AIT: An AI-Enabled Trust Management System for Vehicular Networks Using Blockchain Technology

Chenyue Zhang, Wenjia Li, *Senior Member, IEEE*, Yuansheng Luo, and Yupeng Hu, *Senior Member, IEEE*

***Abstract*—Currently, connected vehicles have gradually stepped into our daily lives, and they generally rely on vehicular networks to generate and exchange traffic-related messages to improve the overall travel safety and efficiency. However, due to the open nature of vehicular networks, these traffic-related messages could be erroneous, which may be caused by various reasons, ranging from an onboard device (OBD) sensor malfunctioning and reporting incorrect reading to the message being tampered by a malicious vehicle. To address these rapidly increasing security challenges, we have proposed an AI-enabled trust management system (AIT) in this article, which is an AI-enabled trust management system for vehicular networks using the blockchain technique. In the AIT system, each vehicle first senses, generates, and exchanges messages with other vehicles. These messages then get validated by the neighboring vehicles. As vehicles receive and validate messages from other nearby vehicles, they will establish and manage the trust of those nearby vehicles, which is enabled by utilizing the deep learning algorithm. Once a vehicle identifies untrustworthy vehicles, it reports them to the nearby roadside unit (RSU), and the RSU will validate the authenticity of the report as well as the identity of the vehicle by using the emerging blockchain technique. The security credentials of untrustworthy vehicles will then be revoked by the RSU. We have conducted an extensive experimental study to evaluate the AIT system. Simulation results clearly indicate that AIT performs better than existing approaches and can manage the trust of vehicles and detect malicious ones in an accurate and efficient manner.**

***Index Terms*—Blockchain, deep learning, security, trust management, vehicular networks.**

I. Introduction

IN THE past few years, we have witnessed an explosive growth to the number of connected and/or autonomous vehicles on the road. News articles revealed that there were already one million connected vehicles made by general motors (GMs) on the road in 2015 [1], and Tesla's CEO Elon Musk estimated in 2019 that over a million of self-driving taxi cabs would be on the road by the end of 2020 [2]. To enhance the safety and efficiency of these connected and autonomous vehicles, it has become essential that they could sense and collect various travel related information and exchange them via vehicular networks, which is enabled by the increasing onboard sensing, computation, and communication capabilities. In vehicular networks, different types of nodes, including vehicles and road-side units (RSUs), could communicate and share information with each other through vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communications. The information that they share generally contains alerts and updates regarding road accidents, traffic conditions (such as congestion, construction, and hazardous weather conditions), and other related transportation events. All these traffic alerts and updates could make vehicles aware of various traffic conditions in a timely manner, thus improving the transportation safety and efficiency.

Despite that vehicular networks may help disseminate these types of important information, caution should still be taken when interpreting and utilizing them, as you generally have little idea whom you are communicating with in a neighboring vehicle because of its high mobility. The situation further deteriorates when there are malicious vehicles that have been compromised and controlled by adversaries, who may deliberately share fake traffic alerts and updates to confuse other vehicles. An example of the fake traffic update shared by a malicious vehicle is depicted in Fig. 1.

From Fig. 1(a), it shows that a malicious vehicle shares a fake traffic update stating that there is no traffic congestion, while there are some road hazards, such as traffic congestion in reality. As shown in Fig. 1(b), the fake traffic update that is disseminated via vehicular networks makes the road even more congested, which clearly demonstrates that the fake traffic updates and alerts could greatly jeopardize the safety and efficiency of the transportation system. As a result, it is critical to properly evaluate the trustworthiness of both traffic-related messages and vehicles, which share them in vehicular networks.

In recent years, various research efforts have been made to develop effective trust management systems for wireless networks [3]–[5], including vehicular networks [6]–[8], sensor networks [9]–[11], and Internet of Things [12]–[14]. While they primarily focused on evaluating the trust of nodes (such as vehicles, sensors, etc.), there have been some research works in which the trust of data that the nodes generate is also evaluated [7], [15], [16]. In the specific context of vehicular

Manuscript received May 11, 2020; revised September 1, 2020 and November 3, 2020; accepted November 30, 2020. Date of publication December 14, 2020; date of current version February 19, 2021. *(Corresponding author: Wenjia Li.)*

Chenyue Zhang and Wenjia Li are with the Department of Computer Science, New York Institute of Technology, New York, NY 10023 USA (e-mail: wli20@nyit.edu).

Yuansheng Luo is with the School of Computer and Communication Engineering, Changsha University of Science and Technology, Changsha 410114, China.

Yupeng Hu is with the College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China.

Digital Object Identifier 10.1109/JIOT.2020.3044296

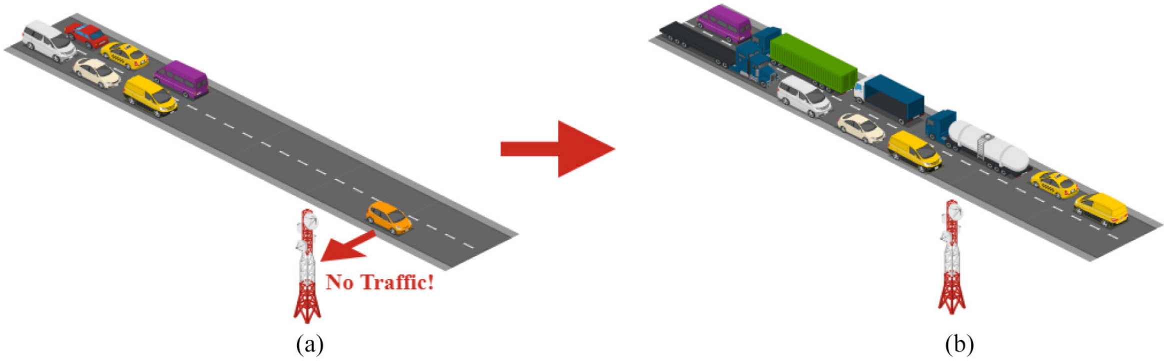


Fig. 1. Example of fake traffic update shared by a malicious vehicle and its outcome in vehicular networks. (a) Malicious vehicle sends a fake traffic update. (b) Outcome of the fake traffic update: severe traffic congestion.

networks, all the vehicles normally travel constantly at a relatively high speed, resulting in a rapidly changing network topology. Consequently, it is challenging for a vehicle to evaluate the trust of all other vehicles that it interacts with in a timely fashion. Moreover, due to its highly mobile and dynamic nature, vehicular networks usually generate a large amount of data, such as traffic alerts and updates reported by both vehicles and RSUs. However, the reliability of these data is weakened by both imperfect reporting devices and malicious vehicles that have been compromised by adversaries, which further undermines the trust management system that largely depends on the data to properly function. Hence, it is essential yet challenging to design effective trust management systems for vehicular networks.

To address this urgent need, we have proposed an AI-enabled trust management system for vehicular networks, namely, AIT, in this article. In contrast to the existing trust management systems, such as [3] and [8]–[10], the AIT system does not rely on a fixed formula to evaluate the trust ratings of individual messages or calculate the overall trust of vehicles. Instead, the emerging deep learning algorithm is used to determine the trust of vehicles in an automatic manner.

The main contributions of this research work are as follows.

1) We proposed a novel trust management system that is based on deep learning to evaluate the trust of nodes (including both vehicles and RSUs) as well as data (such as messages) in an automatic and dynamic manner.
2) We applied the emerging blockchain technology to the trust management system so that both the identity of both vehicles and RSUs and the authenticity of messages sent in the vehicular networks could be validated, thereby remarkably enhancing the security of vehicular networks.
3) To validate and evaluate the performance of the AIT system, we have conducted extensive simulations, and the experimental results clearly show that the proposed trust management system could evaluate the trust in an accurate and efficient manner.

The remainder of this article is organized as follows. Section II summarizes the existing research findings that are related to this work. The research problem is defined in Section III, which contains details regarding both system model and adversary model. Section IV looks at the detailed design of our AIT system. Section V provides an in-depth discussion on the experiments that we have conducted to validate the AIT system. Finally, Section VI concludes this article.

II. Related Work

Due to the widespread and increasing security threats that recently arise in vehicular networks, many research efforts have been made to detect and mitigate them. Researchers have mainly been focusing on developing two categories of approaches in response to those security threats: 1) trust management and 2) malicious node detection.

A. *Trust Management System*

In recent years, trust and reputation management has attracted great attention from the research community.

Guo *et al.* [17] proposed a trust management model that is aware of the context and able to authenticate the message received by evaluating the trust value of the sender. In this approach, the trust is decided by both the related information that is available and the current evaluation strategy. In addition, a reinforcement learning model was developed so that vehicles could dynamically set the evaluation strategy in different driving scenarios.

El Sayed *et al.* [18] studied a hierarchical trust management system for vehicular networks, which performs trust computation for each vehicle. More specifically, the proposed system takes the steps of trust evaluation, trust propagation, and trust aggregation so that the trustworthiness of vehicles could be accurately derived.

