Anti-attack Trust Evaluation Algorithm Based on Bayesian Inference in VANET

Shusong Wei(✉), Xi Li, Hong Ji, and Heli Zhang

Key Laboratory of Universal Wireless Communications, Ministry of Education, Beijing University of Posts and Telecommunications, Beijing, People's Republic of China
{weishusong,lixi,jihong,zhangheli}@bupt.edu.cn

Abstract. Vehicular Ad-hoc Networks (VANETs) are crucial for intelligent transportation, improving traffic efficiency and safety. To enhance the security of VANET, trust management mechanism is implemented to defend against internal attacks in VANET. However, attacks targeting trust management mechanism, such as on-off attack, compromise trust management accuracy. In this paper, we propose the Anti-Attack Trust Evaluation Algorithm (AATEA) based on Bayesian inference to calculate trust values and establish reliable relationships among vehicles. AATEA addresses the challenge of on-off attack, where trust values are accumulated during continuous cooperation and suddenly initiate malicious behavior. Bayesian inference is employed to compute the trust values based on historical interactions. Additionally, we introduce an adaptive decay factor that considers the rate of change in trust values between the current and previous interaction of vehicles, to mitigate on-off attack. A dynamic driving reference set is designed based on the location information of received messages, since the forward and lateral vehicles of driving direction can provide more valuable information. Moreover, we built a VANET simulation platform using NS3 and SUMO, integrating security components, communication modules based on C-V2X, on-off attack and sybil attack module. Experimental results and comparisons with other VANET trust evaluation algorithms demonstrate AATEA's superior performance in trust value principles.

Keywords: VANET · Trust evaluation · An-ti attack · safety

1 Introduction

Vehicle Ad Hoc Network (VANET) is a crucial component of the Intelligent Traffic System (ITS) [1] and brings numerous benefits to road safety, traffic efficiency, and local services. However, VANET faces various security threats, including external and internal attacks [2], due to its characteristics of openness, self-organization, and dynamic topology. Traditional security measures, such as authentication [3], signatures [4], and Public Key Infrastructure (PKI) [5], primarily focus on defensing external attacks and fail to address internal attacks.

F. Gao et al. (Eds.): Chinacom 2023, LNICST 590, pp. 142–161, 2024.
https://doi.org/10.1007/978-3-031-67162-3_10

Internal vehicles with authorization and certification can mislead other vehicles in VANET by broadcasting false road messages. Hence, to against internal attacks, trust evaluation in VANET has been proposed [6]. This method assesses the trustworthiness of vehicles according their social factor and historical interactions, enabling vehicles in VANET to choose trustworthy vehicles for cooperation and avoid malicious vehicles.

However, the high scalability of VANET and the dynamic changes in the network topology caused by fast-moving vehicles pose a significant challenge in quickly accessing the trustworthiness of vehicles in VANET. Moreover, to undermine the trust mechanism's reliability, attacks specifically targeting the trust mechanism have emerged, including on-off attack and sybil attack. Consequently, effectively detecting malicious information and evaluating the trust value of each vehicle becomes a major challenge in VANET.

In recent years, trust mechanism in VANET has received significant attention in VANET security research. In [7], the author analyzed various proposed trust building and management mechanisms in VANET, highlighting the weaknesses of deploying existing trust management methods in VANET. Generally, these algorithms utilize different theories such as graph theory, fuzzy logic, D-S evidence theory, and collaborative filtering. In [8], the author introduced Implicit Web of Trust in VANET (IWOT-V) to generate trust values, utilizing the PageRank algorithm to establish an implicit trust network. However, IWOT-V is vulnerable to some attack models. In [9], a three-factor based trust model was proposed, utilizing expectation, risk and confidence to generate trust values. Nevertheless, this method occupies significant bandwidth, and reducing communication efficiency. Additionally, [10] introduces the concept of actively assessing trust values, where nodes periodically engage in evaluating the trustworthiness of interacting nodes. Furthermore, it introduces the concept of neighbor-recommended trust values, considering factors such as energy consumption. The utilization of blockchain technology is also employed for the sharing of trust values among Roadside Units (RSUs). However, this scheme does not inherently possess built-in resistance against attacks.

In VANET, trust management models can be categorized into three types: entity-based trust models, data-based trust models, and composite trust models. Entity-based trust models typically use direct trust between vehicles or neighbor-recommended trust to calculate trust values for each node. However, a challenge is collecting enough information, especially when vehicle nodes are new or in sparsely populated areas with limited interactions. Data-based trust models focus on assessing the credibility of received data. They collect messages from various sources, filter out untrustworthy data, and establish transient trust for each data event. The limitation is the inability to form permanent trust relationships among vehicle nodes. Composite trust models combine elements of both entity-based and data-based trust models. They assess trustworthiness of vehicle nodes and data credibility, with both aspects influencing each other. However, they tend to be complex and have higher computational costs.

To address these issues, we propose AATEA to access the trustworthiness of vehicles in VANET, which optimizes the composite trust model, referred to

as a hybrid trust model. The process involves two key steps: initially, a Misbehavior Detection System (MDS) identifies messages as either malicious or non-malicious, and subsequently, a trust evaluation algorithm assigns trust values to vehicles. Vehicles with low trust values are placed on a blacklist. We employ Bayesian inference to calculate trust values based on historical interactions, providing a level of independence from expert knowledge. To address internal attacks within VANET, a neural network serves as the MDS, proficient in detecting such attacks. To counter anti-attack challenges, we have designed an adaptive decay factor, contingent on the change rate of weighted trust values. This adaptive factor mitigates the rapid inflation of vehicle trust values, effectively countering on-Off attacks. Additionally, a dynamic trust redemption window, based on the historical interaction, is designed to address situations where malicious vehicles exhibit good behavior. Furthermore, a dynamic driving reference set is introduced based on location information from received messages, utilizing valuable information from forward and lateral vehicles in the driving direction.

Moreover, we have built a VANET simulation platform based on NS3 and SUMO. Within NS3, we enhanced the security components and integrated both the traffic flow loading module and the communication module based on C-V2X technology. Additionally, a neural network has been integrated into the MDS, and modules simulating on-Off attacks and Sybil attacks have been introduced. The proposed trust evaluation algorithm is implemented using the python module. Furthermore, we have conducted validation experiments to assess the effectiveness of our proposed approach, comparing it with other trust evaluation methods in VANET. Lastly, we discussed the feasibility of applying AATEA to practical VANET, as well as the challenges posed by factors such as computational power, latency, scalability, and cost.

