mathematics

MDPI

Article

A Trusted Measurement Scheme for Connected Vehicles Based on Trust Classification and Trust Reverse

Zipeng Diao [1], Mengxiang Wang [1], Qiang Fu [1,*], Bei Gong [2] and Meng Chen [1]

[1] China National Institute of Standardization, Beijing 100191, China; diaozp@cnis.ac.cn (Z.D.); wangmx@cnis.ac.cn (M.W.); chenmeng@cnis.ac.cn (M.C.)
[2] College of Computer Science, Beijing University of Technology, Beijing 100124, China; gongbei@bjut.edu.cn
* Correspondence: fuqiang@cnis.ac.cn

Abstract: As security issues in vehicular networks continue to intensify, ensuring the trustworthiness of message exchanges among vehicles, infrastructure, and cloud platforms has become increasingly critical. Although trust authentication serves as a fundamental solution to this challenge, existing models fail to effectively address the specific requirements of vehicular networks, particularly in defending against malicious evaluations. This paper proposes a novel multidimensional trust evaluation framework that integrates both static and dynamic metrics. To tackle the issue of malicious ratings in peer assessments, a rating reversal mechanism based on K-means clustering is designed to effectively identify and correct abnormal trust feedback. In addition, the framework incorporates an entropy-based trust weight allocation mechanism and a time decay model to enhance adaptability in dynamic environments. The simulation results demonstrate that, compared with traditional approaches, the proposed scheme improves the average successful information rate by 12% and reduces the false positive rate to 6.1%, confirming its superior performance in securing communications within the vehicular network ecosystem.

Keywords: connected vehicles; information security; trusted attestation; trust reverse

MSC: 94-10

check for updates

Academic Editor: Cheng-Chi Lee

Received: 28 March 2025
Revised: 18 April 2025
Accepted: 26 April 2025
Published: 28 April 2025

Citation: Diao, Z.; Wang, M.; Fu, Q.; Gong, B.; Chen, M. A Trusted Measurement Scheme for Connected Vehicles Based on Trust Classification and Trust Reverse. *Mathematics* **2025**, *13*, 1453. https://doi.org/10.3390/math13091453

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Connected vehicles represent a pivotal advancement in transportation technology, offering transformative capabilities to enhance system efficiency, reduce traffic incidents, improve safety measures, and mitigate congestion impacts [1]. Industry projections indicate a substantial growth trajectory, with the global connected vehicle market expected to reach 9.9 billion units by 2035 [2]. However, the rapid development of these technologies has concurrently introduced significant security vulnerabilities. A notable demonstration of these challenges occurred in 2015 when Chrysler initiated a recall of 1.4 million vehicles following the discovery of critical remote attack vulnerabilities [3]. The rapid development of IoT technologies (as surveyed in [4]) and mobile edge computing standardization efforts [5] have created both opportunities and challenges for secure V2X communications.

The automotive industry has progressively integrated vehicle-to-everything (V2X) communication technologies, encompassing both vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) systems, into modern vehicles. From a security perspective, ensuring the trustworthiness of messages exchanged among vehicles, infrastructure, and cloud platforms is paramount. Within the Car Communication Consortium, extensive deliberations have been conducted regarding security protocols and protective measures for

vehicular systems. These discussions culminated in the establishment of comprehensive security frameworks, including the definition of trusted assurance levels and specific security requirements [6].

The connected vehicles network, encompassing V2V, V2I, and V2X communications, exhibits significant architectural similarities with edge computing paradigms, as illustrated in Figure 1. Recent advancements in industrial edge computing [7] have demonstrated the critical role of low-latency processing and distributed trust management in mission-critical systems, providing valuable insights for vehicular network design. This correspondence necessitates a targeted consideration of both connected vehicle network characteristics and edge computing features in system design. The core research focus lies in developing efficient and comprehensive interaction mechanisms for trusted node access within these networks. However, existing trust models demonstrate limited capability in addressing the critical requirements for secure interaction among connected vehicle nodes, highlighting a substantial research gap in this domain.

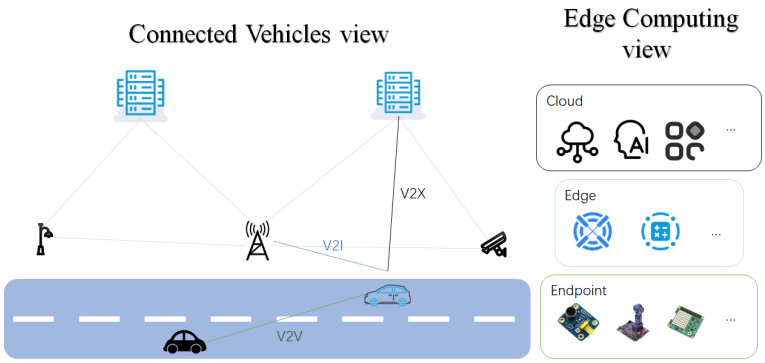


Figure 1. Connected vehicles structure.

The establishment of efficient node interaction fundamentally depends on the maintenance of robust trust relationships, which can be effectively quantified through trust models [8]. However, the implementation of these models necessitates the comprehensive consideration of multiple attributes. Current trust modeling approaches are primarily categorized into two distinct paradigms: centralized trust modeling [9,10] and distributed trust modeling [11,12], differentiated by the entity responsible for trust metric computation. A critical research challenge lies in developing effective trust assessment mechanisms for connected vehicle nodes, particularly in leveraging architectural features to integrate the respective advantages of both centralized and distributed trust models for comprehensive node trust modeling in connected vehicle networks.

In this study, we propose a novel trusted measurement scheme for connected vehicles, incorporating trust classification and trust reversal mechanisms. The primary contributions of this work are as follows:

- We have developed a comprehensive trusted measurement framework specifically designed for connected vehicle systems. This framework integrates carefully selected trust attributes and calculation methodologies, meticulously aligned with the unique security requirements of connected vehicle networks.
- We introduce an innovative score reversal mechanism to address malicious mutual evaluation scenarios. This mechanism demonstrates superior capability in identifying concealed malicious nodes compared to conventional approaches that fail to account for problematic scoring patterns.

This paper is organized into six distinct sections. Section 1 presents the research background and outlines the core innovations of this study. Section 2 provides a comprehensive

review of recent advancements in relevant research domains. Section 3 elaborates on the rationale behind the selection of trust attributes in our proposed framework. Section 4 details the methodology for trust value computation. Section 5 presents the experimental evaluation and analysis of the results. Finally, Section 6 concludes this paper with a summary of the proposed scheme and its key findings.

2. Related Work

2.1. Security Mechanism of Connected Vehicles

Despite extensive research efforts by both academia and industry, connected vehicle nodes continue to face significant security challenges in open environments. Several approaches have been proposed to address these challenges. Jiang et al. [13] identified four primary research directions for security enhancement in IoT and connected vehicles: authentication, access control, management, and low-cost encryption schemes. Alfardus and Rawa [14] developed an RNN-based detection system using wavelet transforms for real-time IVN protection. Aledhari et al. [15] proposed new CAV security protocols and threat classifications. Traditional solutions include Ning's [16] implementation of vendor-provided trusted execution environments to ensure secure node operation and Jung et al.'s [17] development of security-enhanced management platforms utilizing ARM Platform Security Architecture. However, these solutions exhibit limitations in terms of hardware dependency and limited generalizability. Recent studies on edge computing security frameworks [18] have emphasized the necessity of hierarchical protection mechanisms aligned with cybersecurity standards, which resonates with the distributed trust challenges in vehicular networks. Furthermore, while some researchers have explored time-automata-based secure gateways through firewall retrofitting similar to traditional network security approaches, such solutions demonstrate inadequate support for the distributed nature of connected vehicle networks. Current research continues to develop more adaptive solutions to overcome these challenges.The proposed trust evaluation framework operates without relying on proprietary hardware. It integrates multidimensional metrics, including static and dynamic trust, offering strong scalability and platform independence. This enables effective security protection for the Internet of Things and makes the framework well suited for large-scale deployment environments.