In [19], an anti-attack trust management scheme named AATMS was proposed by Zhang *et al.* In AATMS, the local trust will first be computed, which is used as the criterion to determine a set of trustworthy seed vehicles. The trustworthy seed vehicles are then used together with the local trust link to evaluate the global trust of all vehicles.

Li and Song proposed ART in [7], and its primary focus was to ensure that the trust could be evaluated in an efficient and accurate manner in the presence of adversaries. Thus, the proposed attack-resistant trust management scheme has the functionality of both malicious node detection and trust management. Moreover, the trustworthiness of both data and vehicles is evaluated by the ART scheme.

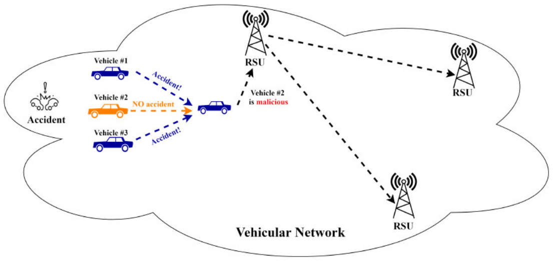


Fig. 2. System model for the AIT in vehicular networks.

Ahmed *et al.* [20] proposed a trust management model for connected vehicles, which is resilient to the well-known man-in-the-middle (MiTM) attack. In this model, dishonest vehicles which perform MiTM attacks could effectively be detected and their security credentials would then be revoked.

Yang *et al.* [8] proposed to use the blockchain to help better evaluate the trustworthiness of vehicles. In the proposed scheme, vehicles first perform message validation using the Bayesian inference model. As the next step, the vehicle will generate a rating for each vehicle that originates the message. Then, the RSUs perform trust evaluation on the vehicles. Based on these ratings, they will add the trust offsets as a block to the blockchain, which is maintained by all the RSUs.

In [21], a hierarchical blockchain structure is studied for managing reputation in vehicular networks. The first layer of the blockchain hierarchy is the vehicles that are located within the same district, and they record reputation values for their own district. Through voting, vehicles can become a miner in the vehicular network. Once a block is generated by the miner, it will be sent over to the RSU that is in the same district. Then, the RSU sends this block to every vehicle in the same district. RSU just sends blocks but they are not responsible for storing the blocks and managing the blockchain. The blocks are stored in each vehicle instead. The second layer covers a district in the first layer along with its neighboring areas.

Although the blockchain technology is used in both [8] and [21] to manage trust for vehicular networks, the proposed AIT system has some fundamental uniqueness and differences when compared to them. First, as described in [8], all the RSU maintain one single blockchain for all vehicles. As a result, it is costly in terms of both time and computational power to locate and mine a block for a specific vehicle because all the blocks for all vehicles are in the same blockchain. As for the hierarchical blockchain structure proposed in [21], each vehicle maintains its own blockchain, which makes the blockchain more vulnerable for data tampering and loss. Unlike these two previous research efforts, we have designed and implemented the distributed blockchain based on the tree structure that is maintained by each RSU, and each branch of the tree corresponds to a vehicle within the RSU's direct communication range. Moreover, we use the Merkle value to prevent data tampering and loss. In this way, the overall security and efficiency of the blockchain in the AIT system is further enhanced.

In addition, there have also been some other research efforts that aim at trust and reputation management for vehicular and other types of wireless networks [13], [22], [23].

B. Malicious Node Detection

The other category of research works that could help better secure vehicular networks is malicious node detection, which aims at detecting and possibly mitigating malicious nodes.

Gu *et al.* proposed a method to detect malicious nodes in fog computing-based VANETs [24], in which the fog server identifies suspicious vehicles via trust evaluation based on the correlation of data and network topology.

In [25], a machine learning-based misbehavior detection system was proposed for vehicular networks, which aimed at detecting and coping with insider attacks.

Chen *et al.* [26] studied a secure message dissemination scheme for vehicular networks in the presence of malicious vehicles. More specifically, the proposed scheme could identify true message(s) when the messages received from multiple nearby vehicles are conflicting, which was achieved by incorporating the underlying network topology information.

In [27], a security framework was proposed for VANETs, in which the SVM algorithm was used to automatically distinguish malicious vehicles from benign ones. Moreover, the proposed approach was aware of the context in which the misbehaviors occurred. By this means, the detection accuracy of malicious vehicles was further improved.

Sedjelmaci *et al.* [28] proposed a misbehavior detection and prevention scheme for vehicular networks, and the misbehavior prediction is based on game theory so that the occurrence of malicious vehicles could be prevented.

III. Problem Definition

In this section, we describe the research problem that is being investigated in this article in more detail. More specifically, we are defining both the network model and the adversary model in a detailed fashion.

A. System Model

As depicted in Fig. 2, the trust management system in vehicular networks is generally composed of the following two types of nodes: 1) RSUs and 2) vehicles.

1) *RSU:* RSUs generally possess large amounts of processing, storage, and communication capacities, and they are stationary. They can keep track of all current vehicle trust levels, based on messages sent by vehicles about other vehicles and the current traffic conditions. They have a long communication range that covers a large section of roads, which allows them to receive messages from other RSUs and vehicles from a far distance. After they receive messages, the messages are analyzed by them. After evaluating the information received, the RSUs will adjust all vehicles' trust values so that they can be sent to other RSUs. Because different RSUs could get different messages from different vehicles about suspicious activity, their evaluation of vehicle trust may be different. The trust values from all RSUs regarding the same vehicle will be averaged to determine if a vehicle is malicious. Once found to be malicious, the vehicle's public key will be revoked, blocking them from sending new messages in the network. In the meanwhile, if they

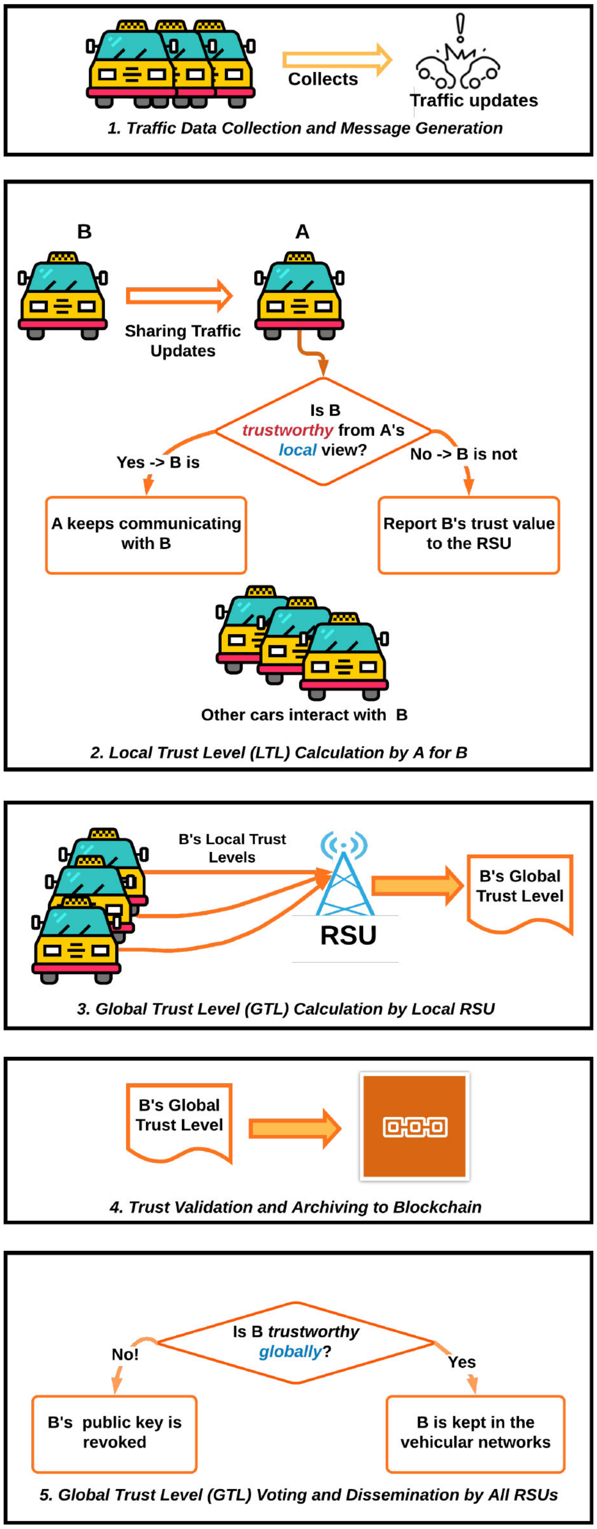

Collects
Traffic updates
1. Traffic Data Collection and Message Generation
B
A
Sharing Traffic Updates
Is B trustworthy from A's local view?
Yes -> B is
No -> B is not
A keeps communicating with B
Report B's trust value to the RSU
Other cars interact with B
2. Local Trust Level (LTL) Calculation by A for B
B's Local Trust Levels
RSU
B's Global Trust Level
3. Global Trust Level (GTL) Calculation by Local RSU
B's Global Trust Level
4. Trust Validation and Archiving to Blockchain
Is B trustworthy globally?
No!
Yes
B's public key is revoked
B is kept in the vehicular networks
5. Global Trust Level (GTL) Voting and Dissemination by All RSUs

Fig. 3. Overall workflow of the AIT system.

want to regain the public key, they will send a request to RSU and the RSU will evaluate its trustworthiness to decide whether to give the vehicle a new public key.