2 System Model

2.1 VANET Message Format

To analyze malicious behavior in our scheme, we focus on exchanging messages related to rationality checking and trust building, without specifying encryption parameters such as keys or certificates in PKI. In this paper, we define the VANET information format as an array (t, s_t, e_t). Here, t represents the time when the message was generated. $s_t = x_t, v_t, a_t, d_t$, which represents the state of a vehicle at time t, including its location, velocity, acceleration and driving direction. The variable e_t represents internal events(e.g. emergency braking) or external events (e.g. road congestion) that may impact the vehicle's current

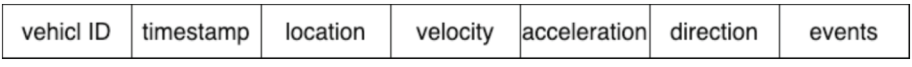


Fig. 1. Packet format in VANET.

state. Consequently, the messages that vehicle v_i broadcasts at time t can be represented as $m_t^i = (i, t, s_t, e_t)$. The packet format in VANET is illustrated in Fig. 1.

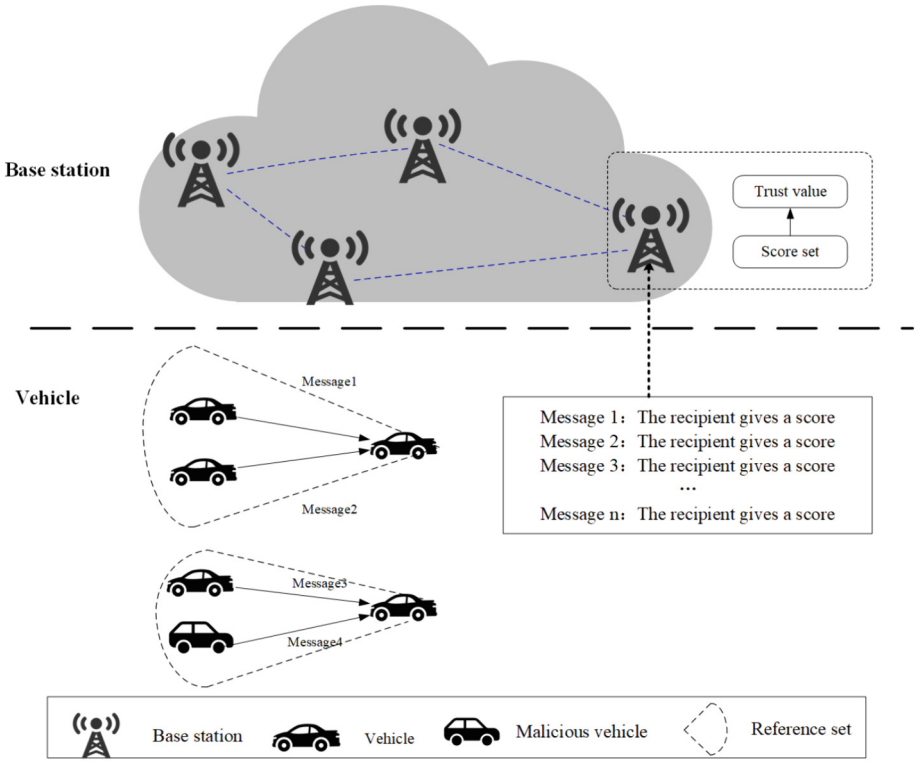


Fig. 2. Trust management system model for VANET.

2.2 Base Station Side

Base stations, compared to vehicles, possess higher computing power, storage capacity, and a reliable energy supply, making them well-suited for undertaking key tasks in trust evaluation. These tasks encompass the collection of score data submitted by vehicles, as well as the computation, storage, and dissemination of trust values. The base stations are interconnected via wired networks, collectively maintaining and updating trust values for all vehicles. This establishes a trust network independent of the actual topology of the VANET, aiming to mitigate the impact of the dynamic VANET topology. In our proposed system model, the base station assumes two primary responsibilities: score data collection and trust value evaluation.

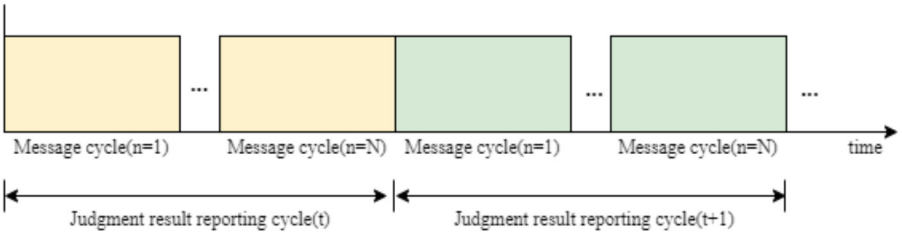


Fig. 3. Vehicles periodically upload score to the base station.

The score data is generated by the receiver of a message, and we define the judgment result of vehicle v regarding the message received from vehicle w at time t as $j_{v,w,t}$. This judgment result serves as an evaluation of the reliability of received messages, presenting a binary distribution that distinguishes between normal and abnormal messages.

Vehicles are required to periodically upload their score data to nearby base stations, as illustrated in Fig. 3. The reporting period is set to 1 s. Within each reporting cycle, vehicles engage in N message sending and receiving cycles, with each cycle lasting approximately 100 ms. During reporting cycles, vehicle v uploads the judgment results of messages received from vehicle w to the base station, which subsequently collects and maintains these score data. Additionally, the base station employs the algorithm proposed in Sect. 3 to generate the trust values of all vehicles which have Interacted with other vehicles. Once the trust value of a vehicle drops below a certain threshold, it is considered a malicious vehicle. The base station then adds it to the global blacklist and periodically distributes this global blacklist to vehicles in VANET, preventing other vehicles from receiving messages from this blacklisted vehicle.

2.3 Vehicle Side

Each vehicle is equipped with an on-board sensing module, computing module, and communication module [11]. These components enable the vehicle to perform data collection, processing, and communication. Additionally, each vehicle node is equipped with MDS for outlier detection.