2.2. Trust Model

The concept of trust, originating from sociological studies, was first adapted for device management by Blaze et al. [19] in 1996. Their pioneering work conceptualized trust models as proactive defense mechanisms, establishing a framework for enhancing the trustworthiness of connected systems through trust extension. Building upon this foundation, recent advancements have significantly expanded trust management techniques for vehicular networks. Shahariar and Phillips [20] proposed a tamper-proof device-based trust management framework for VANETs that regulates driver behavior and reduces communication overhead by disseminating feedback only when conflicting information arises. Mwanje et al. [21] further introduced a fuzzy-logic-based trust evaluation framework that considers factors such as message freshness and sender location to determine the trust level of vehicles in the network. Ma et al. [22] developed an integrated trust-based evaluation model for edge nodes, facilitating efficient edge computing operations, while Sultan et al. [23] proposed a collaborative trust approach for detecting malicious nodes in vehicular ad hoc networks (VANETs). However, the literature lacks detailed documentation regarding the specific trust metric processes employed in their model. In the domain of vehicular networks, Hamssa [24] introduced a hybrid trust model (HTM) coupled with a malicious behavior detection system (MDS) specifically designed for VANET systems.

While effective within its target domain, this approach demonstrates limited applicability to other technological domains. The information-entropy-based multidimensional trust evaluation mechanism designed in this paper incorporates a time decay function, enabling the system to dynamically adjust the weight of each trust dimension over time. This approach allows for more effective differentiation between short-term anomalies and long-term malicious behaviors, thereby improving the accuracy of trust inference and ensuring its effective applicability in the Internet of Things domain.

2.3. Machine Learning

Machine learning (ML) has emerged as a prevalent and effective technique for malicious node detection in various network environments. Akbani et al. [25] pioneered an approach combining machine learning with reputation systems, while Liu et al. [26] developed a trust system that evaluates node trustworthiness through routing path analysis. Subsequent advancements by Liu et al. [27] introduced a classification scheme utilizing linear regression methods to distinguish between malicious and benign nodes, incorporating the potential for multiple mixed attacks. Mehmet Aslan and Sevil [28] Sen proposed a dynamic ML-based trust management model for VANETs, adaptively assessing trust levels to enhance malicious node detection while reducing computational overhead. Further research by Yang et al. [29] identified sophisticated selective edge packet attacks, highlighting the evolving intelligence of malicious nodes in executing targeted attacks. Wang et al. [30] developed MESMERIC, an ML-driven trust management mechanism that integrates direct trust, indirect trust, and contextual information to separate trustworthy vehicles via optimal decision boundaries. Rashid et al. [31] further designed an adaptive real-time detection framework for VANETs, employing a distributed multi-layer classifier validated through OMNET++/SUMO simulations, achieving high DDoS attack detection accuracy. However, despite these advancements, existing machine-learning-based approaches face limitations in their applicability to connected vehicle systems with edge computing capabilities, primarily due to high computational overhead and inherent model constraints. The lightweight rating reversal mechanism proposed in this paper employs the K-means clustering algorithm to detect abnormal rating behaviors and automatically correct malicious ratings. This significantly reduces the system's computational overhead while maintaining high detection accuracy, thereby enhancing its practical deployability in resource-constrained vehicular environments.

3. Trusted Multidimensional Assessment Metrics

In response to the critical data collection requirements and diverse security risks inherent in connected vehicle networks, we have developed a comprehensive trust evaluation framework comprising two distinct components: static trust and dynamic trust. The static trust component evaluates the inherent characteristics and integrity of node hardware and software, enabling the identification of fundamental device faults or persistent security compromises. Conversely, the dynamic trust component assesses real-time node behavior and data quality, facilitating the timely detection of anomalous activities and potential data inaccuracies, even in otherwise operational nodes. This dual-component approach provides a robust mechanism for comprehensive trust evaluation in connected vehicle systems.

The proposed network architecture in this study organizes connected vehicles into multiple clusters $(clu_1, clu_2, \ldots, clu_n)$, where each cluster is responsible for executing similar tasks or connecting to shared infrastructure. Within each cluster, numerous nodes $(node_{n1}, node_{n2}, \ldots, node_{nm})$ operate under the supervision of designated management nodes $(manage_{n1}, manage_{n2}, \ldots, manage_{nm})$. These management nodes, selected from infrastructure components or ordinary nodes, facilitate intra-cluster coordination and inter-

cluster communication. For the purpose of this study, we assume that all management nodes maintain complete security through the implementation of established security mechanisms. The comprehensive network architecture is illustrated in Figure 2.

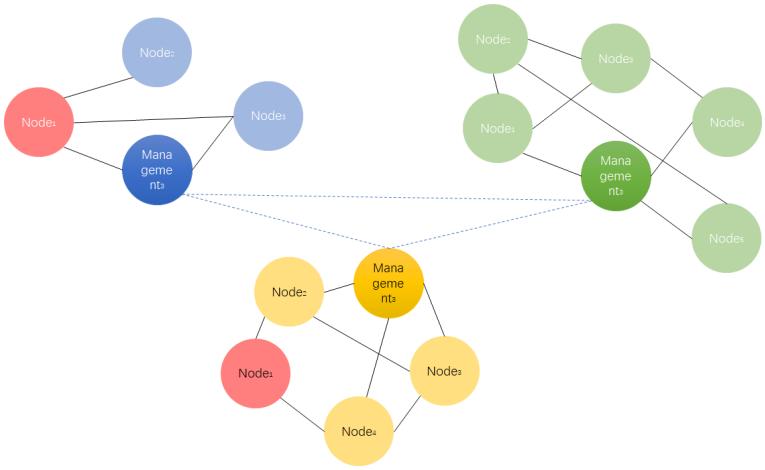


Figure 2. Node organization structure.

Each node must verify its trustworthiness to the manage through both static (inherent characteristics) and dynamic (real-time behavior) trust assessments.

3.1. Hardware Metrics

We represent hardware components as $(h_0, h_1, \ldots, h_k, \ldots, h_n)$, where $(h_1, \ldots, h_k)$ denotes non-replaceable node components and $(h_{k+1}, \ldots, h_n)$ indicates user-replaceable parts (e.g., batteries). During initialization, the nodes send hardware information to the manage as $(h_0^{'}, h_1^{'}, \ldots, h_n^{'})$.

The hardware trust value is shown in Equation (1):

$$Me(Ha) = \prod_{i=1}^{k} (h_i \wedge h_i^{'}) \wedge (\frac{1}{n-k} \sum_{i=k}^{n} h_i \wedge h_i^{'}) \tag{1}$$

The formulation explicitly distinguishes between critical hardware components, which are non-replaceable, and external extended hardware, which permits partial replacement. However, the safety thresholds governing such replacements are exclusively determined by system administrators.

3.2. Software Metrics

We characterize software components as $(so_0, so_1, \ldots, so_k, \ldots, so_n)$, where $(so_0, \ldots, so_k)$ encompasses critical, non-modifiable system software, including a bootloader and OS kernel, while $(so_{k+1}, \ldots, so_n)$ comprises user-configurable applications. Following system configuration, nodes submit their software information $(so_0^{'}, so_1^{'}, \ldots, so_k^{'}, \ldots, so_n^{'})$ to the manage during the initialization phase.