2) *Vehicle:* Vehicles have limited processing, storage, and communication capacities when compared to RSUs, and they are moving constantly in the road network. Each vehicle is represented as a node in the vehicular network. Vehicles could communicate with other vehicles by sending messages, and they pass information to each other to provide current road condition updates. If a hazard is spotted, a vehicle can pass the location and type of the hazard to other vehicles or RSUs. Each time when a vehicle gets a communication request from another vehicle, a trust evaluation is done on the other vehicle to make sure it is not malicious prior to the communication. If the other vehicle is detected to be malicious, the communication request will be denied, and a message is sent to the nearest RSU to report this malicious vehicle.

B. Adversary Model

In this research, we assume that both vehicles and RSUs could be compromised by adversaries, and they will then exhibit various malicious behaviors, such as intentionally sharing a fake traffic alert or falsely reporting another vehicle to be malicious, etc. More specifically, the following three types of malicious attacks are considered in this article.

1) *Simple Attack (SA):* The primary goal of malicious attackers when performing SA is to interfere with the messaging service in vehicular networks by different means, such as sending out an excessive number of messages, so that other benign nodes cannot send out any message successfully during this time. In addition, the compromised nodes may choose to drop or alter the incoming messages so that the current traffic status will be distorted when being shared with other nodes.
2) *Bad Mouth Attack (BMA):* In BMA, malicious attackers deliberately share fake trust ratings for other nodes, such as claiming that a benign vehicle is malicious and *vice versa*. By this means, the security credentials of benign vehicles could potentially be revoked, and the vehicular networks may be dominated by malicious vehicles.
3) *Zigzag Attack (ZA):* The advanced attackers may attempt to avoid being detected by conducting the malicious behaviors in an intermittent manner. For example, an attacker may choose to spoof the incoming messages for some time, then stop for a while before switching to conducting the BMA. Given that each attack is occurring at a lower frequency, we would envision that the ZA (also known as the on-and-off attack) is the most challenging attack to be caught by the trust management system.

IV. Detailed Design of the AIT System for Vehicular Networks

In this section, we discuss the proposed AIT system in detail. The overall workflow of the AIT system is shown in Fig. 3.

As depicted in Fig. 3, there are five important steps in the AIT system, namely: 1) traffic data collection and message generation; 2) local trust level (LTL) calculation and sharing with local RSU; 3) global trust level (GTL) calculation by local RSU; 4) trust validation and archiving by blockchain; and 5) GTL voting and dissemination by all the RSUs.

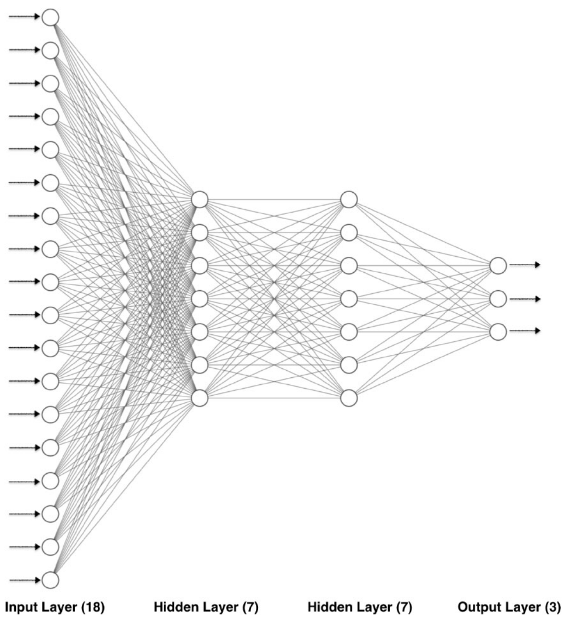


Fig. 4. Structure of the feedforward neural network.

A. Traffic Data Collection and Message Generation

As the first step of the trust management process, each vehicle in the vehicular networks collects the nearby traffic updates and then generates messages that best describe the traffic updates. Given that there may be faulty or even malicious vehicles that could share misleading messages, it is important to take different factors into consideration, such as the distance between vehicles that reports the same traffic update, the number of vehicles in the range where the traffic incident is visible, the location of the traffic incident, the location of the nearest RSU, the communication range that RSU can cover, the direction of the traffic, the location of each vehicle that reports the incident, and so on. All these data may be integrated into the message so that the recipient could better identify the context in which the traffic incident is reported.

B. Local Trust Levels Calculation and Sharing With Local RSU

Once a vehicle receives messages from a nearby vehicle regarding some traffic updates, it will calculate the LTL of the message sender by applying the feedforward neural network algorithm [29], which is depicted in Fig. 4.

As shown in Fig. 4, the input layer is composed of 18 input nodes, such as the location of the reporting vehicle, the location of the traffic incident, the distance that the traffic incident is visible, the current trust rating of the reporting vehicle, and so on. The complete list of the eighteen input data is shown in Table I.

There are two hidden layers with totally 14 hidden neurons. The output layer produces the following three outputs: 1) updated trust rating of the reporting vehicle; 2) prediction of next message time; and 3) level of need for message relay as the output, which are shown in Table II. Note that these three outputs of the feedforward neural network algorithm will be stored in the body of a block in the blockchain.

TABLE I
INPUT OF THE FEEDFORWARD NEURAL NETWORK ALGORITHM

Input	*Note*
location of the reporting vehicle	the x-coordinate
location of the reporting vehicle	the y-coordinate
location of the traffic incident	the x-coordinate
location of the traffic incident	the y-coordinate
location of the receiving vehicle	the x-coordinate
location of the receiving vehicle	the y-coordinate
distance that the traffic incident is visible	based on type of traffic incident, weather, traveling direction of reporting vehicle, and visibility map
current trust rating of the reporting vehicle	the latest reported and calculated trust value
current trust rating of the receiving vehicle	the latest reported and calculated trust value
nearest RSU to the reporting vehicle	the ID of the trusted RSU that is closest to the reporting vehicle
nearest RSU to the receiving vehicle	the ID of the trusted RSU that is closest to the receiving vehicle
type of traffic incident	an integer representing the traffic incident type
traveling direction of the reporting vehicle	represented as a real number $r \in [0, 360)$, where 0, 90, 180, and 270 represents east, north, west, and south, respectively
traveling speed of the reporting vehicle	speed in km/h
traveling direction of the receiving vehicle	represented as a real number $r \in [0, 360)$, where 0, 90, 180, and 270 represents east, north, west, and south, respectively
traveling speed of the receiving vehicle	speed in km/h
elapsed time since last message received from the reporting vehicle	in seconds, (0 means the elapsed time is unavailable/unknown)
Current vehicle mobility level	a real number between 0 and 1 that is used to adjust local speed limit based on a combination of visibility and road conditions. 0 means no visibility or road blocked, with 1 meaning full visibility and road usable at posted speed limit.

TABLE II
OUTPUT OF THE FEEDFORWARD NEURAL NETWORK ALGORITHM

Output	*Note*
(1) change in trust value for the reporting vehicle	
(2) projected time interval to receive next message from the reporting vehicle	0 means unknown
(3) evaluated importance of message	a real number between 0 and 1 to indicate the importance of message, with 0 being least important and 1 meaning most important

It is important to learn which weights and biases that minimize a certain cost function. In the following functions [31], the activation of the last neuron is represented as a, with a superscript L, which represents the current layer. Then, the activation of the previous layer is $a^{(L-1)}$. Cost can be calculated from finding the differences to each neuron from the output it is expected to give minus the current output value. Then, add up the square of those differences between each component onto all the layers in a neural network. Equation (1) describes how the cost in a neural network is computed. Note that y_j stands for the jth output in the output layer. Suppose there are totally L hidden layers, then $n_L - 1$ represents the neuron in the last hidden layer right before the output layer

$$C_0 = \sum_{j=0}^{n_L-1} \left(a_j^L - y_j\right)^2. \tag{1}$$

We then calculate the cost for all layers and then average them, so that the total cost of the neural network is calculated. The derivative of the full cost function is equal to the average of all training examples, which is shown in the following equation:

$$\frac{\partial C}{\partial \omega^L} = \frac{1}{n} \sum_{k=0}^{n-1} \frac{\partial C_k}{\partial \omega^L}. \tag{2}$$

As the next step, the square of the difference we calculated previously is summed up, and then we average the result, which gives us the total cost of the network. Then, we can adjust the weight and differences of the current neurons accordingly. It is important to look for gradients in this cost function because it implies how to change all the weights and biases of all the connections between neurons. In the gradient vector, each element is showing how sensitive the cost function is to each weight and bias. It explains how to get weight and biases for a gradient descent.