The vehicle utilizes the perception module to sense the surrounding environment in real-time and detect specific events such as traffic accidents or road damage. Subsequently, the communication module enables the vehicle to broadcast a message informing nearby vehicles about the type and location of the event. However, not all information is equally relevant to every vehicle. For instance, if a vehicle has already passed through a particular location, reports of events occurring at that location hold less reference value for that vehicle. Consequently, each vehicle maintains a dynamic collection of reference vehicles during its journey, denoted as $V(V_1, V_2, V_3, ...)$. The dynamic reference set is composed of vehicles traveling in the current vehicle's forward and lateral directions, as illustrated in Fig. 2.

Messages transmitted by these reference vehicles possess a higher reference value for influencing the driving decisions of the target vehicle.

Nevertheless, due to potential perception errors, device failures, or malicious behavior, messages posted by vehicles are often unreliable. Therefore, the receiver of a message must utilize the MDS system to assess the trustworthiness of the message by analyzing all messages related to an event. Subsequently, the receiver uploads the score data to a nearby base station.

2.4 On-Off Attack and Sybil Attack Model

On-off attack are internal attacks that can harm distributed networks. In this attack, compromised nodes initially behave normally but switch to malicious actions once a condition is met. They engage in negative behaviors to lower their trust values, and when the values drop below the threshold, they resume positive behaviors to raise their trust values again. This nonstationary behavior poses challenges for trust management. In this paper, the pseudocode for the defined on-off attack is depicted in Algorithm 1.

Algorithm 1. On-off attack algorithm.

```
1: function "OnOffGenerate(T,n,m,sTimeTotal)"
       initial attack cycle T;    //Initialize the attack cycle
       initial good behavior percentage n;    //Percentage of normal behaviors that
   are initialized
       initial bad behavior percentage m;    //Percentage of Initialization Attacks
       initial total simulation time sTimeTotal;    //Total initialization simulation
   time
       query simulation time simTime;    //Query the current simulation time
2:     if simTime <sTimeTotal then
       normalMessage = traci.vehicle.getPositionSpeed;    //Get normal position,
   speed
       attackMessage = (100,100,100);    //False position and velocity as fixed val-
   ues
3:         if simTime <60 then
               send(normalMessage);    //When the simulation moment is less than
   60, it first behaves as a normal vehicle and accumulates trust values
4:         else
5:             if ((simTime - 60) % T) <n * T then
                   send(normalMessage);    //In off state, send normal message
6:             else
                   send(attackMessage);    //In on state, send attack message
7:             end if
8:         end if
9:     end if
10: end function
```

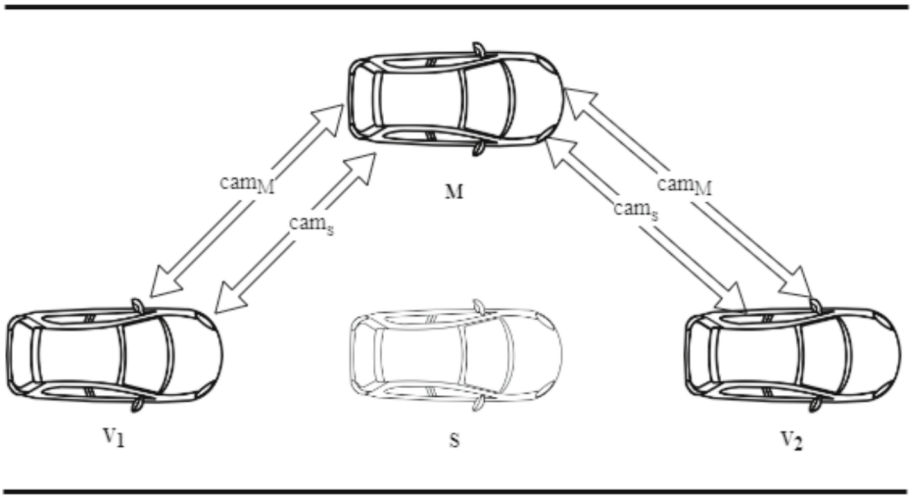


Fig. 4. Sybil attack Illustration.

Sybil attack module primarily simulates sybil attacks by implementing the injection of false location. In VANET, malicious vehicle nodes that have been compromised or intruded upon can collectively send fabricated erroneous information to surrounding normal vehicles, potentially causing traffic accidents or congestion. This type of coordinated deceptive attack poses a severe threat to road safety. In this paper, normal vehicles are denoted as V, malicious nodes are represented as M, and sybil nodes are denoted as S. The process by which attacker M introduces sybil nodes into the VANET is illustrated in Fig. 4. Attacker M broadcasts both cam_M and cam_S into the network, with cam_M containing normal location, speed, and other information computed based on NS3's motion model, resembling typical vehicle information. cam_S on the other hand, is fabricated by attacker M after obtaining the IDs of other normal vehicles. If normal nodes V_1 and V_2 lack MDS, upon receiving cam_S, they erroneously identify S as a normal node and add it to their neighbor lists. The broadcasted cam_S by the attacker can lead normal nodes V_1 and V_2 to mistakenly perceive the sybil node S as a genuinely existing normal vehicle. Consequently, they may initiate evasive lane-changing actions.

3 Anti-attack Trust Evaluation Algorithm

Each vehicle in the VANET maintains a reference set of forward and lateral vehicles denoted as $V(v_1, v_2, v_3, ...)$. The base station stores the historical trust values of all vehicles that have interacted with it, defined as $T_V(t-1)$. When a vehicle v needs to receive a message from another vehicle w, vehicle v first checks its local blacklist. If vehicle w is on the blacklist, it rejects receiving the message. If vehicle w is not on the local blacklist, it first requests the latest

global blacklist for the area from nearby base stations and performs a second check. If it is confirmed that vehicle w is not on the blacklist, it then invokes the MDS equipped on the vehicle to inspect the received message. If the inspection confirms it as a normal message, it is accepted; otherwise, it is discarded. Let $R(v, w, t)$ represent the message received by vehicle v from vehicle w at time t.

Upon receiving a message from vehicle w, vehicle v executes its on-board MDS, which analyzes the input message $R(v, w, t)$ to determine if it is a normal message. In this evaluation, we assume that the results are categorized into two levels: *trustworthy* or *untrustworthy*. We denote the judgment result of vehicle v on the message received from vehicle w at time t as below:

$$j_{v,w,t} = \begin{cases} 1, & abnormal, \\ -1, & normal. \end{cases} \tag{1}$$

If the message is deemed an abnormal, vehicle v will reject the message and assign a score of -1 to it. The evaluation process flowchart is illustrated in Fig. 6.