The software trust value is shown in Equation (2):

$$Me(So) = \prod_{i=1}^{k} (so_i \wedge so_i^{'}) \wedge (\frac{1}{n-k} \sum_{i=k}^{n} so_i \wedge so_i^{'}) \tag{2}$$

The formulation explicitly distinguishes between critical software components, which are non-replaceable, and external extended software, which permits partial modification. While users may sequentially deploy additional applications, the software trust value

maintains inherent resilience. However, the safety thresholds governing such modifications are exclusively determined by system administrators.

3.3. Dynamic Metrics

Given the frequent state transitions of operational nodes, the manage requires a periodic dynamic assessment of each node.

We categorize dynamic trust metrics into two distinct dimensions based on node characteristics: behavioral trust and data trust. Behavioral trust encompasses network behavior patterns, energy consumption profiles, and security policy compliance. Data trust is established through a comparative analysis of data collected by individual nodes against peer node datasets.

The dynamic assessment framework incorporates five key parameters: security policy compliance $Me(St)$, energy efficiency $Me(\Delta E)$, behavioral consistency $Me(Beh)$, network activity $Me(Na)$, and data reliability $Me(Data)$.

3.3.1. Security Policy Score

In connected vehicle networks, multiple nodes collaborate through inter-node communication to accomplish shared tasks. Within each device, communication occurs between specific processes. We model these communication processes as $c_1, c_2, \ldots, c_n$, with the corresponding permission sets required for task execution denoted as $\{j_1, j_2, \ldots, j_n\}$. The access privileges granted to external processes by a node are represented as j_a.

Then, the security policy trust value is shown in Equation (3):

$$Me(St) = \frac{1}{l-1} \sum_{i=1}^{l} j_i \wedge j_u \tag{3}$$

The security policy score quantifies the degree of alignment between the permissions required by the current process and the authorized permission set.

3.3.2. Energy Score

Energy status serves as a critical metric for node valuation, providing dual insights into system operation and security. Primarily, energy consumption patterns reflect the operational load distribution across IoT devices, potentially indicating network segmentation or uneven workload distribution. Secondarily, anomalous energy fluctuations may signify device malfunctions or security breaches. Node energy consumption originates from three primary sources: data transmission/reception, computational processing, and device operation. The input/output energy consumption E_{IO} is calculated as $E_{IO} = C_{IO} \cdot b + C_{Init} \cdot b \cdot d^2$, where b represents the number of transmitted/received bits, d denotes the inter-node distance, C_{IO} indicates the average energy consumption per bit, and C_{Init} accounts for the energy required to establish communication standards.

Computational energy consumption (E_{Com}) is determined by task complexity and device efficiency, while collection device consumption (E_{Coll}) depends on device power requirements and quantity. Given the practical challenges in the real-time measurement of E_{IO}, E_{Com}, and E_{Coll}, we employ an estimation approach based on the assumption of relatively stable operational patterns. Within a defined timeframe, the total energy reduction can be equated to the sum of these consumption components, the function as shown in Equation (4):

$$\begin{aligned} \Delta E &= E_t - E_{t-1} \\ &= E_{IO} + E_{Com} + E_{Coll} \end{aligned} \tag{4}$$

These consumption variations serve as critical indicators for dynamic trust assessment. For the energy score evaluation, we establish a threshold-based mechanism. Let Eth_i represent the acceptable energy deviation threshold for device i. The collective threshold vector for n devices is defined as $Energythreshold = \{Eth_1, Eth_2, \ldots, Eth_n\}$. Considering historical energy consumption data ΔE^h, the energy score can be computed through the following formulation, as shown in Equation (5):

$$Me(\Delta E) = \begin{cases} 1 & \Delta E - \Delta E^h < Energythreshold \\ \left| \frac{1}{\Delta E - \Delta E^h} \right| & \Delta E - \Delta E^h \geq Energythreshold \end{cases} \tag{5}$$

3.3.3. Behavior Score

Node behavior analysis focuses on data forwarding characteristics, encompassing three critical metrics:

Network forwarding rate nfr: When $node^{'}$ requests N data packets from node and node successfully forwards n packets $(n \leq N)$, the forwarding rate is evaluated based on the ratio of delivered packets received by $node^{'}$.

Network repeat rate nrr: This metric quantifies data duplication in node reports. A lower repetition rate indicates higher node trustworthiness. Conversely, increased repetition rates may suggest malicious activity, as compromised nodes often replay historical messages to evade detection, thereby reducing their trustworthiness.

Network transmission delay ntd: The transmission delay must remain within the network architecture's permissible range. The time interval for data forwarding from $node^{'}$ is constrained by system-specified thresholds.

The above node behavior ΩB can be described as shown in Equation (6):

$$\Omega B = (nfr, nrr, ntd) \tag{6}$$

The behavior score of ΩB is shown in Equation (7):

$$\mathrm{Me}(Beh) = \sum_{i=1}^{n} \Omega B_i \times {\Omega B_i}^h \tag{7}$$

${\Omega B_i}^h$ represents the historical value of the i-th component of the node behavior metric vector. It reflects the normal behavior pattern of the node in the past. By comparing the current value ${\Omega B_i}^h$with its historical value ${\Omega B_i}^h$, we can effectively detect abnormal changes in the node's behavior.

3.3.4. Node Activity

Node activity serves as a crucial indicator of the vitality and operational stability of a node. The trustworthiness of a node is positively correlated with its activity level, which is determined by the frequency and success rate of interactions with other nodes and clusters.

We define the node activity calculation function $NodeActCalc$ using Equation (8):

$$NodeActCalc(x) = 1 - \frac{1}{x + \varepsilon} \tag{8}$$

ε is a positive regulation constant; when $\varepsilon \to \infty$, $textcolorredNodeActCalc(x) \to 1$.

The node activity $Me(Na)$ is determined using Equation (9):

$$Me(Na) = \frac{NodeActCalc(node') + NodeActCalc(clu')}{2} \tag{9}$$

$NodeActCalc(node')$ represents the number of nodes interacting with the target node, while $NodeActCalc(clu')$ denotes the number of clusters interacting with the node.

3.3.5. Data Trust

Data trust is established by comparing a node's reported values with those of other nodes in the same environment to assess data accuracy. For instance, a temperature variation exceeding 5 °C within a confined space is highly improbable.

Significant deviations in a node's reported data relative to its peers may indicate either node malfunction or environmental anomalies. This cross-node validation mechanism aligns with the IoT data integration frameworks proposed in [32], where distributed device collaboration is essential for ensuring data credibility in large-scale sensing networks. To quantify data trust, we define an acceptable offset threshold for each measurable metric within the system. The collective threshold vector for n metrics is represented as $Datathreshold = \{Dtth_1, Dtth_2, \ldots, Dtth_n\}$. Given historical data $Data^h$, the data trust value can be calculated by the following formula, as shown in Equation (10):

$$Me(Data) = \sum_{i=1}^{n} \begin{cases} 1 & Data_{\mathrm{i}} - Data_i^h < Dtth_i \\ \left| \frac{1}{Data_{\mathrm{i}} - Data_i^h} \right| & Data_{\mathrm{i}} - Data_i^h \geq Dtth_i \end{cases} \tag{10}$$

4. Trusted Multidimensional Assessment Methods

4.1. Dynamic Trusted Weight

To effectively weight different trust dimensions, our methodology incorporates information entropy, a concept adapted from thermodynamic entropy by Shannon information theory. This approach provides a robust mechanism for quantifying uncertainty in trust evaluation, ensuring that metrics from more reliable nodes align more closely with actual conditions.

Building upon the trust value components outlined in Section 3, the information entropy Ct for each dimension is computed as shown in Equation (11):

$$\mathrm{Ct}(Me(i)) = -Me(i) \log Me(i) - (1 - Me(i)) \log(1 - Me(i)) \tag{11}$$

$Me(i)$ represents the trust value of a specific dimension for a node, and $1 - Me(i)$ denotes the corresponding untrustworthiness.