Backpropagation is an algorithm to compute the gradient that is used to minimize the cost and biases. Each component of the cost function shows how sensitive the cost function is to each weight and biases. ω represents the weight

$$a^L = \sigma\left(\omega_0^{L-1} a_0^{L-1} + \omega_1^{L-1} a_1^{L-1} + \cdots + \omega_{n-1}^{L-1} a_{n-1}^{L-1} + b\right). \tag{3}$$

For the neuron a, which lies in layer L, is defined as a^L. N stands for the total number of nodes in a layer. The weighted sum is calculated from each individual weights times the neuron value that is $\omega_0^{(L-1)} a_0^{(L-1)}$. a^L is calculated from a certain weighted sum of all the activations in the previous layer, plus b, which is a bias. They are plugged into a sigmoid function, which is represented as σ.

From the equation, there are three elements that can be changed so as to change the value of a^L, and they are weights, biases, and the value of the previous node. The following (4) contains all of these three elements of the weighted sum and is called z^L, which represents the total weighted sum regarding the weights of the activation of the previous layer plus a bias of the current layer

$$z^L = \omega^L a^{(L-1)} + b^L. \tag{4}$$

It is important to figure out how sensitive the changes of weight will influence the cost function. That implies the key is to figure out what is the derivative of C in regarding to ω^L, which is $(\partial C / \partial \omega^L)$

$$\frac{\partial C_0}{\partial \omega^L} = \sum_{j=0}^{n_L-1} \frac{\partial z^L}{\partial \omega^L} \frac{\partial a^L}{\partial z^L} \frac{\partial C_0}{\partial a^L}. \tag{5}$$

This chain rule function describes how sensitive the cost is to a specific weight. It implies the sensitivity of the cost regarding small changes of weights, even slight changes can cause a huge impact. The goal is to figure out how sensitive the cost function is too small changes in our weight ω^L. After calculating how sensitive the cost function is to the activations, processes are repeated for all the weights and biases feeding into the layer. The second term in the chain rule is also called a sigmoid function, which is a critical component of artificial intelligence. With the help of sigmoid function, most of the functions can be predicted and estimated.

The sigmoid function can be both amplified and shifted. By changing the gains of the hidden layers, functions can be amplified while through changing the constant input or bias, functions can be shifted. With the help of multiple Sigmoid functions, by amplifying and shifting, we can then add them together and deliver a good approximate function that is estimated and used in the neural network. In addition, the backpropagation algorithm is effective in calculating the derivative on weights.

By applying the feedforward neural network algorithm, the trust management system can successfully identify which vehicles are malicious and learn the potential correlation among those malicious vehicles. As a result, the LTL of the reporting vehicle will be calculated and shared with the RSU after this process.

C. Global Trust Levels Calculation by Local RSU

Each vehicle's behaviors may vary over time, especially if the malicious vehicle is adopting the ZA as its attack pattern, in which the malicious vehicle can show different malicious behaviors at different times with some intervals in between. In this case, an instantaneous local trust view could be inaccurate and misleading if the malicious vehicle carefully spreads its activities over many neighboring vehicles at different times.

Thus, as the next step, the GTLs of all vehicles will be calculated by RSUs. The trust of each message coming into the RSU is also calculated with the same deep learning algorithm as the vehicles use for their local trust evaluation. The result of the GTL calculation is an adjustment to the initial LTL for each individual vehicle. This process produces a more accurate and comprehensive view of each vehicle's trust level, which can help better protect vehicular networks against those malicious behaviors. Suppose the RSU r_1 receives an LTL update of vehicle v_i from the reporting vehicle v_j, the evaluation process of GTL is shown in Algorithm 1. Note that the change in trust rating (ΔT_i) corresponds to the output (1) of the feedforward neural network.

D. Trust Validation and Archiving by Blockchain

Once the GTLs for all the vehicles are calculated by the local RSU, the identity of each vehicle is first validated by

Algorithm 1: GTL Update

```
Input: VehicleID of $v_i$, $\Delta T_i$
Output: Updated trust rating of $v_i$ ($T'_i$)
if VehicleID is valid and $\Delta T_i \neq 0$ then
    $T'_i \leftarrow T_i + \Delta T_i$;
else
    stop the update process;
end
if $T'_i \geq 0.5$ then
    trust validation for $v_i$ succeeds;
else
    trust validation for $v_i$ fails;
end
```

the blockchain technology, and the GTL will be encoded and added as a new block to the blockchain. This provides for an unbreakable chain of GTLs. Without blockchain, fake trust levels can be distributed by malicious RSUs.

More specifically, RSU keeps track of all the transaction history from all the vehicles that are located within the direct communication range of the RSU by using the Merkle root hash value. In addition, the blockchain maintained in each RSU is built by following the tree structure. The root node r of this tree represents the RSU, and its immediate child nodes $v_i (i = 0, 1, 2, \ldots, n)$ represent the vehicles that are located within that RSU's direct communication range. Therefore, the number of immediate child nodes for a specific RSU should be equal to the number of vehicles that are in its direct communication range. Each v_i corresponds to a blockchain for the corresponding vehicle. Fig. 5 demonstrates how the blockchain technology is applied to maintain all the trust ratings in the AIT system.

Unlike a recent research effort in this domain [21], in which vehicles maintain blockchain and store all the blocks, which is unsafe and inefficient, in our approach each RSU manages the blockchain for all the vehicles in its direct communication range, and it only provides information to vehicles as needed. Vehicles only keeps the Merkle root, while the RSU keeps all the blocks. By this means, the resource intensive tasks, such as creation and mining of blocks, are only performed on the RSUs instead of vehicles, which could improve the scalability of vehicular networks.

Moreover, the Merkle root, which is a hash value generated by the hash function, serves as a stub between vehicles and RSU to protect the blocks from malicious attacks from the RSU itself and other malicious vehicles that attempt to change the blocks. After a vehicle v_i generates a new message, it will be sent to the RSU r to generate a block and then be attached to the end of the blockchain that corresponds to v_i in the tree structure. Every time with a newly added block, the merkle value that corresponds to each vehicle in this RSU's tree structure will be updated. The hash will be the new stub as the transaction continues. As a result, both v_i and r contain the up to date Merkle value. In the future, before every transaction, v_i will send a request to r asking for its current Merkle root value. If the hash values match, then the transaction is validated and becomes successful. Otherwise, the validation fails, and the blockchain has to be resynchronized by backtracking previous transactions until the hashes match, which is achieved by both v_i and r.

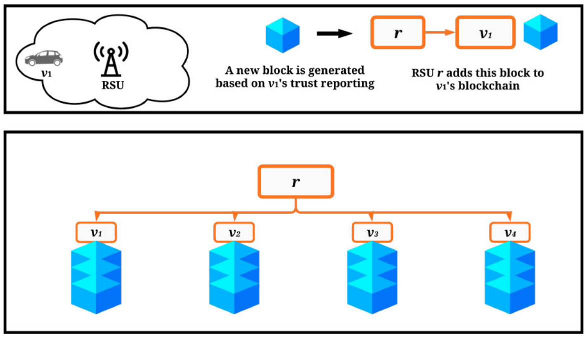

v₁
RSU
A new block is generated based on v₁'s trust reporting
r
v₁
RSU r adds this block to v₁'s blockchain
r
v₁
v₂
v₃
v₄

Fig. 5. How RSU manages blockchain for the neighboring vehicles.

As shown in Fig. 5, each RSU uses a tree structure to store the blockchains for all the neighboring vehicles, in which each branch of the tree corresponds to one vehicle's blockchain. As a result, the computational complexity to locate each vehicle's blockchain will be $O(1)$. Therefore, the overall computational complexity to use blockchain to validate the hash values will be $O(n)$.

E. Global Trust-Level Voting and Dissemination by All the RSUs

In Section IV-D, each RSU calculates GTLs for all the vehicles in its transmission range using information collected from all other vehicles in that range. In general, the composition of vehicles is different for each RSU, because of transmission range limitation. Therefore, GTL for the same vehicle could be different when being calculated by different RSUs. To address this issue, all RSUs will share the GTLs they have calculated for vehicles that are in their communication range at a predefined time interval. The average of all the trust levels for each vehicle is then calculated and used as its GTL. This new GTL will also be encoded and added to the blockchain as a new block. Different evaluation time durations could be used in practice because short-term trust may be okay for less sensitive messages, such as temperature, but a longer term of trust is required for more critical messages, such as traffic congestion or accidents. The new GTLs are distributed as links to the blockchain so that the GTL cannot be spoofed by a malicious RSU anymore.

