake messages to some nodes and behave normally to other nodes. Therefore, both the predicted edge trust score and the predicted node trust score are examined in the performance evaluation. The trust scores predicted by the model are then used for classification of nodes as genuine or malicious using thresholds between 0.50 and 0.70. Similar to the local evaluation, threshold = 0.65 yielded the best results, for this purpose, model's performance is examined in two different situations: (i): without trust feedback related attacks such as goodMouthing, BadMouthing. (ii): with 30% malicious trust feedbacks in whole network. The results are shown in following tables and figures.

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 demonstrate that the deep learning model effectively identifies 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 Selective Misbehavior Attack to be minimized. The trust score between each pair indicates the feedback of each node towards other nodes and allows detection if a malicious node has targeted only some nodes in the network.

4.4 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.

4.4.1 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

4.4.2 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 [19]. Nevertheless, to assess the performance of the proposed model, it has been compared with existing methods by implementing and executing 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 employed to assess 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 machine learning-based approach presented in [20].

***Table 9**. comparison of the approaches without and with 30 % of 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

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

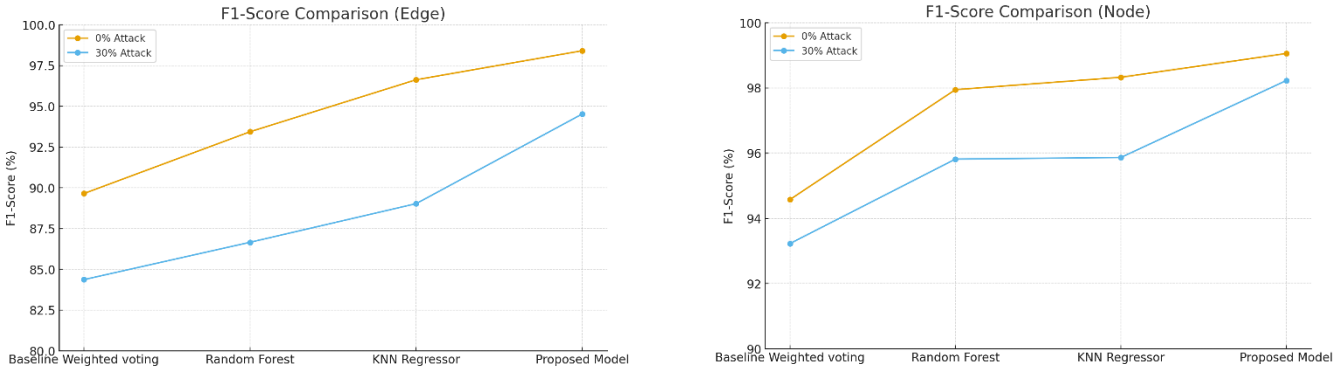

Fig. 6. comparison of approaches without and with 30 % malicious trust attacks

As is clear from the results, in a situation where the rate of good-mouthing and bad-mouthing attacks in the network is low, despite the relative superiority of the proposed method over other compared methods, other methods have also identified malicious nodes in the network with a good 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 nodes in the network, the proposed model still has the ability to identify malicious nodes with a high percentage, which indicates the appropriate performance 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 compared to the evaluated methods lies in its use of deep neural networks to calculate 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. *CONCLUSION & FUTURE WORK*

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 both at the vehicle level and the RSU level, 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 results of the evaluation, along with comparisons against baseline methods, demonstrate that the proposed approach effectively manages trust by assigning dynamic trust scores to vehicles. Moreover, it achieves higher detection rates of malicious nodes and shows strong resilience against trust-related attacks compared to 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 recurrent neural networks (LSTM) allows 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 intelligent transportation systems, enhancing both the security and efficiency of vehicular networks.

5.1 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, but it could be extended to weight the contributions of RSUs themselves, based on their historical reliability, during the consensus process.

Consensus mechanism optimization – Future work should investigate lightweight, scalable consensus mechanisms for RSUs, potentially using 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 after each state, while previous interaction data is discarded. Optimizing data storage and retention strategies could improve long-term trust assessment.

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 communicate with each other; however, this pseudo-ID is defined as fixed and unchangeable. Future research in this area can address changing the pseudo-IDs of different vehicles in their various interactions with other nodes with the aim of maintaining 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., accident or hazard warning), the trust scores of both the sender and relay nodes could be incorporated alongside 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.

***Data Availability Statement:** The IoVDS dataset presented within this manuscript-at-hand will be publicly available at GitHub.*

6. *REFERENCES*

[1] *Najafi, Maryam, Lyes Khoukhi, and Marc Lemercier. "Decentralized prediction and reputation approach in vehicular networks." Transactions on Emerging Telecommunications Technologies 33.7 (2022): e4456.*

[2] *Siddiqui, Sarah Ali, et al. "A survey of trust management in the internet of vehicles." Electronics 10.18 (2021): 2223.*

[3] *Mahmood, Adnan, et al. "When trust meets the internet of vehicles: Opportunities, challenges, and*

future prospects." 2021 IEEE 7th International Conference on Collaboration and Internet Computing (CIC). IEEE, 2021.