Figure 3 illustrates the reporting period for each vehicle's judgment, which is set at 1 s. Within each reporting period, the vehicle engages in N cycles of message sending and receiving. During each message sending and receiving cycle, the vehicle evaluates the messages sent by other vehicles and decides whether to receive them or not. We define the judgment result of vehicle v for all messages sent by vehicle w during each reporting period as $M_(v, w) = m(v, w, 1), m(v, w, 2), ..., m(v, w, t)$. At the conclusion of each reporting period, vehicle v reports the resulting judgment result $M_(v, w)$ to the nearest base station by uploading the information.

During each judgment reporting cycle, all vehicles that have uploaded their judgment results to a specific base station are denoted as $S = 1, 2, 3, ...$ The base station receives the judgment result $M_{S,w}$ from the vehicle collection S for all messages sent by vehicle w during the current reporting cycle, where $S = 1, 2, 3, ...$ The base station proceeds to calculate the trust value of each vehicle using the following definitions: K represents the number of times vehicle v judges that the message sent by vehicle w is a normal message, and P represents the number of times vehicle v judges that the message sent by vehicle w is an abnormal message.

Utilizing the Bayesian inference of the beta distribution, we set $\alpha = K + 1, \beta = P + 1$, Consequently, the local trust of vehicle v towards vehicle w at time t can be calculated as follows:

$$T_{v,w,t} = \frac{\alpha}{\alpha + \beta}. \tag{2}$$

Algorithm 2 provides a detailed description of the algorithm process for calculating global trust values.

Define $T_(w, t)weight$ as the weighted value of all vehicle's local trust to vehicle w at time t. It can be represented as:

$$T_{(w,t)}\text{weight} = \frac{T_1 t}{T_1 t + \ldots + T_S t} T_{1,w,t} + \ldots + \frac{T_S t}{T_1 t + \ldots + T_S t} T_{s,w,t}, \tag{3}$$

Algorithm 2. Local trust evaluation algorithm

Input: $k \geq 0$ //number of normal messages
$p \geq 0$ //number of abnormal messages
$n_{\min} > 0$ //minimum number of samples required for the first update
Output: $T_{v,w,t}$ //Local trust value of vehicle v to vehicle w
if First calculation $T_{v,w,t}$ **then**
  if $n < n_{\min}$ **then**
    return null
  else
    set $\alpha = 1$ and $\beta = 1$
  end if
else
  Read from database:α and β
end if
$\alpha = k + 1$
$\beta = p + 1$
$T_{v,w,t} = \frac{\alpha}{\alpha+\beta}$
save α,β,$T_{v,w,t}$to database
return $T_{v,w,t}$

where $T_S t$ represents the trust value of vehicle S at time t.

To prevent the trust value from rapidly increasing, we introduce an adaptive decay factor λ to update the global trust value. This decay factor is determined by the change rate of the weighted trust value between the current moment and the previous moment. The change rate of the weighted trust value γ can be expressed as follows:

$$\gamma = \frac{T_{(w,t)}weight - T_{(w,t-1)}weight}{T_{(w,t)}weight}, \tag{4}$$

where $T_(w,t)weight$ is the weighted value of each vehicle's trust score to the w vehicle at time t; $T_(w,t-1)weight =$ is the weighted trust value of vehicle w in last moment. Define θ as the adjustment parameter, and $\theta \geq 0$. θ determines the change rate of λ. Finally, we get the adaptive decay factor λ as follows:

$$\lambda = \frac{1}{1 + e^{-\theta\gamma}}, \tag{5}$$

We drew inspiration from the Sigmoid function's S-shaped curve, ranging from 0 to 1, and introduced an adjustable parameter, θ, to control the curve's slope. In Fig. 4, we depict the variability of the adaptive decay factor, λ, concerning γ for different θ values. Larger θ values result in a more pronounced λ change near zero, emphasizing the influence of even minor differences between $T_{(w,t)}$ and $T_{(w,t-1)}$. By combining Eqs. 4–6, we observe that rapid trust value increases lead to higher λ, keeping $T_{(w,t)}$ relatively lower, effectively curbing abrupt trust value spikes and countering on-off attacks. Similarly, swift trust value decreases result in smaller λ, causing $T_{(w,t)}$ to drop quickly, adhering to the "hard to get and easy to lose" principle in trust models (Fig. 5).

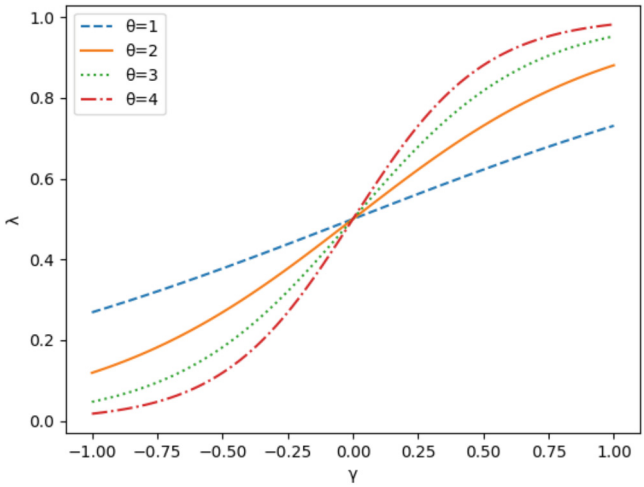


Fig. 5. The decay factor λ under different parameter θ and its variation with respect to parameter γ.

By combining the scores reported by the vehicle collection S, the base station calculates the global trust value of vehicle w as follows:

$$T_{(w,t)} = \lambda T_{(w,t-1)} + (1-\lambda) T_{(w,t)} weight, \tag{6}$$

where λ is designed as a dynamic variable to control the growth rate of the trust value score, ensuring slow growth and rapid decrease. After the base station completes the calculation of the trust value for vehicle w at time t, it updates the historical trust value of the vehicle in the database. Additionally, If there are updates to the global blacklist, it is disseminated to all vehicles within the communication range. Algorithm 3 provides a detailed description of the algorithm process for calculating global trust values.

