Malicious Node Detection Scheme Based on Correlation of Data and Network Topology in Fog Computing-based VANETs

Ke Gu, XinYing Dong, and WeiJia Jia

***Abstract*—In vehicle ad hoc networks (VANETs), if a legal vehicle node becomes malicious, then it is more likely to tamper with transferred data or provide false data easily. Because the malicious node is a valid internal user in VANETs, its behaviour is difficult to be detected only through some cryptographic methods. Then the behaviour may cause many serious traffic accidents. Based on the available (unencrypted) data only, how to detect out the internal malicious vehicle nodes by some lightweight methods needs to be researched in VANETs. Additionally, fog computing seamlessly integrates heterogeneous computing resources widely distributed in edge networks and then provides stronger computing services for users. Therefore, in this paper, we propose a malicious node detection scheme in fog computing-based VANETs, where the fog server uses the reputation calculation to score each suspicious node based on the correlation of acquired data and network topology. In our proposed scheme, we build a reputation mechanism to score each suspicious node according to the correlation between outlier detection of acquired data and influence of nodes. Based on our proposed experiments, our proposed scheme can efficiently and effectively detect out malicious vehicle nodes so that fog server can acquire more true data.**

***Index Terms*—Malicious node; VANETs; Fog computing; Data security; Reputation.**

I. Introduction

A. Background

VANET refers to an open mobile network consisting of vehicles and related infrastructure relying on wireless communication and sensor network. It can acquire real-time data (such as traffic smoothness), improve traffic safety environment and reduce and prevent traffic accidents through close cooperation of vehicles, roads and people. It can also provide real-time traffic information service for traffic participants and decision support for traffic managers. For example, VANETs can provide and improve safety services and control decisions for drivers through collaborative control technology between vehicles [1]. The Vehicle Networking Forum of ITU points out that the deployment and application of VANETs in the future can reduce 30-70% traffic accidents. Currently the related safe messages in vehicle networking mainly include vehicle collision warning messages, traffic accident messages, traffic congestion messages and road early-warning messages. Because the transmission of the messages is related to life safety, there are higher requirements for transmission delay and data security [2]. However, in the VANET environment, we have to face various complex traffic environments. How to detect malicious vehicle nodes and improve data transmission effectiveness is of the most challenging application in VANETs. Because of the high mobility and rapid topological change of vehicles, the link maintenance time between nodes is short so that the connectivity of VANET is reduced. Thus, lower connectivity is easily exploited by malicious vehicle nodes to launch denial of service (DOS) attacks [3]. This phenomenon of low connectivity is particularly common in highway of remote areas or expressway with low vehicle density. Even in the urban traffic environment with high vehicle density, the practical communication distance between vehicles is shorter than the theoretical communication range due to the existence of a large number of buildings, intensive wireless signals and strong interference. At the same time, due to the extensive deployment of traffic lights, vehicles on the road often appear in clusters, and vehicle nodes can not form a network with stable topology and complete connectivity. Then, it is difficult to guarantee the connectivity of VANET. Therefore these factors greatly increase the transmission delay, insecurity and instability of traffic messages, which may cause various attacks such as Sybil attack, wormhole attack, and purposeful attack. In the attacks, it is heavy harmful for drivers and vehicles that malicious vehicle nodes tamper with traffic messages. For example, when a traffic accident occurs, a related malicious vehicle node provides other drivers with the traffic message without such accident, thus it leads to that some vehicles on this road continue to drive to the place that the accident occurred. The behaviours of malicious nodes may cause more serious traffic accidents. Therefore, all these kinds of attacks in VANETs not only affect drivers, but also damage traffic safety. Further, if a legal vehicle node becomes malicious, then the internal node easily tampers with transferred data, and the cryptographic methods are difficult to detect it. Therefore, how to detect out the internal malicious vehicle nodes by some lightweight methods needs to be focused on.