When comparing the information entropy of two distinct dimensions, a higher entropy value indicates greater uncertainty within that trust dimension. This increased uncertainty suggests a more significant contribution to the overall trustworthiness assessment, thereby warranting a higher weighting in the evaluation process.

Following the computation of information entropy Ct for various trust indicators, the trust differentiation Td for each dimension can be derived as shown in Equation (12):

$$Td_i = \begin{cases} 1 - \frac{1}{\log L} Ct(Me(i)) & Ct(Me(i)) > \rho \\ 0 & Ct(Me(i)) < \rho \end{cases} \tag{12}$$

L represents the number of trust levels, standardized to five in this study. In practical applications, this parameter is configurable by system administrators. The metric weights α_i for different trust indicators are calculated as shown in Equation (13):

$$\alpha_i = \frac{Td_i}{\sum_{i=1}^{n} Td_i} \quad i = 1, 2, \ldots, n \tag{13}$$

In the formula, $0 \leqslant \alpha_i \leqslant 1$, and $\sum_{i=1}^{n} \alpha_i = 1$.

4.2. Time Decay

Trust evaluation should incorporate not only the current state but also historical trust patterns and their temporal evolution. This temporal sensitivity necessitates a dynamic decay mechanism for historically acquired trust values over time.

Consider a node subjected to n trust evaluations between time $t - \Delta t$ and t, yielding a sequence of multidimensional trust assessment indicators $\{M1, M2, \ldots, M_n\}$. Here, M_1 represents the oldest metric while M_n corresponds to the most recent assessment. The time weight coefficient is defined as in Equation (14):

$$w_i = h(i)/i \tag{14}$$

$h(i)$ denotes the time decay function, constrained by $h(i) \in [0, 1]$. This decay function assigns temporally weighted significance to trust metrics, prioritizing recent interactions over historical ones. The time decay function can be formulated as shown in Equation (15):

$$h(i) = \begin{cases} 1 & i = n \\ h(i-1) = h(i) - \frac{1}{n} & 1 \leqslant i \leqslant n \end{cases} \tag{15}$$

4.3. Trust Feedback and Dispersion

Nodes inherently operate within clusters, necessitating frequent communication with peer devices and servers. Consequently, trust evaluation must incorporate assessments from other devices, with trust feedback and dispersion serving as fundamental mechanisms for node management. Each node is obligated to provide feedback following interactions, enabling the server to maintain unified control and update trust assessments based on collective feedback.

Following an interaction between a node and other devices, the subject node evaluates the guest node's behavior on a scale of $[0, 1]$, where 0 indicates complete dissatisfaction and 1 represents full satisfaction.

Let Ex_{ij} denote the interaction satisfaction of node P_i toward node P_j, which can be calculated as shown in Equation (16):

$$Ex_{ij} = \left[\prod_{i=1}^{n} \alpha_i \Omega B_i \right]^{1/n} \tag{16}$$

In this formulation, α_i represents the trust weight defined in Section 4.1 while ΩB_i denotes the i-th component of node behavior metrics.

Upon computing the interaction satisfaction, the *server* stores the results and updates the historical record vector as shown in Equation (17):

$$Ex_{ij}^{h} = (\left\langle t_1, Ex_{ij}^{h(1)} \right\rangle, \left\langle t_2, Ex_{ij}^{h(2)} \right\rangle, \ldots, \left\langle t_n, Ex_{ij}^{h(n)} \right\rangle) \tag{17}$$

t_i represents the timestamp of the i-th interaction and $Ex_{ij}^{h(i)}$ records the subjective satisfaction at time t_i.

The feedback trust value is calculated by aggregating historical interaction satisfaction, as shown in Equation (18):

$$Me(Ex) = w_t Ex_{ij} + w_{t-1} T_{ij}^{h} \tag{18}$$

Given the potential instability of node services, it is essential to calculate the trust dispersion based on historical interaction satisfaction after obtaining the direct trust value. This dispersion metric serves as a subjective evaluation of the reliability of the node's self-assessment and is subsequently reported to the server during feedback submission. The dispersion degree ρ_{ij} is calculated as shown in Equation (19):

$$\rho_{ij} = \frac{1}{\sqrt{\frac{1}{n-1}\sum_{k=1}^{n-1}\left(Ex_{ij}^{h(k)} - Me(Ex)\right)^2 + 1}} \tag{19}$$

The server maintains a feedback trust record matrix structured as shown in Equation (20):

$$TMa^{P_i \to P_j} = \begin{pmatrix} \langle Ex_{11}, \rho_{11} \rangle & \langle Ex_{12}, \rho_{12} \rangle & \cdots & \langle Ex_{1n}, \rho_{1n} \rangle \\ \langle Ex_{21}, \rho_{21} \rangle & \langle Ex_{22}, \rho_{22} \rangle & \cdots & \langle Ex_{2n}, \rho_{2n} \rangle \\ \cdots & \cdots & \cdots & \cdots \\ \langle Ex_{n1}, \rho_{n1} \rangle & \langle Ex_{n2}, \rho_{n2} \rangle & \cdots & \langle Ex_{nn}, \rho_{nn} \rangle \end{pmatrix} \tag{20}$$

The feedback trust value from P_i to P_j, denoted as $TMa^{P_i \to P_j}$, along with the dispersion degree ρ_{ij}, is recorded. When $i = j$, this record is excluded from the calculation of trust feedback, dispersion, and overall trust.

The trust evaluation for node j, denoted as $TMa^{P_* \to P_j}$, is calculated as the mathematical expectation of the j-th column in the feedback trust matrix.

4.4. Detection and Reversal Mechanism of Scores from a Malicious Evaluation

In scenarios where a system is compromised, attackers may infiltrate multiple devices to collaboratively manipulate trust evaluations. These compromised nodes may artificially inflate trust scores for other malicious nodes during the feedback process, thereby deceiving the system into classifying them as trustworthy. To address this vulnerability, we propose a score reversal mechanism designed to mitigate the impact of fraudulent evaluations from compromised devices.

Prior to implementing the reversal mechanism, we employ a filtering process to identify suspicious scores. Our methodology utilizes the k-means clustering algorithm for anomaly detection in trust evaluations.

The input is node list $node = \{node_{n1}, node_{n2}, \ldots, node_{nm}\}$, with the value of each node's trust Me and the feedback record matrix $TMa^{P_i \to P_j}$ as each variable's threshold. The K-means maximum number of iterations is $km^{\max}$. The output is a list of problematic mutual evaluations, denoted as fix. The algorithm procedure consists of the following steps:

- Initialization phase: The method utilizes node i's trust value Me_i and its evaluation tuple $\langle Ex_{ij}, \rho_{ij} \rangle$ toward node j to construct data points. Two initial centroids are randomly selected from these points: $(Me_x. < Ex_{xj}, \rho_{xj} >)$ and $(Me_y. < Ex_{yj}, \rho_{yj} >)$. The iteration counter kmi is initialized to 0, and a list fix is created to store identified problematic nodes.
- K-means clustering: Two cluster lists, $List1$ and $List2$, are initialized to represent the K-means clusters. A centroid change flag $flag_change$ is introduced to track updates to the centroids, with its default value set to false. In each iteration, kmi is increased by 1. For each data point $(Me_i. < Ex_{ij}, \rho_{ij} >)$, distances $dist1$ and $dist2$ to the current centroids are computed. The point is assigned to the cluster with the smaller distance, and centroids are recalculated based on the updated clusters. The algorithm terminates if the maximum iteration count km^{max} is reached or if centroids remain unchanged.

- Screening stage: The direct trust values of nodes in $List1$ and $List2$ are compared, and the cluster with lower trust values is identified as containing problematic mutual evaluations. These nodes are added to the problem group fix.