In this scenario, when vehicles that have been blacklisted start exhibiting good behavior and attempt to rejoin the VANET, we have implemented a dynamic trust redemption time window. Within this time window, the base station actively engages with the vehicles on the blacklist, requesting them to report messages. The base station then evaluates the reported messages to decide whether to lift the blacklist restrictions. We denote this dynamic trust redemption time window as:

$$Redeem_t = \begin{cases} \frac{1}{P} \cdot \frac{1}{B} \cdot 60s, & \text{if } B < 3, \\ 0, & \text{if } B \geq 3, \end{cases} \tag{7}$$

Algorithm 3. Global trust evaluation algorithm.

Input: $T_{v,w,t}$ for all vehicles //number of normal messages
$\theta > 0$ //Adaptive parameters
$N > 0$ //Total number of vehicles in VANET
V //Dynamic driving reference
Output: $T_{w,t}$ //Global trust value of vehicle w
1: **for** each vehicle $v \in V$ **do**
2: Calculate the weighted trust value of vehicle w according to formula 3
3: **end for**
4: Query the weighted trust value of the previous moment
5: Calculate the weighted trust value change rate γ according to formula 4
6: Calculate the dynamic attenuation factor λ according to formula 5
7: Calculate the global trust value of vehicle w using formula 6
8: Save $T_{w,t}$ to database
9: return $T_{w,t}$

where P represents the number of times vehicle w sends abnormal messages, while B represents the number of times vehicle w is blacklisted. The worse the historical performance of vehicle w, the shorter its trust redemption window will be. If it has been blacklisted more than three times, it will be permanently placed on the global blacklist.

4 Simulation Platform

4.1 Overall Architecture of the Simulation Platform

In this chapter, we introduce the Vehicle-to-Everything (V2X) trust simulation platform, which consists of three key modules. The first module is the SUMO module, responsible for generating mobile models and simulating traffic flow. This module plays a crucial role in creating a realistic environment for vehicle movement and traffic scenarios. The second module is the NS3 module, primarily used for simulating vehicle interactions and communication, as well as network attack scenarios. This module allows researchers to explore the impact of attacks on trust and evaluate the resilience of trust management mechanisms. The third module serves as the primary component for implementing our trust evaluation algorithm. It includes the trust calculation module, responsible for computing trust values based on predefined algorithms, and the neural network for MDS, which identifies and flags any suspicious or malicious messages. These modules work in conjunction to assess the trustworthiness of vehicles within the V2X network.

4.2 SUMO Modular

We used the SUMO micro traffic flow simulator to simulate and control vehicle nodes in detail. The simulator generated track files compatible with network

simulation software. Our scenario consisted of a 2 km by 2 km area with a controlled intersection, including two-way roads, four lanes, and four traffic lights. Vehicles approached the intersection, obeying the traffic lights and broadcasting their acceleration, speed, position, and other data to nearby vehicles.

SUMO communicates with NS3 through the *TraCI* interface, and researchers can observe the real-time simulation process in the SUMO user interface on the NS3.

4.3 NS3 Modular

We utilize C-V2X technology to establish the VANET communication function, enabling vehicles to send and receive data packets. The NS3 module incorporates the vehicle flow loading module, which facilitates the selection of online OSM maps and enables bidirectional coupling with the SUMO module. Moreover, we have designed an on-off attack and sybil attack module. NS3 serves as the central component of our platform, seamlessly integrating with the SUMO and python modules to implement the trust management mechanisms.

The NS3 module of our VANET simulation platform encompasses several essential steps, which include:

Input/Output Flow: Create a *CSV* file to record all the information regarding interactions between vehicles during the simulation process. This include vehicle IDs, speeds, accelerations, timestamps, and other relevant data. This *CSV* file will be utilized by the python module for vehicle trust management.

Vehicle Networking Node Container: It initializes a container to store all vehicle nodes, used to organize and manage vehicle nodes for batch creation, configuration, and operations, simplifying the management and maintenance of complex network simulations.

Node Initialization: This step involves creating and initializing vehicle nodes. Each node is configured with network equipment, physical layer settings, IP address, application layer settings, etc.

Mobile Model Loading: The trace file generated by the SUMO module is read, and each vehicle node is associated with the trace file. The NS3 mobile model is initialized for each node using the *ConstantPositionMobility* Model, which controls the node's trajectory based on the SUMO trace file.

Packet Sending and Receiving: This includes the sub-functions for receiving and sending packets. The *SidelinkV2xAnnouncementMacTrace* function is used to receive packets and is scheduled in the NS3 event plan. It is defined that the current node broadcasts a message every 100 ms.

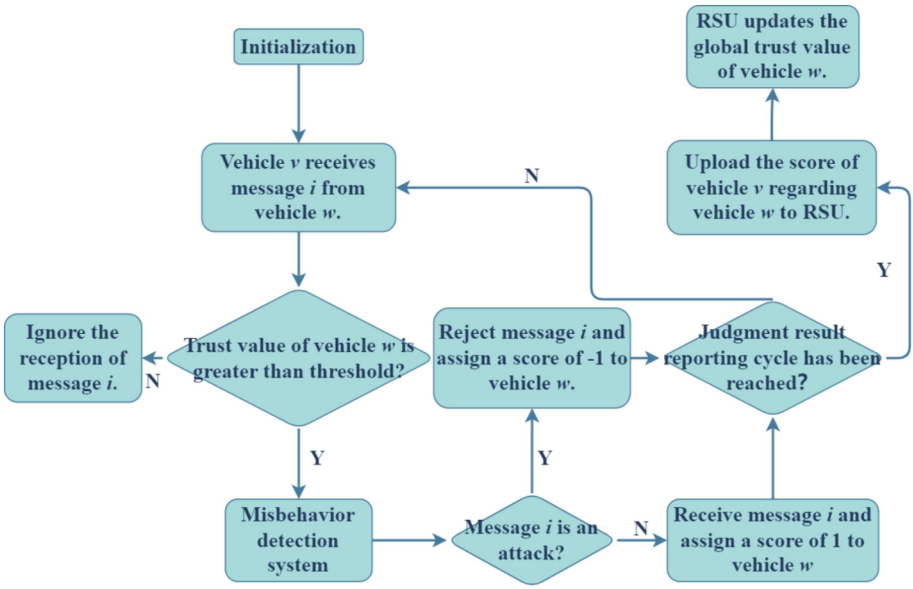


Fig. 6. Flow chart for vehicle w trust value update.