Additionally, the Cisco company [4] proposed a new computing concept called as fog computing, which moves computing, storage and other functions of cloud computing from the center to the edge of network [5]. In fog computing, fog servers are deployed at the edge of network. Each fog server is a highly virtualized computing system, similar to a lightweight cloud server. Fog servers can provide users with

Ke Gu and XinYing Dong are with School of Computer & Communication Engineering, Changsha University of Science & Technology, Changsha 410114, China.

WeiJia Jia is with Department of Computer and Information Science, University of Macau, Taipa, Macau.

Manuscript received 2019.

data storage, computing and wireless (wired) communication. Also, fog servers may communicate directly with fog devices. The specific structure is shown in Figure 1. Therefore, fog computing may provide a choice for traffic message acquisition and detection of malicious nodes.

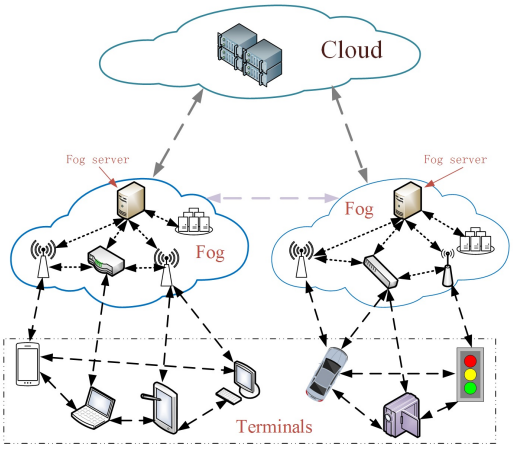


Figure 1 Hierarchical structure of fog computing

The architecture of fog computing may be introduced into VANETs, as shown in Figure 2. In the combined architecture, each vehicle is regarded as a mobile intelligent device (fog node) with multi-sensors[1], which has the computing and communication ability to acquire useful traffic information [6]. Then fog servers can be deployed at the edge of vehicle networking to acquire, process and store traffic data through base stations[2] in real time. According to the characteristic that fog services are closer to terminals, fog servers can be conducive to that vehicle nodes can form a network with stable topology and complete connectivity in VANETs. Therefore, we focus on how to detect out the internal malicious vehicle nodes based on the combined architecture.

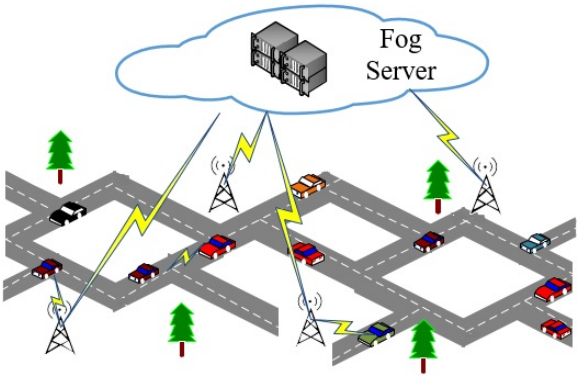


Figure 2 The architecture of fog computing-based VANETs

[1]The devices are seen as On Board Units (OBUs).
[2]The base stations are seen as Road Side Units (RSUs).

B. Our contributions

In this paper, we propose a malicious node detection scheme in fog computing-based VANETs, where the fog server uses the reputation calculation to score each suspicious node based on the correlation of acquired data and network topology. In our proposed scheme, we show how to calculate node influence in a network topology and propose an outlier detection approach to detect abnormal data. Further, based on the correlation between outlier detection of acquired data and influence of nodes, we propose a reputation mechanism to score each suspicious node. In this paper, our contributions are as follows:

(1) We analyze the security problems of acquiring unencrypted data in VANETs[3]. For these problems, we show a problem definition that how to detect malicious vehicle nodes based on the correlation of acquired data and network topology, where we use the reputation calculation to score each suspicious node.