The pseudo-code is listed in Algorithm 1.

Algorithm 1 Calculate problematic scores

Input: Node list $node = \{node_{n1}, node_{n2}, \ldots, node_{nm}\}$; Each node's trust value Me, Feedback Record Matrix $TMa^{P_i \to P_j}$; Each variable's thresholdl; Maximum number of iterations $km^{\max}$.
Output: Problematic scores record matrix fix.
Each manager of clusters calculates its own nodes.

for $node_{ni}$ **do**
  Constructing $(Me_i. < Ex_{ij}, \rho_{ij} >)$.
end for
Randomly select two points $(Me_x. < Ex_{xj}, \rho_{xj} >)$ and $(Me_y. < Ex_{yj}, \rho_{yj} >)$, $flag_change = 0, kmi = 0$.
repeat
  $C_1 = , C_2 = , flag = false, kmi++$
  for $(Me_i. < Ex_{ij}, \rho_{ij} >)$ **do**
    $dist_1 = \sqrt{(Me_i - Me_x)^2 + \left(\langle Ex_{ij}, \rho_{ij}\rangle - \langle Ex_{xj}, \rho_{xj}\rangle\right)^2}$
    $dist_2 = \sqrt{(Me_i - Me_y)^2 + \left(\langle Ex_{ij}, \rho_{ij}\rangle - \langle Ex_{yj}, \rho_{yj}\rangle\right)^2}$
    if $dist_1 \le dist_2$ **then**
      $C_1 = C_1 \cup (Me_i. < Ex_{ij}, \rho_{ij} >)$
    else
      $C_2 = C_2 \cup (Me_i. < Ex_{ij}, \rho_{ij} >)$
    end if
  end for
  $\left(Me_{temp1}; \langle Ex_{temp1j}, \rho_{temp1j}\rangle\right) = \left(\frac{1}{|C_1|}\sum_{Me_i \in C_1} Me_i; \frac{1}{|C_1|}\sum_{\langle Ex_{ij}, \rho_{ij}\rangle \in C_1} \langle Ex_{ij}, \rho_{ij}\rangle\right)$
  $\left(Me_{temp2}; \langle Ex_{temp2j}, \rho_{temp2j}\rangle\right) = \left(\frac{1}{|C_2|}\sum_{Me_i \in C_2} Me_i; \frac{1}{|C_2|}\sum_{\langle Ex_{ij}, \rho_{ij}\rangle \in C_2} \langle Ex_{ij}, \rho_{ij}\rangle\right)$
  if $\left(Me_{tempx}; \left\langle Ex_{tempxj}, \rho_{tempxj}\right\rangle\right) \neq (Me_x. < Ex_{xj}, \rho_{xj} >)$ **then**
    $(Me_x. < Ex_{xj}, \rho_{xj} >) = \left(Me_{tempx}; \left\langle Ex_{tempxj}, \rho_{tempxj}\right\rangle\right)$
    $flag = true$
  end if
  if $\left(Me_{tempy}; \left\langle Ex_{tempyj}, \rho_{tempyj}\right\rangle\right) \neq (Me_y. < Ex_{yj}, \rho_{yj} >)$ **then**
    $(Me_y. < Ex_{yj}, \rho_{yj} >) = \left(Me_{tempy}; \left\langle Ex_{tempyj}, \rho_{tempyj}\right\rangle\right)$
    $flag = true$
  end if
until $kmi \le km^{max}$ or $flag == false$
if $Me_x \le Me_y$ **then**
  $Fix = C_1$
else
  $Fix = C_2$
end if
return Fix

The score reversal mechanism operates as follows: when a node receives a high trust score during feedback evaluation but demonstrates consistently low scores across other trust dimensions, the server will automatically replace the inflated score with a lower,

more representative value. Simultaneously, the system generates an administrative alert regarding the anomalous evaluation of the node in question.

The pseudo-code is listed in Algorithm 2.

Algorithm 2 Trust reverse

Input: Node list $node = \{node_{n1}, node_{n2}, \ldots, node_{nm}\}$; Each node's trust value Me, Feedback Record Matrix $TMa^{P_i \to P_j}$; Each variable's threshold; Problematic scores record matrix fix.
Output: Fixed feedback record Matrix ${TMa_{fix}}^{P_i \to P_j}$.
Each manager of clusters calculates its own nodes.

for $node_{ni}$ **do**
  if $node_{ni}$'s $Me < threshold$ **then**
    for $\langle Ex_{yj}, \rho_{yj} \rangle \in TMa^{P_i \to P_j}$ **do**
      if $\langle Ex_{yj}, \rho_{yj} \rangle > Fix$ and $node_{nj}$'s $Me < threshold$ **then**
        $TMa_{fix}^{P_i \to P_j} = TMa^{P_i \to P_j} - 1$
      else
        $TMa_{fix}^{P_i \to P_j} = TMa^{P_i \to P_j}$
      end if
    end for
  end if
end for
return $TMa_{fix}^{P_i} \to P_j$

4.5. Trust Grouping

Connected vehicle nodes exhibit characteristics analogous to social communities, where each cluster functions as a micro-society with distinct structures and roles. Within these clusters, nodes share resources and engage in interactions. A node's initial trust is significantly influenced by its cluster's reputation, particularly before establishing its own trust history.

Building upon the variables defined in previous sections, we define the static metric vector of a node as $IC = (Me(Ha), Me(So))$, and its operating vector as $RC = (Me(St), Me(\Delta E), Me(Beh))$.

Let $\overline{IC} = (\overline{Me(Ha)}, \overline{Me(So)})$ represent the expected static metric vector of managed devices from the perspective of $manage_{nm}$. For a node under evaluation $node_n$, the static metric evaluation function Me_{IC} in the trust grouping process is computed as shown in Equation (21):

$$Me_{IC}(\overline{IC}, IC) = \frac{\overline{IC} \cdot IC}{\sqrt{\overline{IC}^2}\sqrt{IC^2}} \tag{21}$$

$node_n$ exhibits dynamic trust characteristics, requiring the evaluation of its reliability over the time interval $[t_i, t_{i+1}]$ to determine its trustworthiness. This reliability is calculated as shown in Equation (22):

$$\mathrm{Me}_{RC} = \frac{MeIC(\overline{IC}, IC)}{t_{i+1} - t_i} \int_{t_i}^{t_{i+1}} Me(St)Me(\Delta E)\mathrm{Me}(Beh)dt \tag{22}$$

The administrator must establish a threshold th for group membership. A node $node_n$ is permitted to join the group managed by $manage_{nm}$ if its calculated trust metric satisfies $Me_{TC} > th$.

For nodes lacking direct neighbor trust relationships, their feedback trust values are derived from the cluster's average trust value.

4.6. Total Trust

For each node, we obtain both its direct trust status and the modified trust feedback from other nodes. Consequently, our proposed method can be applied to compute the total trust of a node as shown in Equation (23):

$$HT_{total} = \beta_o \sum_{i=1}^{7} \alpha_i Me(i) + \beta_s TMa^{P_* \to P_j} \quad (23)$$

β_0 and β_s represent the subjective and objective coefficients, respectively, which are determined by the system's conditions. These coefficients satisfy the constraint $\beta_0 + \beta_s = 1$. The term $Me(i)$ denotes the value of each relevant variable as defined in the previous section.

Other nodes assess whether they should interact with a given node based on its computed total trust T_{total} and their own trust requirements.

Following an interaction, the global trustworthiness of both nodes is updated based on the evaluation of the new interaction.

5. Scheme Analysis

To validate the proposed scheme, we developed a comprehensive test model and established an experimental environment, the detailed configuration of which is presented in Table 1.

Table 1. Experiment parameter.