When the trust level of a vehicle drops below 0.5, it is considered untrustworthy, and any message that it sends out will be ignored by other vehicles and RSUs for forwarding. However, even if a vehicle is untrustworthy, its messages are still evaluated for trust level update purpose. If its trust level rises above 0.5 later, it becomes trustworthy again. The same principle applies to RSUs: they are also dynamically evaluated of their trust levels. If the trust level of a RSU falls below 0.5, then it will be viewed as untrustworthy, and its messages will be ignored. It could only become trustworthy later again when its trust level rises above 0.5, which could be simply decided by a single vehicle or RSU.

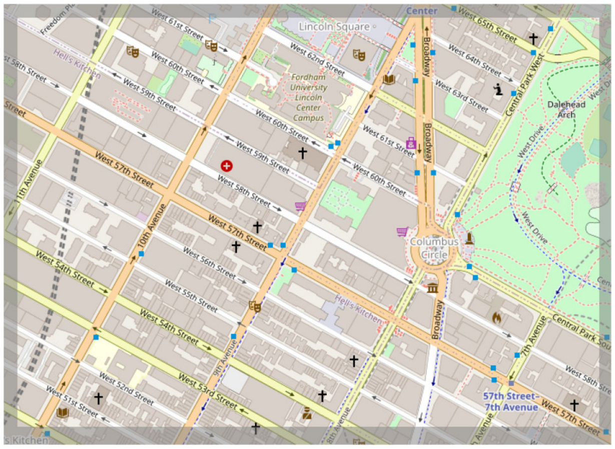


Fig. 6. Map generated by SUMO, which represents part of the New York city road network.

Unlike vehicles, if an RSU is compromised by the adversary and considered as malicious after trust evaluation by a vehicle or another RSU, the untrustworthy RSU will be reported to other RSUs. The other RSUs have the authority to make the final decision on whether this RSU is malicious or not based on their prior observations and knowledge on the compromised RSU. Once the compromised RSU is confirmed as untrustworthy by other RSUs, its security credentials will be revoked and it will no longer be able to participate in vehicular networks. All the RSUs have to decide whether or not to allow an RSU rejoin the network after its trust level rises above 0.5 again. By this means, we could ensure that RSUs always maintain higher security requirements when compared to vehicles.

V. Experimental Study

In this section, we provide a detailed discussion on how we set up the simulation for the experimental study, and also the experimental results based on the network simulation.

A. Real-Time Virtual Map Simulation

In this article, we use simulator of urban mobility (SUMO) [30] to download and generate maps for our vehicular network simulation using its Web Wizard utility. SUMO is a widely used road traffic simulation tool, which is designed to handle large road networks. SUMO uses the downloaded road information to generate vehicles that are running on the downloaded map with different directions, speeds, and initial locations. The data set generated by SUMO provides a good simulation that mimics what is happening in the real world. Fig. 6 shows a map that is downloaded and generated by SUMO, which represents the part of the road networks in New York city. The size of the map is 602 by 622 m.

Using this map data generated from SUMO, a real-time vehicular network simulation can be performed. The data generated from SUMO last for 900 s, which includes the information for the location, ID, and velocity of vehicles. In the raw simulation data from SUMO, vehicles could enter and leave the simulation area during the simulation, which makes it more difficult to locate the malicious vehicles that may only be in the simulation area for a short period of time during the simulation. To address this issue, we adjusted the SUMO data to keep the vehicles from leaving the simulation area: if a vehicle is about to leave the simulation area, its direction of travel is reversed, and it will travel back to where it started. The network simulations were set up by placing 50, 100, and 200 vehicles in the simulation area during the simulation runs. The raw SUMO simulation uses 900 vehicles to generate a training set of 20 000 inputs/outputs. Then, a simulation was run using SUMO and all the data generated was outputted to a .xml file containing each individual vehicle's information, such as real-time location and speed. The data used are split to 60% training, 20% generalization, and 20% validation.

TABLE III
Simulation Parameters for Experimental Study

Parameter	*Value*
Simulation area	602 m by 622 m
Number of vehicles	50, 100, 200
Transmission range	120 m
Initial placement of vehicles	SUMO traces
Number of malicious vehicles	5, 10, 15, 20, 25, 30, 35, 40
Vehicle travel speed	5 m/s, 10 m/s, 20 m/s
Simulation duration	900 s

We use NS-2 as the network simulator [31], and the simulation parameters are listed in Table III.

To evaluate the performance of the proposed AIT system, we use the following three metrics, namely, precision (P), recall (R), and communication overhead (CO). Precision and recall are widely used performance metrics to evaluate the accuracy in machine learning and classification [32]. More specifically, in this research, precision and recall are defined as follows:

$$P = \frac{\text{Number of truly malicious vehicles detected}}{\text{Total number of malicious vehicles reported}} \tag{6}$$

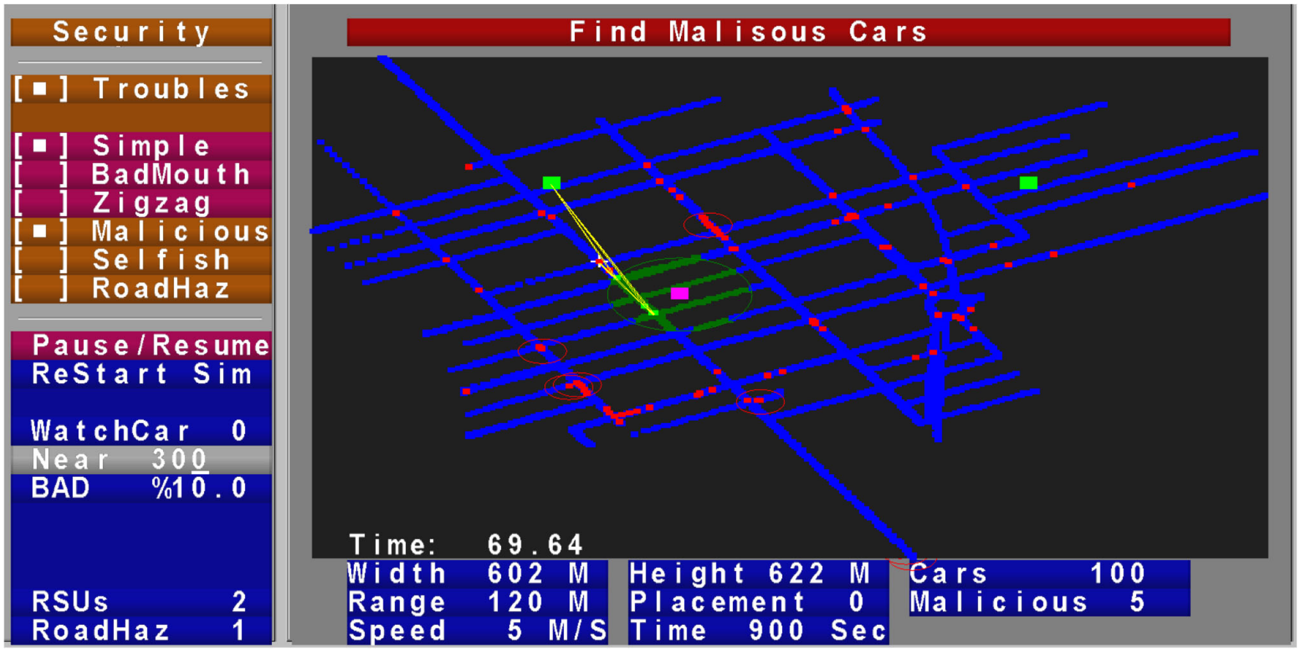
$$R = \frac{\text{Number of truly malicious vehicles detected}}{\text{Total number of truly malicious vehicles}}. \tag{7}$$

An example of network simulation that we built is shown in Fig. 7. We have compared the performance of the proposed AIT system with the following two baseline approaches: 1) the well-known weighted voting approach, which was widely adopted in various existing trust management schemes for wireless networks [15], [33], [34] and 2) the ART scheme [7].