[4] *Hbaieb, Amal, Samiha Ayed, and Lamia Chaari. "A survey of trust management in the Internet of Vehicles." Computer Networks 203 (2022): 108558.*

[5] *Alalwany, Easa, and Imad Mahgoub. "Security and Trust Management in the Internet of Vehicles (IoV): Challenges and Machine Learning Solutions." Sensors 24.2 (2024): 368.*

[6] *Rehman, Abdul, et al. "State-of-the-art IoV trust management a meta-synthesis systematic literature review (SLR)." PeerJ Computer Science 6 (2020): e334.*

[7] *El-Sayed, Hesham, et al. "Trust enforcement in vehicular networks: challenges and opportunities." IET Wireless Sensor Systems 9.5 (2019): 237-246.*

[8] *Kamel, Joseph, et al. "Simulation framework for misbehavior detection in vehicular networks." IEEE transactions on vehicular technology 69.6 (2020): 6631-6643.*

[9] *Ababsa, Mohamed, et al. "Deep Multimodal Learning for Real-Time DDoS Attacks Detection in Internet of Vehicles." arXiv preprint arXiv:2501.15252 (2025).*

[10] *Siddiqui, Sarah Ali, et al. "Trust in vehicles: toward context-aware trust and attack resistance for the internet of vehicles." IEEE Transactions on Intelligent Transportation Systems 24.9 (2023): 9546*

[11] *Rehman, Abdul, et al. "CTMF: Context-aware trust management framework for internet of vehicles." IEEE Access 10 (2022): 73685-73701.*

[12] *Elsayed, Marwa A., and Nur Zincir-Heywood. "BoostGuard: interpretable misbehavior detection in vehicular communication networks." NOMS 2022-2022 IEEE/IFIP Network Operations and Management Symposium. IEEE, 2022.*

[13] *Guleng, Siri, et al. "Decentralized trust evaluation in vehicular Internet of Things." IEEE Access 7 (2019): 15980-15988.*

[14] *Najib, Warsun, and Selo Sulistyo. "Survey on trust calculation methods in Internet of Things." Procedia Computer Science 161 (2019): 1300-1307.*

[15] *Xu, Qian, Lei Zhang, and Yixiao Liu. "Enhancing Trust Management System for Connected Autonomous Vehicles Using Machine Learning Methods: A Survey." arXiv preprint arXiv:2505.07882 (2025).*

[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. https://doi.org/10.3390/data9090103.*

[17] *Wang, Jingwen, et al. "A survey on trust evaluation based on machine learning." ACM Computing Surveys (CSUR) 53.5 (2020): 1-36.*

[18] *Wang, Yingxun, et al. "Towards Strengthening the Resilience of IoV Networks—A Trust Management Perspective." Future Internet 14.7 (2022): 202.*

[19] *El-Sayed, Hesham, et al. "Machine learning based trust management framework for vehicular networks." Vehicular Communications 25 (2020): 100256*

[20] *Wang, Yingxun, et al. "Mesmeric: machine learning-based trust management mechanism for the internet of vehicles." Sensors 24.3 (2024): 863.*

[21] *Ou, Xiaolong, Bo Mi, and Hangcheng Zou. "Trust Evaluation of Intelligent Connected Vehicles Based on GNN: Multi-feature Fusion and Temporal Modeling." 2025 IEEE 14th Data Driven Control and Learning Systems (DDCLS). IEEE, 2025.*

[22] *Luo, Tingxi, et al. "Graph neural networks for trust evaluation: Criteria, state-of-the-art, and future directions." IEEE Network (2025).*

[23] *Yang, Liuqing, and Huiyun Li. "Vehicle-to-vehicle communication based on a peer-to-peer network with graph theory and consensus algorithm." IET Intelligent Transport Systems 13.2 (2019): 280*

[24] *Wu, Zonghan, et al. "A comprehensive survey on graph neural networks." IEEE transactions on neural networks and learning systems 32.1 (2020): 4-24.*

[25] *Zhang, Si, et al. "Graph convolutional networks: a comprehensive review." Computational Social Networks 6.1 (2019): 1-23.*

[26] *Velickovic, Petar, et al. "Graph attention networks." stat 1050.20 (2017): 10-48550.*

[27] *Sommer, Christoph, et al. "Veins: The open source vehicular network simulation framework." Recent advances in network simulation: the OMNeT++ environment and its ecosystem (2019): 215-252.*

[28] *Ruan, Wenbo, et al. "A double-layer blockchain based trust management model for secure internet of vehicles." Sensors 23.10 (2023): 4699.*