Parameter	Value	Description
Node	100	Number of nodes
Management	4	Number of clusters
Times	150	System run cycle
Transenergy	20	Energy consumption during transmission (nJ per time)
Computeenergy	2	Energy consumption at the time of calculation (nJ per time)
Init problem node rate (%)	5 10 15 20	Number of problem nodes at initialization

We conducted comprehensive performance evaluations using MATLAB R2022a, comparing the proposed scheme with Hussein's scheme 1 [33] and Xie's scheme 2 [34]. And, Table 2 shows the comparison of the model architectures of the three solutions.

Table 2. Comparison of trust evaluation schemes.

Dimension	My Scheme	Hussein's Scheme	Xie's Scheme
Trust Dimensions	Five-dimensional—static + dynamic (hard/soft/behavior/energy/data)	Three-dimensional—community trust (interaction frequency/success rate/reputation)	Four-dimensional—dynamic behavior (network/computation/storage/communication)
Malicious Node Detection	Two-stage detection (K-means clustering + score reversal)	Community-voting-based anomaly removal	Single-threshold behavior deviation detection
Dynamic Adaptability	Time decay function + entropy-based dynamic weighting	Fixed sliding time window averaging	Exponentially weighted moving average (EWMA)
Computational Complexity	O(n log n), distributed computing	$O(n^2)$, full-connection voting mechanism	O(n), centralized computation
Applicable Scenario	Highly dynamic vehicular networks	Static IoT device networking	Medium/low-speed mobile sensor networks

The experimental setup consists of 100 connected vehicle nodes distributed within a 100 × 100 m area. Nodes in this environment are susceptible to malicious attacks or internal failures, both of which result in reduced trust values. Upon detection of anomalous behavior, compromised nodes are isolated from the network. We measured two critical performance metrics: the information success rate (ISR), which reflects communication reliability, and the trust rate (TR), which indicates system integrity.

Each simulation iteration involves randomly deploying a predetermined number of malicious nodes within the test area, followed by system initialization and operation.

For each evaluated scheme, we executed the simulation environment over 150 operational cycles. The resulting system node states are visually represented in Figures 3 and 4.

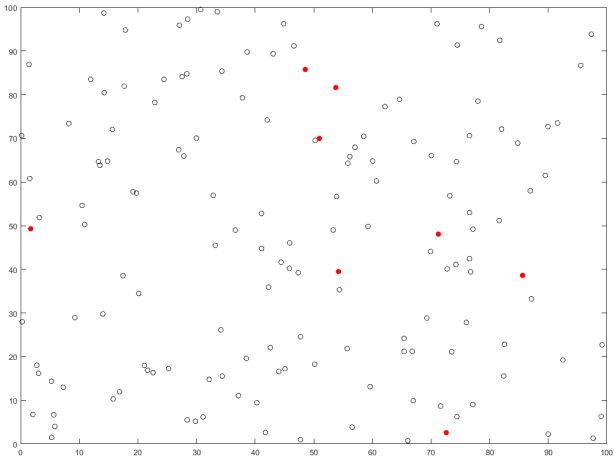
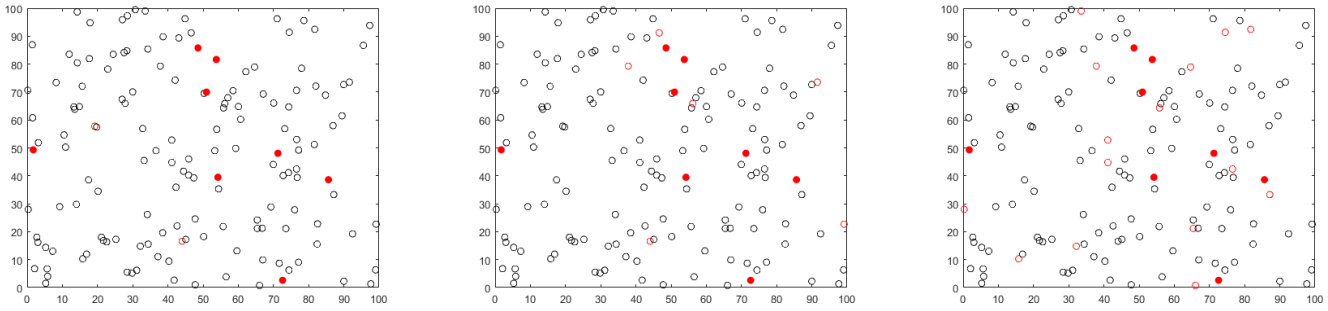


Figure 3. Initialized node distribution with 5% problem node status.



(**a**) (**b**) (**c**)

Figure 4. Node status after 150 cycles (5% problem node initial) for different evaluated schemes. Red nodes indicate malicious devices, the rest are normal devices. (**a**) Node status after 150 cycles with 5% initial problem nodes in our scheme. (**b**) Node status after 150 cycles with 5% initial problem nodes in Scheme 1. (**c**) Node status after 150 cycles with 5% initial problem nodes in Scheme 2.

A comparative analysis of the three schemes reveals that our proposed approach demonstrates superior performance in minimizing the incorporation of problematic nodes, represented by red hollow nodes in the visualization.

We systematically measured the average information success rate (ISR) across varying initial percentages of problematic nodes, ranging from 5% to 20%. The comprehensive results are presented in Figure 5.

The information success rate (ISR) for each operational cycle is graphically presented in Figure 6.

As illustrated in Figure 6, our proposed scheme demonstrates superior capability in maintaining the system ISR, particularly under conditions of low initial problematic node concentrations. Furthermore, in scenarios with higher initial problematic node populations, our approach exhibits enhanced performance following the stabilization of mutual evaluation metrics.

We systematically analyzed the trust rate (TR) across various initial system configurations, with the comprehensive results presented in Figure 7.

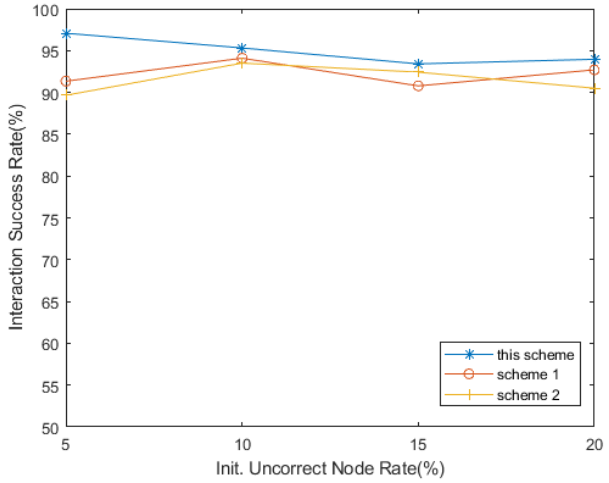


Figure 5. Average ISR in different initial states. Schemes follow the same formatting.