B. Experimental Results

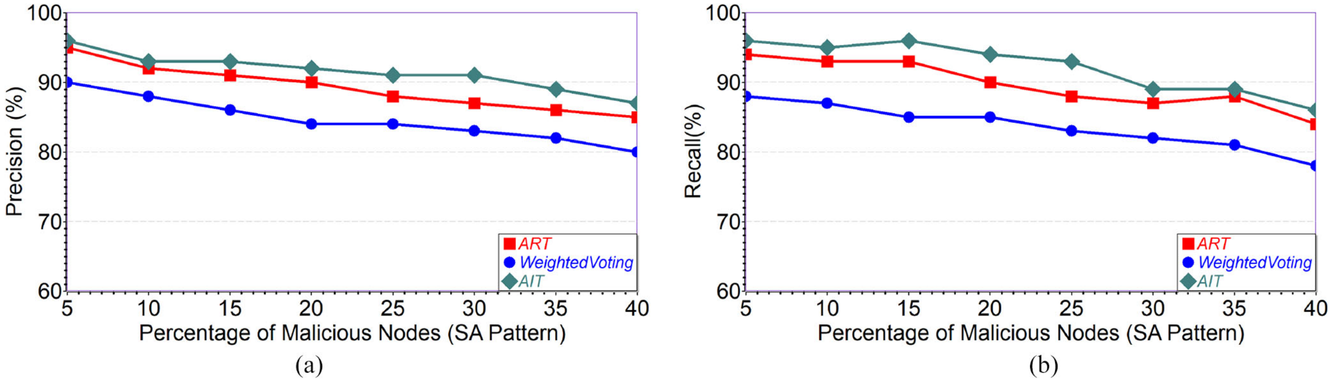
There are three series of experiments, and each of them aims at evaluating the proposed AIT system from different perspectives. The first series of experiments aims at evaluating the performance of the AIT system under different adversary models. As discussed in Section III-B, we have considered the following three types of malicious attacks in this work, namely, SA, BMA, and ZA. The experimental results for the first series of experiments are shown in Figs. 8–10.

As we can find from Figs. 8–10, the proposed AIT generally outperforms both two baseline approaches. The high precision and recall values for the AIT system are achieved owing to the application of the deep learning algorithm and



WatchCar 0
Near 300
BAD %10.0
RSUs 2
RoadHaz 1
Time: 69.64
Width 602 M
Range 120 M
Speed 5 M/S
Height 622 M
Placement 0
Time 900 Sec
Cars 100
Malicious 5

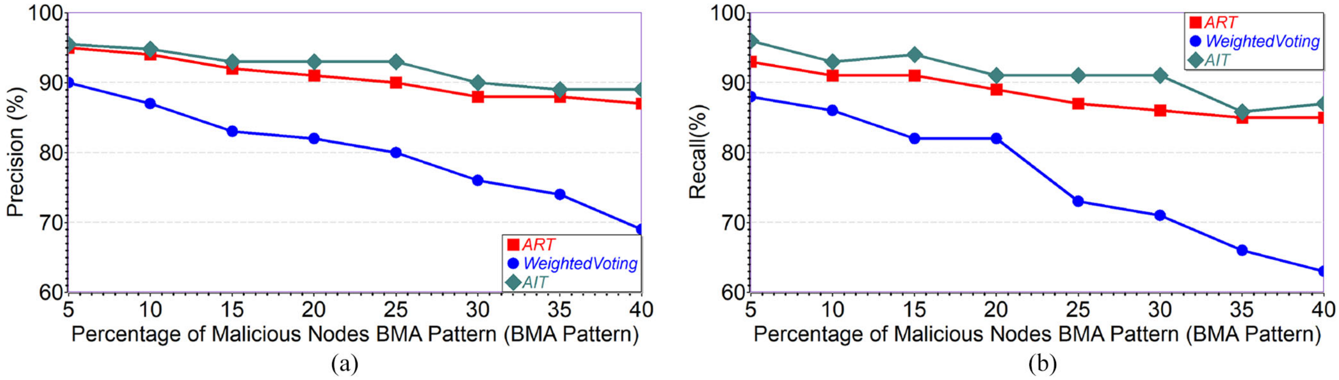
Fig. 7. Example of the network simulation for our experimental study.



(a)

(b)

Fig. 8. Performance comparison of AIT versus baseline approaches with the SA pattern. (a) Precision. (b) Recall.



(a)

(b)

Fig. 9. Performance comparison of AIT versus baseline approaches with the BMA pattern. (a) Precision. (b) Recall.

the blockchain technology, because the deep learning algorithm works well especially when there is a large volume of data, which is the case in the vehicular networks. In addition, the application of the blockchain technology will ensure the validity and authenticity of both vehicles and messages, thus resulting in a more accurate trust evaluation and detection of malicious vehicles.

Moreover, as clearly shown in Fig. 8, we find that the gap between AIT and the baseline approaches is not that significant when the adversary is adopting the SA pattern, which indicates

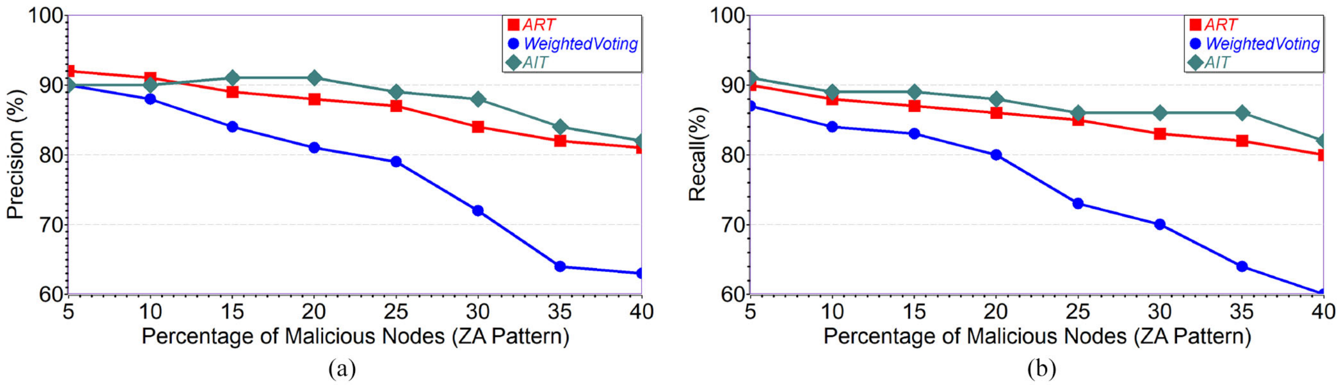


Fig. 10. Performance comparison of AIT versus baseline approaches with the ZA pattern. (a) Precision. (b) Recall.

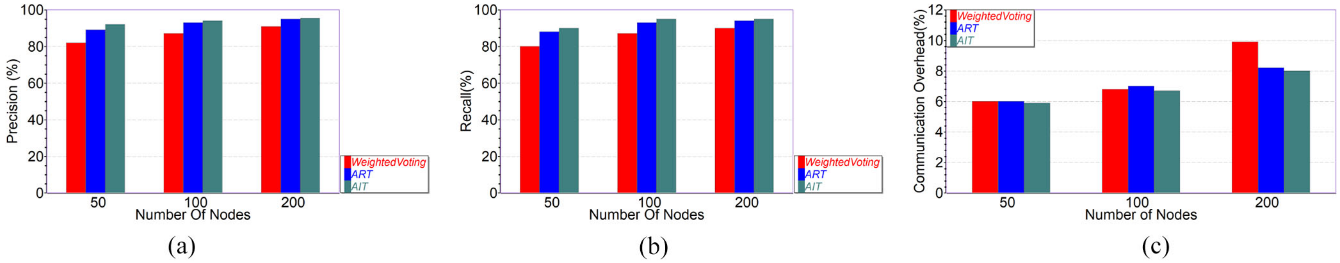


Fig. 11. Performance comparison of AIT versus baseline approaches with different numbers of vehicles. (a) Precision. (b) Recall. (c) CO.

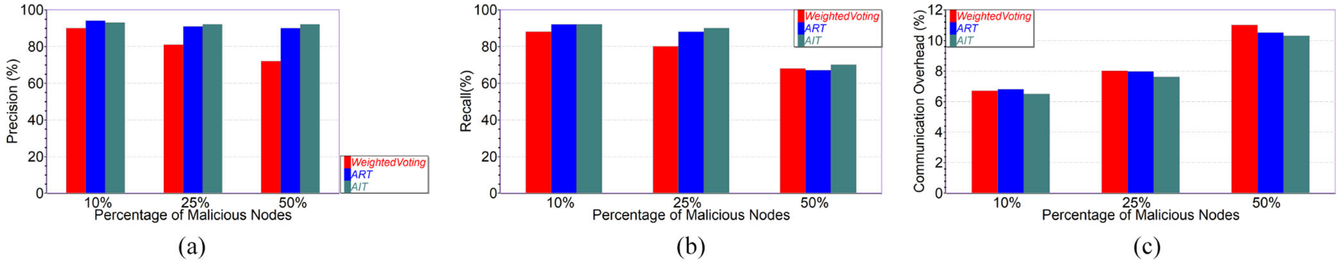


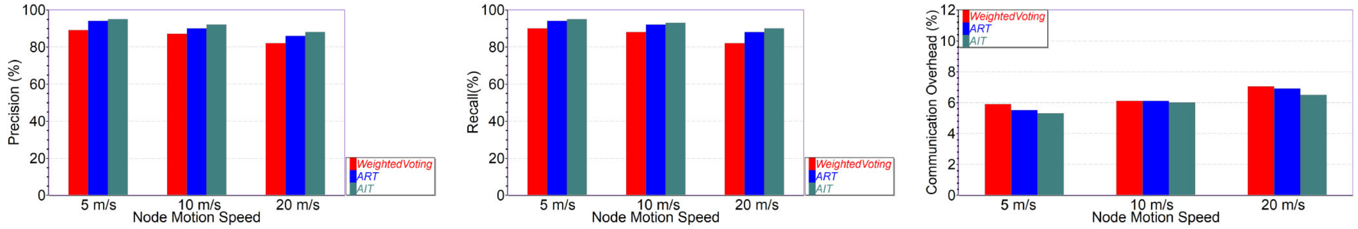
Fig. 12. Performance comparison of AIT versus baseline approaches with different numbers of malicious vehicles. (a) Precision. (b) Recall. (c) CO.

that the SA, as its name suggests, may not be very difficult to be detected. In contrast, the difference between AIT and the two baseline approaches is more significant with the BMA and ZA patterns, which is demonstrated in Figs. 9 and 10. This is the case because AIT benefits from the application of both deep learning and blockchain, which makes it more resistant to these two types of more sophisticated attacks.