(2) We propose a data acquisition framework for dynamic VANETs based on fog computing. In the proposed framework, each vehicle selects one of the nearby vehicles within its communication range as the next hop to transfer data until the data is transferred to the cluster head; finally the cluster head sends the data and the corresponding network topology to the fog server by the related based station.

(3) We show a malicious node detection scheme for VANETs. First, we show a method to calculate node influence for a dynamical network topology. Second, we show an outlier detection approach to detect abnormal data. Since logical data is discrete, we construct a binary system to transform logical data to continuous data. Thirdly, we build an approach to construct the correlation between outlier detection of acquired data and influence of nodes. Finally, we make reputation calculation to score each suspicious node for the detection of malicious vehicle nodes.

(4) We make some experiments to test and analyze the performance of our proposed scheme. We first test the running times of calculation of influences, detection of abnormal data, detection of suspicious nodes and detection of malicious nodes under different number of vehicle nodes. Second, for the effectiveness of our scheme, we test the correlation between node influence and abnormal data and the relation between node reputation and node influence. Further, we test the reputation change of the different types of vehicle nodes and the detection accuracy of malicious nodes (and suspicious nodes) under different proportion of dishonest nodes. Additionally, based on the detection rate of malicious nodes, we compare our scheme with other related works [7,8,25].

[3]In this paper, we mainly consider how to detect the internal malicious vehicle nodes that the legal vehicle nodes turned into.

C. Organization

The rest of this paper is organized as follows. In Section 2, we discuss the related works about malicious node detection in various networks. In Section 3, we show an attack model and a problem definition for malicious node detection. In Section 4, we propose a malicious node detection scheme in fog computing-based VANETs. In Section 5, we make some experiments to test the efficiency and effectiveness of our scheme. Finally, we draw our conclusions in Section 6.

II. Related work

Many researchers have focused on malicious node detection in various networks. Wang et al. [9] proposed a trust-based malicious node detection mechanism for mobile ad hoc networks (MANETs). They use the concepts of evidence chain and trust fluctuation to evaluate a node in the network. In their scheme, evidence chain is used to detect the malicious behavior of a node, trust fluctuation is used to reflect the high volatility of a node's trust value over a period of time. Ebinger et al. [10] proposed a cooperative intrusion detection method for MANETs based on trust evaluation and reputation exchange. In their scheme, credit information is first divided into trust and exchanged reputation, and then they are combined to be used for intrusion detection. In mobile wireless sensor networks (MWSNs), several trust management protocols for network security, data integrity and secure routing have been proposed [11,12,13]. However, the schemes are mainly based on reputation mechanism to manage related nodes through the structure of MWSN, without considering the mobility of mobile wireless sensors. For example, Shaikh et al. [13] proposed a grouping-based trust management scheme for MWSNs. In their scheme, they only consider directly observed-based QoS metrics (such as message passing rate in time window). Althunibat et al. [14] considered that dependent and independent malicious nodes have the same impact on target detection WSNs, and then proposed an effective algorithm that detects malicious nodes in the network regardless of their type and number. To explore reliable data fusion in MWSNs under Byzantine attacks, Abdelhakim et al. [15] proposed an effective malicious node detection scheme for adaptive data fusion under time-varying attacks, which is analyzed using the entropy-based trust model. Vempaty et al. [16] proposed two schemes for the mitigation of Byzantine attacks. The first scheme is based on a Byzantine identification method under the assumption of identical local quantizers. Further, the second scheme is proposed in conjunction with their proposed identification scheme where dynamic non-identical threshold quantizers are used at the sensors.