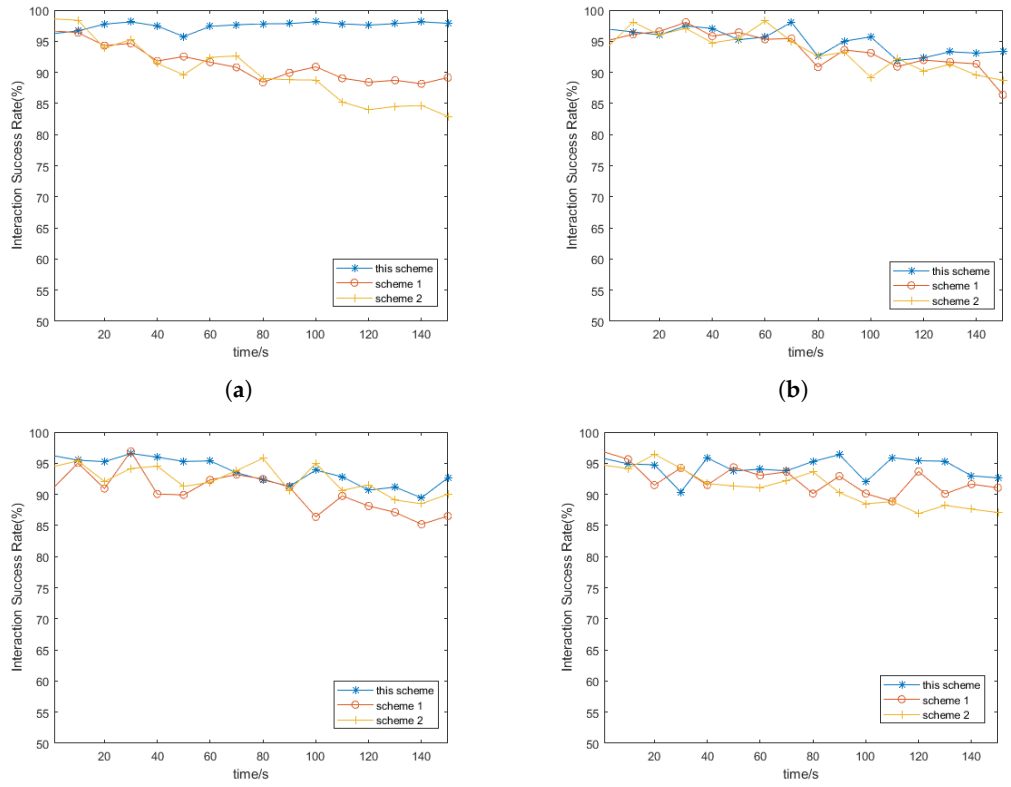


Figure 6. Change in system trust value in different scheme. (**a**) ISR change in 5% initial problem mode. (**b**) ISR change in 10% initial problem mode. (**c**) ISR change in 15% initial problem mode. (**d**) ISR change in 20% initial problem mode.

The proposed scheme demonstrates superior resilience, as evidenced by a slower degradation rate of trust states across various initial conditions compared to alternative approaches. When analyzed in conjunction with ISR metrics, these results confirm the efficacy of our novel score reversal mechanism and validate the overall effectiveness of our proposed solution.

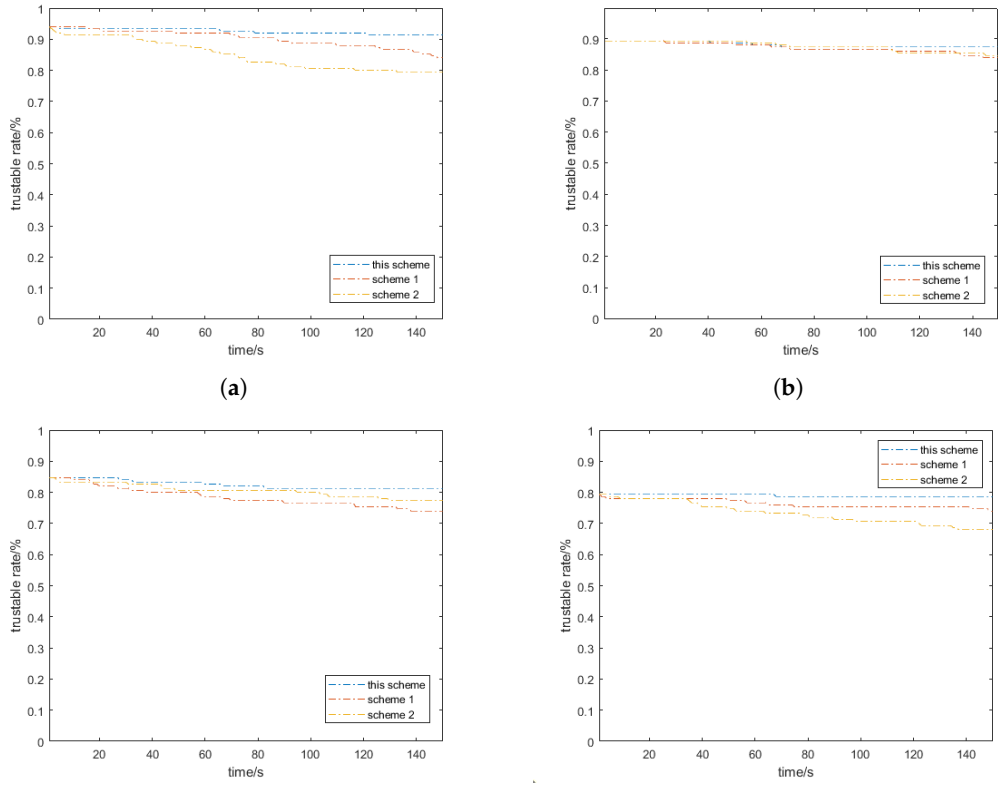


Figure 7. Change in system trust value in different scheme. (**a**) System trust value change in 5% initial problem mode. (**b**) System trust value change in 10% initial problem mode. (**c**) System trust value change in 15% initial problem mode. (**d**) System trust value change in 20% initial problem mode.

6. Conclusions

This study develops a trusted measurement framework for connected vehicle networks that unifies the static attestation of hardware and software with an entropy-weighted, time-decayed assessment of five dynamic dimensions. A two-stage defense, built on K-means outlier detection followed by score reversal, neutralizes collusive rating abuse and preserves rating integrity. Simulations with one hundred vehicles arranged in four clusters reveal that the framework raises the information success rate by 12 percent and caps the false-positive isolation rate at 6.1 percent, markedly outperforming Hussein's community trust and Xie's dynamic behavior baselines under identical conditions. Because the core algorithms execute with distributed O(n log n) complexity, they can run in real time on resource-constrained vehicular edge devices. Although the current evaluation is limited to MATLAB, the design is hardware-agnostic and cluster-aware, suggesting readiness for large-scale V2X deployments. Future work will extend verification to 5G/ITS-G5 testbeds, introduce reinforcement-based threshold tuning, and investigate latency and energy overhead in high-mobility settings to further refine practical applicability.

Author Contributions: Z.D. Methodology; M.W. software; Q.F. Resources; B.G. Validation; M.C. Text polishing. All authors have read and agreed to the published version of the manuscript.

Funding: Special Fundamental Research Fund for the Central Public Scientific Research Institutes (602024Y-11436, 602025Y-12515 and 602025Y-12516).