Trust Value Calculation: A python module for node trust value calculation is stored. The python module is integrated into the NS3 event schedule and is called every 2 s to calculate the trust value.

By following these steps, the NS3 module of our simulation platform enables the realistic simulation and evaluation of VANET, mobility patterns, trust value calculation, and other important aspects in the context of the VANET.

4.4 Python Modular

The python module serves as the core component responsible for implementing the trust management mechanism. It adopts the concept of federated learning, where a central neural network is deployed at the RSU and distributed to vehicles within its communication range. Each vehicle utilizes the neural network as its MDS. During each iteration, vehicles optimize the neural network parameters and upload them to the RSU. The RSU integrates these parameters and redistributes the updated neural network to the vehicles. Upon receiving the package, each vehicle performs MDS detection and conducts trust evaluation based on the detection results. The flowchart of the Python module is depicted in Fig. 7

The vehicle trust value is computed based on the information stored in the *CSV* file generated by the NS3 module. This file contains essential data, including the sending vehicle ID, receiving vehicle ID, sending time, receiving time, indication of whether it is an attack message, location information, and speed information. The overall flowchart of this module is shown in Fig. 7.

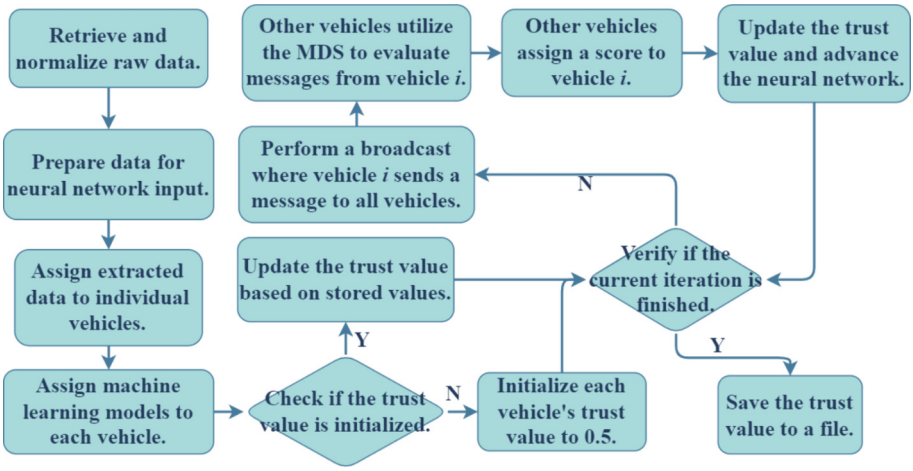


Fig. 7. Python module overall flowchart.

The process begins by reading and initializing the *CSV* dataset from the NS3 module. Relevant data such as sending time, receiving time, location, and speed information are normalized and used as training data for a neural network. The data is then allocated to each vehicle, distinguishing between malicious ($ID \leq 0$) and normal vehicles.

Malicious vehicles send false messages labeled as 1, while genuine messages are labeled as 0 in the NS3 module. These labels serve as supervision data for neural network training and accuracy assessment. Each vehicle is assigned its own copy of the neural network model, and the initial trust value is set to 0.5.

During the broadcast cycle, vehicles send messages to one another, and the receiving vehicles utilize their neural network to evaluate the messages. Calculation results, including the number of correct and incorrect messages, are generated using the scoring algorithm from Chapter III. The scoring results are stored accordingly.

The current neural network model parameters are used to update the overall neural network model, which is then redistributed to each vehicle. At the end of each cycle, vehicles broadcast their trust values to update the trust values of all vehicles, and the updated results are stored.

5 Simulations and Analysis

Trust management models rely on long-term cooperative behaviors, as negative actions can disrupt trust relationships. We compare proposed AATEA frameworks with Implicit Web of Trust in VANET (IWOT-V) [8] and three-factor based data-centric trust model (ERC-TM) [9], to validate the effectiveness of our program. IWOT-V consists primarily of two algorithms, namely BayesTrust and VehicleRank. The calculation of LTV is accomplished through BayesTrust,

Table 1. Simulation parameters.

Parameters Name	Value
Initial trust value	0.5
Number of vehicles	100
Adjustment parameter	3
Size of reference set	20
Radio Propagation Models	3GPP urban microcell
Simulation Scenario	Crossroad

which utilizes Bayesian statistics techniques. The calculation of GTV is achieved through VehicleRank, which constructs an implicit trust network and employs link analysis techniques. The trustworthiness of vehicles is dependent on their speed and energy consumption. ERC-TM utilizing expectation, risk and confidence to generate trust values. Nevertheless, this method occupies significant bandwidth, and reducing communication efficiency.

Furthermore, we evaluate the impact of integrating a trust management mechanism on Packet Received Rate(PRR) and latency in VANET. At last, we discussed the feasibility of applying AATEA to practical VANET, as well as the challenges posed by factors such as computational power, latency, scalability, and cost.

5.1 Simulation Parameters

We used a 2 km by 2 km OSM map in Beijing, China for our simulation, with a duration of 1000 s. The dynamic driving reference set consisted of the 20 closest vehicles to each vehicle. The received signal power was calculated using the 3GPP urban microcell propagation power loss model, which considers path loss, penetration loss, interference, multipath loss, and shadow fading to accurately estimate the received signal power. The detailed simulation parameter settings are shown in Table 1.

We chose IOW-T and ERC-TM as the comparison algorithms. The former is a Bayesian-based trust assessment method, which has shown excellent performance in recent years among Bayesian trust assessment methods. The latter is a three-factor-based trust assessment algorithm, highly efficient in discerning malicious vehicles.