The second series of experiments aim at evaluating the performance of the proposed AIT system in different experimental settings, such as the different numbers of nodes, different numbers of malicious nodes, and different traveling speeds. The experimental results for the second series of experiments are depicted in Figs. 11–13.

From Fig. 11(a), we can find that the AIT system generally works better when compared with the weighted voting method and ART scheme in terms of precision. As for the recall, Fig. 11(b) shows that AIT always outperforms both weighted voting and the ART scheme when the number of nodes differs. Moreover, the precision and recall are both higher when there are more nodes in the network. This is the case because it is more likely to receive true traffic updates from other vehicles when there are more benign vehicles in the network. Finally, as depicted in Fig. 11(c), the CO brought by AIT is slight lower than both weighted voting and ART, and even for 200 node case, the CO for AIT is under 8%, which suggests that the proposed AIT system does not generate too much additional network traffic.

Fig. 12 shows the effect of malicious nodes on AIT and the two baseline approaches. As Fig. 12(a) illustrates, the precision for AIT is generally better compared to those for the two baseline approaches. When there are 10% of malicious nodes in the vehicular network, the ART scheme works slightly better than AIT, but the difference is minimal. Fig. 12(b) shows that the recall for AIT is always higher than the two baseline approaches. Based on the comparison in terms of precision and recall, we also notice a performance drop when the percentage of malicious nodes increases. This is rational because with a higher percentage of malicious nodes in the networks, it is generally more difficult to receive the true traffic message from a trustworthy neighbor, which makes it harder to accurately evaluate the trust of vehicles and also successfully identify all the malicious vehicles.



(a) (b) (c)

Fig. 13. Performance comparison of AIT versus baseline approaches with different traveling speeds of vehicles. (a) Precision. (b) Recall. (c) CO.

TABLE IV
Performance Comparison Among FF, RNN, and CNN

Model	*Time to train (second)*	*Accuracy*
FF	39	96.7%
RNN	207	95.2 %
CNN	61	96.5%

Fig. 13 demonstrates the performance of AIT and the two baseline approaches with different node traveling speeds. As the map generated by SUMO represents part of the road networks in New York city, we are fully aware that in reality vehicles may not travel as fast as 20 m/s (which is equivalent to 72 km/h or approximately 45 m/h), due to the traffic condition as well as the designated city speed limit. However, we would still like to observe the performance of the three approaches when vehicles are traveling at that speed, so that we could get a more holistic view of how well they perform, especially when the traveling speed is relatively high. As shown in Fig. 13(a), AIT always outperforms both two baseline approaches in terms of precision, which clearly indicates that it works well when vehicles are traveling at different speeds. Fig. 13(b) shows that in terms of recall, AIT works better when vehicles are traveling at a lower speed, and ART may work slightly better in terms of the recall when the traveling speed is higher. But it is worth noting that the recall value for AIT and ART is similar when the traveling speed is 10 and 20 m/s. As shown in Fig. 13(c), the baseline, ART, and ATI systems all use communications bandwidth in order to evaluate trust on the network. The communication charts all show the percentage of total communication bandwidth that is used by each system. If malicious cars are found quicker using local trust, the extra communications needed for future evaluations is eliminated.

To summarize, we can see from Figs. 11–13 that the proposed AIT system generally outperforms the weighted voting and ART schemes under different circumstances, which also indicates that the AIT system can accurately evaluate the trust of vehicles and detect malicious ones with small overhead.

In the third series of experiments, our goal is to study the effect of different deep learning models and check to see which of them may be most suitable for the AIT system.

It is well understood that nodes in vehicular networks are constantly moving, and the network topology is rapidly changing due to the node mobility. Consequently, it is critical for the deep learning model to achieve high accuracy while taking less time to train the model.

We compare the performance of feedforward neural network (FF) with two other well-known deep learning models, namely, recurrent neural network (RNN) and convolutional neural network (CNN). The result is shown in Table IV.

From Table IV, we can find that all the three deep learning models perform similarly in terms of the accuracy, and the difference is minimal. In contrast, when it comes to the training time cost, FF clearly outperforms the other two deep learning models. This is true because FF has the simplest connection structure among nodes when compared to RNN: connections between the nodes in FF do not form a cycle, whereas connections between nodes form a directed graph in RNN. As for CNN, the use of convolutional function generally incurs additional time overhead when compared to the threshold function that FF generally uses for the hidden layers. Therefore, we conclude that FF is a suitable deep learning model for the AIT system because of its high accuracy and low training time cost.

VI. Discussion

A. Privacy Concern in Vehicular Networks

One of the assumptions that we have made in this research is that each vehicle's identity is unique and it could be used to distinguish one vehicle from others, which could possibly cause privacy leakage. In general, privacy has been viewed as one of the major concerns when securing vehicular networks [35].

To address the privacy concern in vehicular networks, one possible solution is to simply use the public key of each vehicle as its identifier [8]. Alternatively, a more sophisticated solution could be obtaining a pseudonym from the well-designed pseudonym scheme and using it to represent the vehicle [36]. However, it is worth noting that both solutions will incur a substantial amount of computational and CO, which may also be a valid concern for vehicular networks due to the resource constraint nature. As a result, an open research problem that should be further explored is how to effectively evaluate trust in vehicular networks while maintaining the privacy of vehicles.

VII. Conclusion

In this article, we proposed AIT, an AIT for vehicular networks using block technology. In the proposed AIT system, both vehicles and RSUs will participate in the trust management process, and both local trust evaluation and global trust evaluation help determine the trust of vehicles in a more accurate fashion. Moreover, the application of blockchain technology can ensure the validity and authenticity of information, such as the identity of nodes and the calculated trust values. To validate and evaluate the performance of the AIT system, extensive experiments have been conducted through network simulation. The experimental results clearly show that the AIT system can evaluate the trust of nodes and detect malicious ones in an accurate and efficient manner.

References

[1] H. Clancy. (Oct. 2015). *GM Already Has 1 Million Connected Cars on the Road*. [Online]. Available: https://fortune.com/2015/10/02/gm-1-million-connected-cars/

[2] F. Lambert. (Apr. 2019). *Tesla Unveils Robotaxi Plan for Self-Driving Ride-Sharing Network Next Year*. [Online]. Available: https://electrek.co/2019/04/22/tesla-robotaxi-network-self-driving-fleet-ride-sharing-cars/

[3] W. Li, A. Joshi, and T. Finin, "Coping with node misbehaviors in ad hoc networks: A multi-dimensional trust management approach," in *Proc. 11th IEEE Int. Conf. Mobile Data Manag. (IEEE MDM)*, 2010, pp. 85–94.

[4] W. Li, J. Parker, and A. Joshi, "Security through collaboration and trust in manets," *Mobile Netw. Appl.*, vol. 17, no. 3, pp. 342–352, 2012.

[5] Z. Wei, H. Tang, F. R. Yu, M. Wang, and P. Mason, "Security enhancements for mobile ad hoc networks with trust management using uncertain reasoning," *IEEE Trans. Veh. Technol.*, vol. 63, no. 9, pp. 4647–4658, Nov. 2014.

[6] K. Rostamzadeh, H. Nicanfar, N. Torabi, S. Gopalakrishnan, and V. C. Leung, "A context-aware trust-based information dissemination framework for vehicular networks," *IEEE Internet Things J.*, vol. 2, no. 2, pp. 121–132, Apr. 2015.

[7] W. Li and H. Song, "ART: An attack-resistant trust management scheme for securing vehicular ad hoc networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 4, pp. 960–969, Apr. 2016.

[8] Z. Yang, K. Yang, L. Lei, K. Zheng, and V. C. M. Leung, "Blockchain-based decentralized trust management in vehicular networks," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1495–1505, Apr. 2019.

[9] D. He, C. Chen, S. Chan, J. Bu, and A. V. Vasilakos, "Retrust: Attack-resistant and lightweight trust management for medical sensor networks," *IEEE Trans. Inf. Technol. Biomed.*, vol. 16, no. 4, pp. 623–632, Jul. 2012.