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Gupta, R.; Kumari, A.; Tanwar, S. A Taxonomy of Blockchain Envisioned Edge as a Connected Autonomous Vehicles. *Trans. Emerg. Telecommun. Technol.* **2021**, *32*, e4009. [CrossRef]
2. Fuji Keizai. *Investigations of the Connected Vehicle Marketing*; Marketing Reports; Fuji Keizai: Tokyo, Japan, 2018.
3. Miller, C.; Valasek, C. Remote Exploitation of an Unaltered Passenger Vehicle. In Proceedings of the Black Hat USA, Las Vegas, NV, USA, 1–6 August 2015.
4. Al-Fuqaha, A.; Guizani, M.; Mohammadi, M.; Aledhari, M.; Ayyash, M. Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications. *IEEE Commun. Surv. Tutor.* **2015**, *17*, 2347–2376. [CrossRef]
5. Hu, Y.C.; Patel, M.; Sabella, D.; Sprecher, N.; Young, V. Mobile Edge Computing—A Key Technology Towards 5G. *ETSI White Paper* **2015**, *11*, 1–16.
6. Kiening, A.; Angermeier, D.; Seudie, H.; Stodart, T.; Wolf, M. Trust Assurance Levels of Cybercars in V2X Communication. In Proceedings of the 2013 ACM Workshop on Security, Privacy & Dependability for Cyber Vehicles, Berlin, Germany, 4 November 2013; ACM: New York, NY, USA, 2013; pp. 49–60.
7. Wang, Q.; Jin, G.; Li, Q.; Wang, K.; Yang, Z.; Wang, H. Industrial Edge Computing: Vision and Challenges. *Inf. Control* **2021**, *50*, 257–274.
8. Li, X.-Y.; Gui, X.-L. Research on Dynamic Trust Model for Large Scale Distributed Environment. Ph.D. Thesis, South China University of Technology, Guangzhou, China, 2013.
9. Niu, B.; You, W.; Tang, H.; Wang, X. 5G Network Slice Security Trust Degree Calculation Model. In Proceedings of the 2017 3rd IEEE International Conference on Computer and Communications (ICCC), Chengdu, China, 13–16 December 2017; IEEE: New York, NY, USA, 2017; pp. 1150–1157.
10. Al-Khafajiy, M.; Baker, T.; Asim, M.; Guo, Z.; Ranjan, R.; Longo, A.; Puthal, D.; Taylor, M. COMITMENT: A Fog Computing Trust Management Approach. *J. Parallel Distrib. Comput.* **2020**, *137*, 1–16. [CrossRef]
11. You, J.; Shangguan, J.; Xu, S.; Li, Q.; Wang, Y. Distributed Dynamic Trust Management Model Based on Trust Reliability. *J. Softw.* **2017**, *28*, 2354–2369.
12. Amirian, S.; Taha, T.R.; Rasheed, K.; Arabnia, H.R. Generative Adversarial Network Applications in Creating a Meta-Universe. In Proceedings of the 2021 International Conference on Computational Science and Computational Intelligence (CSCI), Las Vegas, NV, USA, 15–17 December 2021; IEEE: New York, NY, USA, 2021; pp. 175–179.
13. Jiang, Y.; Zhang, Y.; Xu, A.; Kuang, X.; Meng, J.; Chu, H. An Overview: Data Security Mechanism of Power Terminal in Edge Computing. In Proceedings of the 2020 IEEE International Conference on Energy Internet (ICEI), Sydney, NSW, Australia, 24–28 August 2020; IEEE: New York, NY, USA, 2020; pp. 22–27.
14. Aledhari, M.; Razzak, R.; Rahouti, M.; Yazdinejad, A.; Parizi, R.M.; Qolomany, B.; Guizani, M.; Qadir, J.; Al-Fuqaha, A. Safeguarding Connected Autonomous Vehicle Communication: Protocols, Intra- and Inter-Vehicular Attacks and Defenses. *Comput. Secur.* **2025**, *151*, 104352. [CrossRef]
15. Alfardus, A.; Rawat, D.B. Machine Learning-Based Anomaly Detection for Securing In-Vehicle Networks. *Electronics* **2024**, *13*, 1962. [CrossRef]
16. Ning, Z.; Zhang, F.; Shi, W. A Study of Using TEE on Edge Computing. *J. Comput. Res. Dev.* **2019**, *56*, 1441–1453.
17. Jung, J.; Kim, B.; Cho, J.; Lee, B. A Secure Platform Model Based on ARM Platform Security Architecture for IoT Devices. *IEEE Internet Things J.* **2021**, *9*, 5548–5560. [CrossRef]
18. Wu, W.; Zhang, Q.; Wang, H.J. Edge Computing Security Protection from the Perspective of Classified Protection of Cybersecurity. In Proceedings of the 2019 6th International Conference on Information Science and Control Engineering (ICISCE), Shanghai, China, 20–22 December 2019; IEEE: New York, NY, USA, 2019; pp. 278–281.
19. Blaze, M.; Feigenbaum, J.; Lacy, J. Decentralized Trust Management. In Proceedings of the 1996 IEEE Symposium on Security and Privacy, Oakland, CA, USA, 6–8 May 1996; IEEE: New York, NY, USA, 1996; pp. 164–173.
20. Shahariar, R.; Phillips, C. A Trust Management Framework for Vehicular Ad Hoc Networks. *Int. J. Secur. Priv. Trust Manag.* **2023**, *12*, 15–36. [CrossRef]
21. Mwanje, M.D.; Kaiwartya, O.; Aljaidi, M.; Cao, Y.; Kumar, S.; Jha, D.; Naser, A.; Lloret, J. Cyber Security Analysis of Connected Vehicles. *IET Intell. Transp. Syst.* **2024**, *18*, 1175–1195. [CrossRef]
22. Ma, X.; Li, X. Trust Evaluation Model in Edge Computing Based on Integrated Trust. In *Proceedings of the 2018 International Conference on Algorithms, Computing and Artificial Intelligence*; ACM: Sanya, China, 2018; pp. 1–6.
23. Sultan, S.; Javaid, Q.; Malik, A.J.; Al-Turjman, F.; Attique, M. Collaborative-Trust Approach Toward Malicious Node Detection in Vehicular Ad Hoc Networks. *Environ. Dev. Sustain.* **2022**, *24*, 7532–7550. [CrossRef]
24. Hasrouny, H.; Samhat, A.E.; Bassil, C.; Laouiti, A. Trust Model for Secure Group Leader-Based Communications in VANET. *Wirel. Netw.* **2019**, *25*, 4639–4661. [CrossRef]

25. Akbani, R.; Korkmaz, T.; Raju, G.V.S. A Machine Learning Based Reputation System for Defending Against Malicious Node Behavior. In Proceedings of the IEEE GLOBECOM 2008—2008 IEEE Global Telecommunications Conference, New Orleans, LA, USA, 30 November–4 December 2008; IEEE: New York, NY, USA, 2008; pp. 1–5.
26. Liu, X.; Abdelhakim, M.; Krishnamurthy, P.; Tipper, D. Identifying Malicious Nodes in Multihop IoT Networks Using Dual Link Technologies and Unsupervised Learning. *Open J. Internet Things* **2018**, *4*, 109–125.
27. Liu, L.; Ma, Z.; Meng, W. Detection of Multiple-Mix-Attack Malicious Nodes Using Perceptron-Based Trust in IoT Networks. *Future Gener. Comput. Syst.* **2019**, *101*, 865–879. [CrossRef]
28. Aslan, M.; Sen, S. A Dynamic Trust Management Model for Vehicular Ad Hoc Networks. *Vehic. Commun.* **2023**, *41*, 100608. [CrossRef]
29. Yang, L.; Liu, L.; Ma, Z.; Ding, Y. Detection of Selective-Edge Packet Attack Based on Edge Reputation in IoT Networks. *Comput. Netw.* **2021**, *188*, 107842. [CrossRef]
30. Wang, Y.; Mahmood, A.; Sabri, M.F.M.; Zen, H.; Kho, L.C. MESMERIC: Machine Learning-Based Trust Management Mechanism for the Internet of Vehicles. *Sensors* **2024**, *24*, 863. [CrossRef]
31. Rashid, K.; Saeed, Y.; Ali, A.; Jamil, F.; Alkanhel, R.; Muthanna, A. An Adaptive Real-Time Malicious Node Detection Framework Using Machine Learning in Vehicular Ad-Hoc Networks (VANETs). *Sensors* **2023**, *23*, 2594. [CrossRef]
32. Georgakopoulos, D.; Jayaraman, P.P. Internet of Things: From Internet Scale Sensing to Smart Services. *Computing* **2016**, *98*, 1041–1058. [CrossRef]
33. Hussein, D.; Bertin, E.; Frey, V. A Community-Driven Access Control Approach in Distributed IoT Environments. *IEEE Commun. Mag.* **2017**, *55*, 146–153. [CrossRef]
34. Xie, L.X.; Wei, R.X. A Dynamic Trustworthiness Assessment Method for Internet of Things Nodes. *Comput. Appl.* **2019**, *3*, 2597–2603.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.