5.2 Performance Comparison Without Attacks

In Fig. 8, we observe that in the case of continuous cooperative behavior, the trust value of node j in AATEA exhibits a slower rate of change compared to IWOT-V and ERC-TM, given the same initial conditions. However, in Fig. 9, when the change in trust value is associated with continuous noncooperative

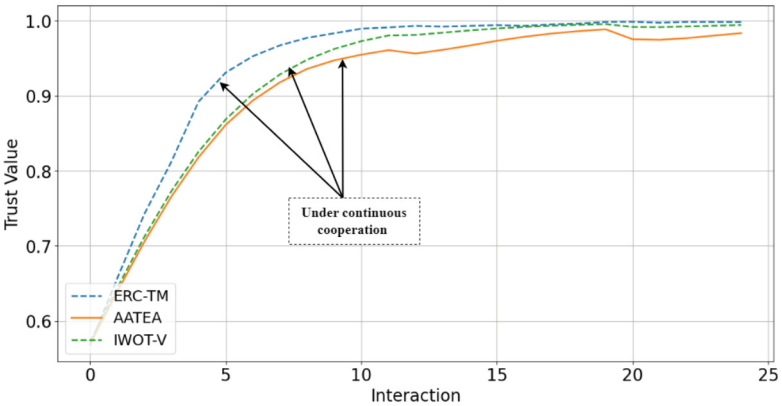


Fig. 8. Change of trust value between IWOT-V, ERC-TM and AATEA under continuous cooperation.

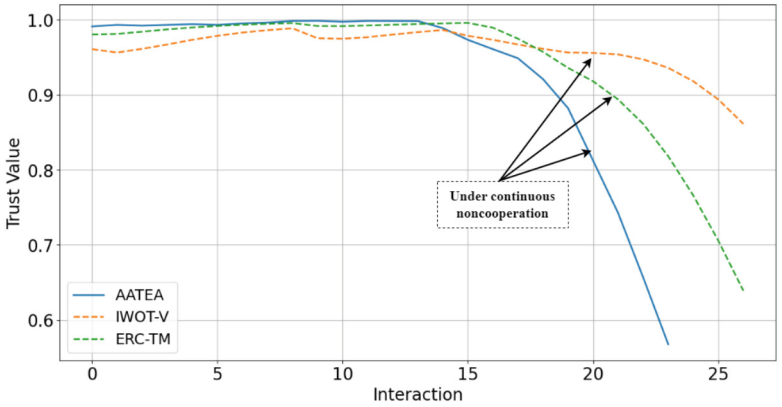


Fig. 9. Change of trust value between IWOT-V, ERC-TM and AATEA under continuous noncooperation.

behavior, the trust value in AATEA declines slightly faster. This characteristic of AATEA aligns with the desired "hard to get and easy to lose" nature of trust values.

Furthermore, it is crucial for trust values to promptly reflect changes in behavior, both in terms of continuous cooperation and noncooperation nodes. The comparison between IWOT-V, ERC-TM and AATEA highlights the significance of trust value dynamics in capturing behavioral changes within these models.

5.3 Performance Under On-Off Attack and Sybil Attack

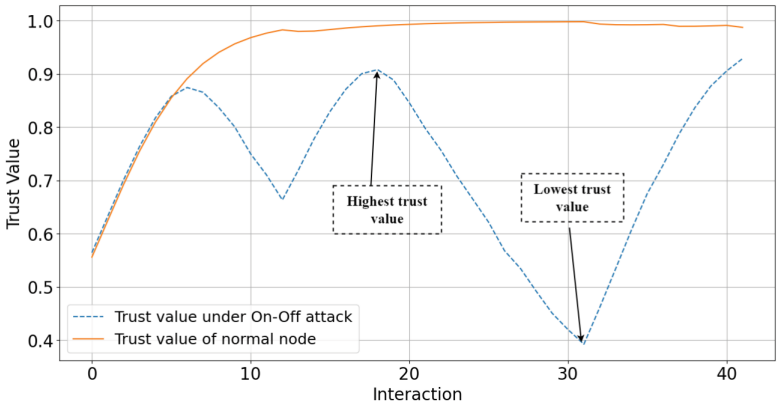
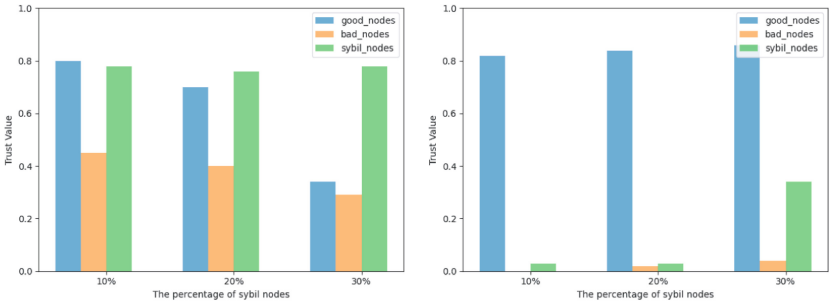


Fig. 10. Detecting on-off attack by AATEA. (Color figure online)

In Fig. 10, the blue line represents the trust value associated with a typical on-off attack. It shows that a compromised node can rapidly attain a significantly higher trust value, which then gradually declines over time. Moreover, the wireless nature of VANET introduces challenges in achieving error-free information transmission. This difficulty in distinguishing between malicious attack behaviors (such as tampering, replaying, forging, or discarding data packets) initiated by compromised nodes and natural interference during mobile interactions further complicates the trust evaluation process.

To overcome these challenges, we propose and simulate the AATEA architecture, which prioritizes low latency and real-time transmission. We assess the performance of AATEA in the presence of on-off attack, where one vehicle acts as the attacker for 50 rounds. Initially, the attacker cooperatively accumulates trust values for the first 5 rounds. However, in rounds 5 to 7, they abruptly switch to behaving as speeding vehicles before returning to their cooperative behavior.

By adjusting location factors, we effectively differentiate between good and bad vehicles. Figure 10 demonstrates the trust value of the on-off attacker after accumulating sufficient trust values over 10 rounds. In AATEA, the adaptive decay factor for on-off attacker is set to be small, resulting in a rapid decline in their trust values during attacks. Our scheme retains information about their past behavior, leading to a longer recovery period and a significant decrease in trust value compared to the initial attack, even after the attackers cease their attacks. After 15 rounds, the trust values of the on-off attacker consistently remain low, showcasing the effectiveness of our approach in safeguarding the trust model against on-off attackers.



(a) IWOT-V (b) AATEA

Fig. 11. The average global trust values of good nodes, bad nodes and sybil nodes in IWOT-V and AATEA under different percentage of sybil nodes.