At present, many malicious node detection schemes have also been proposed in VANETs. Raya et al. [17] proposed an error detection system to exclude malicious vehicle nodes from the communication system of VANET. Their method is based on the deviation between behaviors of attacking nodes and normal behaviors. Leinmuller et al. [18,19] proposed a basic location verification method for evaluating vehicle geographic routing cooperation in VANETs. In their scheme, the consistency of location data is checked according to the changes of vehicle movement and vehicle density and map. Yan et al. [20,21] also proposed a location verification method. In their work, it is assumed that each vehicle node is equipped with GPS device. Then their method may check whether the GPS position claimed by the sender matches the estimated position. Xiao et al. [22] proposed a locally distributed scheme for detecting Sybil (ID spoofing) attack in VANETs. Their scheme makes statistical analysis of signal intensity distribution to detect the attack through roadside base stations. Lv et al. [23] proposed a cooperative RSS-based Sybil attack detection method, which is suitable for all static sensor networks with fixed transmission power. In their scheme, they assume that a vehicle node is either honest or malicious, and each node can eavesdrop on the data packets and calculate the distance to other nodes using the received signal strength. So, each node can create a group of adjacent nodes according to the similarity of RSS values, and broadcast the grouping result regularly. Some nodes with similar RSS values are grouped into suspicious group. Dhurandher et al. [24] proposed a vehicular security detection algorithm through reputation and plausibility checks to address the most important issue of security in VANETs. Their proposed algorithm can provide security against the attacks of event modification, false event generation, data aggregation and data dropping. Al Zamil et al. [25] proposed a prediction scheme, which is based on a binary classification using hidden Markov model (HMM). Their scheme can facilitate the precise prediction of false contents. Pereira et al. [26] showed a generic architecture for the deployment of fog computing applications and services in a VANET environment, and proposed a proof-of-concept system to perform data analytics in a hybrid VANET/Fog environment.

Additionally, many scholars focus on fog computing and its applications, where it is widely researched that how fog computing is used to VANETs. However, the combination of VANET and fog computing is facing many security challenges [27-31], such as data protection, detection of malicious nodes and detection of various attacks. Soleymani et al. [27] proposed a fuzzy trust model based on experience and credibility to ensure the security of vehicle networking. Their scheme uses fog nodes as a tool to evaluate the accuracy of traffic event location. Ni et al. [28] studied the security, privacy and fairness requirements of VANETs based on fog computing, and proposed some possible solutions to achieve security, privacy protection and incentive fairness. Yi et al. [29] considered that fog computing naturally inherits some security and privacy issues from cloud computing, and it faces new security and privacy challenges. Lee et al. [30] proposed several security threats to Internet of Things (IoT) when fog computing is introduced into IoT. They also proposed some potential security technologies against the threats. Stojmenovic et al. [31] proposed many applications of fog computing, such as smart grid, intelligent traffic lights in vehicle networking, software defined network and so on. In recent years, fog computing is also used to many other applications [32-37]. Hu et al. [32,33] proposed a framework of face recognition based on fog computing, and summarized its security and privacy problems. In [32], they suggested that some approaches, such

TABLE I
THE COMPARISON OF RELATED WORKS

Related works	High detection rate (≥90%)	Low detection cost	Detection of malicious nodes evolved from honest nodes	Detection of false injection data	For dynamic networks	For different types of malicious nodes and different types of data
Reference[14]	×	√	√	√	×	√
Reference[15]	√	√	×	√	√	×
Reference[16]	√	√	×	√	×	×
Reference[24]	×	√	×	√	√	×
Reference[25]	×	√	×	√	√	×
Reference[26]	×	√	×	×	√	×
Reference[27]	×	√	×	×	√	×
Reference[38]	×	×	√	×	√	×
Reference[39]	×	√	×	×	√	×

as authentication, session key protocol, data encryption and data integrity checking, can be used to solve the problems of confidentiality, integrity and availability. In [33], they implemented a prototype system based on local binary pattern identifier, where some computing overhead is transferred from cloud devices to fog devices.