[10] F. Bao, R. Chen, M. Chang, and J.-H. Cho, "Hierarchical trust management for wireless sensor networks and its applications to trust-based routing and intrusion detection," *IEEE Trans. Netw. Service Manag.*, vol. 9, no. 2, pp. 169–183, Jun. 2012.

[11] Y. Ren, V. I. Zadorozhny, V. A. Oleshchuk, and F. Y. Li, "A novel approach to trust management in unattended wireless sensor networks," *IEEE Trans. Mobile Comput.*, vol. 13, no. 7, pp. 1409–1423, Jul. 2014.

[12] I.-R. Chen, F. Bao, and J. Guo, "Trust-based service management for social Internet of Things systems," *IEEE Trans. Depend. Secure Comput.*, vol. 13, no. 6, pp. 684–696, Nov./Dec. 2016.

[13] W. Li, H. Song, and F. Zeng, "Policy-based secure and trustworthy sensing for Internet of Things in smart cities," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 716–723, Apr. 2018.

[14] M. A. Azad, S. Bag, F. Hao, and A. Shalaginov, "Decentralized self-enforcing trust management system for social Internet of Things," *IEEE Internet Things J.*, vol. 7, no. 4, pp. 2690–2703, Apr. 2020.

[15] M. Raya, P. Papadimitratos, V. D. Gligor, and J.-P. Hubaux, "On data-centric trust establishment in ephemeral ad hoc networks," in *Proc. IEEE INFOCOM 27th Conf. Comput. Commun.*, 2008, pp. 1238–1246.

[16] X. Yao, X. Zhang, H. Ning, and P. Li, "Using trust model to ensure reliable data acquisition in VANETs," *Ad Hoc Netw.*, vol. 55, pp. 107–118, Feb. 2017.

[17] J. Guo *et al.*, "TROVE: A context awareness trust model for VANETs using reinforcement learning," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 6647–6662, Jul. 2020.

[18] H. El Sayed, S. Zeadally, and D. Puthal, "Design and evaluation of a novel hierarchical trust assessment approach for vehicular networks," *Veh. Commun.*, vol. 24, Aug. 2020, Art. no. 100227.

[19] J. Zhang, K. Zheng, D. Zhang, and B. Yan, "AATMS: An anti-attack trust management scheme in vanet," *IEEE Access*, vol. 8, pp. 21077–21090, 2020.

[20] F. Ahmad, F. Kurugollu, A. Adnane, R. Hussain, and F. Hussain, "MARINE: Man-in-the-middle attack resistant trust model in connected vehicles," *IEEE Internet Things J.*, vol. 7, no. 4, pp. 3310–3322, Apr. 2020.

[21] W. Dong, Y. Li, R. Hou, X. Lv, H. Li, and B. Sun, "A blockchain-based hierarchical reputation management scheme in vehicular network," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, 2019, pp. 1–6.

[22] X. Liu, H. Huang, F. Xiao, and Z. Ma, "A blockchain-based trust management with conditional privacy-preserving announcement scheme for VANETs," *IEEE Internet Things J.*, vol. 7, no. 5, pp. 4101–4112, May 2020.

[23] H. Xia, S.-S. Zhang, Y. Li, Z.-K. Pan, X. Peng, and X.-Z. Cheng, "An attack-resistant trust inference model for securing routing in vehicular ad hoc networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 7, pp. 7108–7120, Jul. 2019.

[24] K. Gu, X. Dong, and W. Jia, "Malicious node detection scheme based on correlation of data and network topology in fog computing-based vanets," *IEEE Trans. Cloud Comput.*, early access, Apr. 3, 2019, doi: 10.1109/TCC.2020.2985050.

[25] S. Gyawali and Y. Qian, "Misbehavior detection using machine learning in vehicular communication networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2019, pp. 1–6.

[26] J. Chen, G. Mao, C. Li, and D. Zhang, "A topological approach to secure message dissemination in vehicular networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 1, pp. 135–148, Jan. 2020.

[27] W. Li, A. Joshi, and T. Finin, "SVM-case: An SVM-based context aware security framework for vehicular ad-hoc networks," in *Proc. IEEE 82nd Veh. Technol. Conf. (VTC-Fall)*, 2015, pp. 1–5.

[28] H. Sedjelmaci, S. M. Senouci, and T. Bouali, "Predict and prevent from misbehaving intruders in heterogeneous vehicular networks," *Veh. Commun.*, vol. 10, pp. 74–83, Oct. 2017.

[29] D. J. Montana and L. Davis, "Training feedforward neural networks using genetic algorithms," in *Proc. IJCAI*, vol. 89, 1989, pp. 762–767.

[30] P. A. Lopez *et al.*, "Microscopic traffic simulation using SUMO," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, 2018, pp. 2575–2582.

[31] *NS-2 Network Simulator*. Accessed: Dec. 27, 2020. [Online]. Available: http://nsnam.sourceforge.net/wiki/index.php/Main_Page

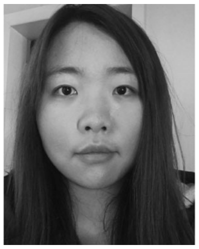
[32] J. Davis and M. Goadrich, "The relationship between precision-recall and roc curves," in *Proc. 23rd Int. Conf. Mach. Learn.*, 2006, pp. 233–240.

[33] S. Buchegger and J.-Y. Le Boudec, "Performance analysis of the confidant protocol," in *Proc. 3rd ACM Int. Symp. Mobile Ad Hoc Netw. Comput.*, 2002, pp. 226–236.

[34] I.-R. Chen, F. Bao, M. Chang, and J.-H. Cho, "Dynamic trust management for delay tolerant networks and its application to secure routing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 5, pp. 1200–1210, May 2014.

[35] B. Parno and A. Perrig, "Challenges in securing vehicular networks," in *Proc. Workshop Hot Topics Netw. (HotNets-IV)*, 2005, pp. 1–6.

[36] J. Petit, F. Schaub, M. Feiri, and F. Kargl, "Pseudonym schemes in vehicular networks: A survey," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 1, pp. 228–255, 1st Quart., 2015.



Chenyue Zhang received the B.S. degree in mechanical engineering from Case Western Reserve University, Cleveland, OH, USA, in May 2016, and the master's degree in computer science from New York Institute of Technology, New York, NY, USA, in December 2020.

She worked on developing prototypes for industrial machines and manufacturing robotics, which were built and developed based upon biological features and principles. Her research interests are in cyber security, blockchain, vehicular networks, machine vision, and machine learning.

Wenjia Li (Senior Member, IEEE) received the Ph.D. degree in computer science from the University of Maryland Baltimore County, Baltimore, MD, USA, in 2011.

He was a tenure-track Assistant Professor of Computer Science with Georgia Southern University, Statesboro, GA, USA, from 2011 to 2014. In 2014, he joined the Department of Computer Science, New York Institute of Technology, New York, NY, USA, as a tenure-track Assistant Professor, where he has been a Tenured Associate Professor since September 2020. His research has been supported by the National Institute of Health and the U.S. Department of Transportation Region two University Transportation Research Center. He has authored or coauthored over 80 peer-reviewed publications in various journals and conference proceedings. His current research interests include cyber security, mobile computing, and wireless networking, particularly security, trust, and policy issues for wireless networks, cyber–physical systems, Internet of Things, and intelligent transportation systems.

Dr. Li was a recipient of the 2019 IEEE Region 1 Technological Innovation (Academic) Award. He has served as the Organizing Committee Member for many international conferences, such as ACM WiSec, IEEE DySPAN, IEEE MDM, IEEE IPCCC, and IEEE Sarnoff, and he also served as a Reviewer for many prestigious journals, such as the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, the IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, the IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, the IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, and the IEEE INTERNET OF THINGS JOURNAL.

Yuansheng Luo received the B.S. and M.S. degrees from Hunan University, Changsha, China, in 2002 and 2005, respectively, and the Ph.D. degree from Xi’an Jiaotong University, Xi’an, China, in 2010.

He is currently a Lecturer with the School of Computer and Communication Engineering, Changsha University of Science and Technology, Changsha. His current interests include fog/edge computing, service computing, and wireless networks.

Yupeng Hu (Senior Member, IEEE) received the M.S. and Ph.D. degrees in computer science from Hunan University, Changsha, China, in 2005, and 2008, respectively.

He is currently a Professor with the College of Computer Science and Electronic Engineering, Hunan University, where he is the Dean of the Department of Cyberspace Security. He was with the Department of Computer Science and Engineering, University of Texas at Arlington, Arlington, TX, USA, as a Visiting Scholar from 2015 to 2016. He was also with IBM China Development Laboratory, Shanghai, China, as an Academic Visitor in 2012. He has published more than 60 journal articles, book chapters, and refereed conference papers. His research interests include big data and storage systems security, erasure coding, AI security, and network and system security.

Prof. Hu is a Senior Member of ACM.