Additionally, we compare the performance of AATEA and IWOT-V under sybil attack. We set the percentage of collusion nodes as 10%, 20% and 30% respectively to test the influence on IWOT-V and AATEA. Besides, we fix the number of bad nodes as 20. Figure 11a illustrates the average global trust values of good nodes, bad nodes, and collusion nodes in both IWOT-V and AATEA as the percentage of collusion nodes changes. In Fig. 11b, it is evident that in IWOT-V, the average global trust value of sybil nodes is consistently higher than that of other nodes, particularly when the percentage of sybil nodes exceeds 10%. This suggests that sybil nodes can accumulate relatively high trust values in IWOT-V, even when they are not actual vehicles. However, in AATEA, the average global trust value of sybil nodes is lower than that of legitimate vehicles. Although the trust value of sybil nodes increases significantly as the percentage of sybil nodes rises, it remains lower than the trust value of legitimate nodes. Notably, the trust value of malicious nodes remains close to zero across various percentages of sybil nodes, indicating that the presence of sybil nodes has no significant impact on malicious nodes. It is evident that AATEA can effectively defend against sybil attacks when the percentage of collusion nodes within the set of seed vehicles is below 30%.

5.4 The Impact of Trust Management System on VANET

Table 2 presents a comparison of the performance of our trust simulation platform with equipment that includes AATEA, IWOT-V, ERC-TM, and without any trust management mechanisms. Prior to integration, the platform achieved a PRR of 0.995507 and a Latency of 0.99236. After equipping AATEA, the PRR decreased to 0.99236, while the Latency increased to 16.3198. On the other hand, with the addition of IWO-V and ERC-TM, the PRR decreased to 0.98934 and 0.98347, respectively, while the latency increased to 18.46214 and 19.94231. We can conclude that AATEA has a relatively smaller impact on the vehicular ad hoc network system compared to IWO-V and REC-TM, resulting in lower

additional system overhead. Regarding the AATEA algorithm, despite a slight decrease in PRR and a marginal increase in latency, these changes are considered tolerable, given the overall improvement in the security of the VANET system. While the simulation results may differ from actual VANET, they demonstrate that our proposed AATEA can ensure real-time communication and meet the requirements of low latency and scalability in VANET.

Table 2. Comparison of PRR and Latency for VANET after trust management system integration.

Algorithm name	PRR	Latency
AATEA	0.99236	16.3198
IWOT-V	0.98934	18.46214
ERC-TM	0.98347	19.94231
Not equipped	0.99507	15.8829

6 Conclusion

In this paper, we propose AATEA, an advanced anti-attack trust evaluation algorithm utilizing a Bayesian network. The system utilizes Bayesian inference with adaptive decay factors to determine the global trust value. And, we introduced the concept of a dynamic trust redemption window, which dynamically adjusts its duration based on a vehicle's past performance. We also integrate forward and lateral driving reference sets to enhance vehicle resilience against attacks. Additionally, we develop a comprehensive C-V2X trust management simulation platform based on NS3 and SUMO, providing a realistic environment for evaluating trust-related algorithms. Through simulation experiments, we demonstrate the superiority of AATEA over RFSN and ERC-TM. It effectively captures "hard to gain and easy to lose" trust values, with higher accuracy in detecting on-off attack and sybil attack. We also discussed the feasibility of applying AATEA to real-world VANET. The results indicate that AATEA meets the requirements of low latency and real-time performance.

In future research, we aim to investigate our trust decision-making scheme further. We seek to establish a clearer relationship between trust management and application scenarios in VANET. Moreover, we will validate the deployment feasibility of AATEA in practical VANET.

Acknowledgment. This work is jointly supported by National Natural Science Foundation of China (Grant No. 62171061).

References

1. Panigrahy, S.K., Emany, H.: A survey and tutorial on network optimization for intelligent transport system using the Internet of Vehicles. Sensors **23**(1), 555 (2023)

2. Al-Shareeda, M.A., Manickam, S.: A systematic literature review on security of vehicular ad-hoc network (VANET) based on VEINS framework. IEEE Access **11**, 46218–46228 (2023)
3. Azam, F., Yadav, S.K., Priyadarshi, N., et al.: A comprehensive review of authentication schemes in vehicular ad-hoc network. IEEE Access **9**, 31309–31321 (2021)
4. Thumbur, G., Rao, G.S., Reddy, P.V., et al.: Efficient and secure certificateless aggregate signature-based authentication scheme for vehicular ad hoc networks. IEEE Internet Things J. **8**(3), 1908–1920 (2020)
5. Khan, S., Luo, F., Zhang, Z., et al.: Survey on issues and recent advances in vehicular public-key infrastructure (VPKI). IEEE Commun. Surv. Tutorials **24**(3), 1574–1601 (2022)
6. Zhang, J.: A survey on trust management for VANETs. In: 2011 IEEE International Conference on Advanced Information Networking and Applications, pp. 105–112 (2016)
7. Amari, H., Abou, E.Z., Khoukhi, L., et al.: Trust management in vehicular ad-hoc networks: extensive survey. IEEE Access **11**, 47659–47680 (2023)
8. Xiao, Y., Liu, Y.: BayesTrust and VehicleRank: constructing an implicit Web of trust in VANET. IEEE Trans. Veh. Technol. **68**(3), 2850–2864 (2019)
9. Zhang, S., He, R., Xiao, Y., et al.: A three-factor based trust model for anonymous bacon message in VANETs. IEEE Trans. Veh. Technol. **72**(9), 11304–11317 (2023)
10. Li, F., Guo, Z., Zhang, C., et al.: ATM: an active-detection trust mechanism for VANETs based on blockchain. IEEE Trans. Veh. Technol. **70**(5), 4011–4021 (2021)
11. Rehman, A., Hassan, M.F., Hooi, Y.K., et al.: CTMF: context-aware trust management framework for Internet of Vehicles. IEEE Access **10**, 73685–73701 (2022)
12. Hasrouny, H., Samhat, A.E., Bassil, C., et al.: Misbehavior detection and efficient revocation within VANET. J. Inf. Security Appl. **46**, 193–209 (2019)
13. Ganeriwal, S., Balzano, L.K., Srivastava, M.B.: Reputation-based framework for high integrity sensor networks. ACM Trans. Sens. Netw. (TOSN) **4**(3), 1–37 (2008)
14. Li, B., Liang, R., Zhu, D., et al.: Blockchain-based trust management model for location privacy preserving in VANET. IEEE Trans. Intell. Transp. Syst. **22**(6), 3765–3775 (2020)