Therefore, in vehicle ad hoc networks and mobile wireless sensor networks, many schemes were proposed to detect the data transmitted by the intermediate nodes and discover the behavior of data falsification. However, for the detection of malicious internal nodes, when the nodes can tamper with data or provide false data, many proposed schemes (such as encryption and decryption, identity authentication, etc.) are difficult to detect out specific nodes, especially for dynamic real-time network. In VANETs, when a legal node becomes malicious, it can obtain available data in the transmission process because the whole collection process needs aggregation calculation. Then there is a chance to tamper with data or inject false data by the node. For this type of attacks, although many related works [14-16,24-27,38,39] have studied the behavior of malicious nodes, these proposed schemes have some limitations to detect out specific malicious internal nodes. The advantages and disadvantages of these schemes are shown in Table 1. Further, some schemes mainly use the difference of network flow to detect malicious (abnormal) nodes, thus the schemes do not consider the correctness of data. In this paper, considering the correctness of transferred data, how to detect out malicious vehicle nodes by some lightweight methods is focused on.

III. ATTACK MODEL AND PROBLEM DEFINITION

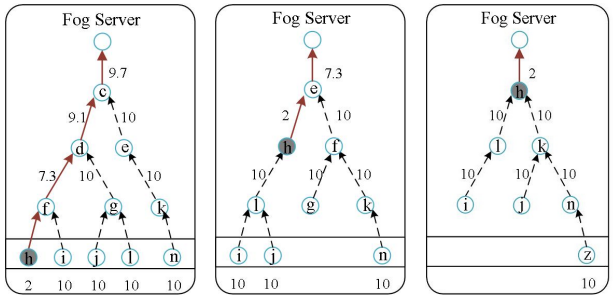
A. Attack model

In the section, we discuss the attack threats for transferred data in VANETs when the malicious vehicle nodes are some internal nodes. In our attack threat model, the malicious internal nodes mainly tamper with transferred data. We analyze three kinds of attack threat for transferred data in VANETs as follows:

1) Malicious vehicle nodes do not provide any data collection services in a period of time. These nodes do not send the required data to the fog server regularly, or when the fog server sends the data collection requests to these nodes, they do not respond. In this case, the fact that the fog server can not acquire any data from some malicious nodes should have a limited impact on data collection. Because VANET is a mobile network, the fog server can easily get the required data from other honest nodes.
2) Fog servers acquire continuous data from vehicle nodes, where some malicious nodes tamper with continuous data. For example, each vehicle may evaluate and score current traffic smoothness by its own perceived speed data. As shown in Figure 3, the fog server acquires the score data from the vehicle nodes, where the vehicle node h is a malicious node and the continuous data is the score of traffic smoothness. The score varies from 0 to 10, the value of 0 indicates that the current traffic is very smooth and the value of 10 indicates that the current traffic jam is very serious. As VANET is a mobile network, Figure 3 shows a dynamic network topology over time, where we assume that the network topology is a tree structure. In Figure 3(a), at the time $t = 1$, the node h is a leaf node at the bottom of the tree, the score of h uploading to the next hop is 2, which means that the current traffic is smooth. The influence of h is the smallest for data aggregation in the tree. So, when the final score data is uploaded to the fog server through data aggregation, the final result is 9.7, which means that the current traffic jam is serious. In Figure 3(b), at the time $t = 2$, the node h is at the middle of the tree, the score of h uploading to the next hop is still 2. Because the influence of h is bigger than that of the previous time, the final aggregation result is 7.3, which means that the current traffic is still not smooth. In Figure 3(c), at the time $t = 3$, the node h is as a cluster head at the top of the tree, similarly the score of h uploading to the next hop is 2. Because the influence of h is the greatest, the final aggregation result is 2, which means that the current traffic is smooth. It can be seen that the greater the influence of a malicious node is, the greater the deviation of the final result from the true value is.
3) Fog servers acquire logical data[4] from vehicle nodes, where some malicious nodes tamper with logical data. For example, when a fog server sends a request to the related vehicle nodes to inquire whether a traffic

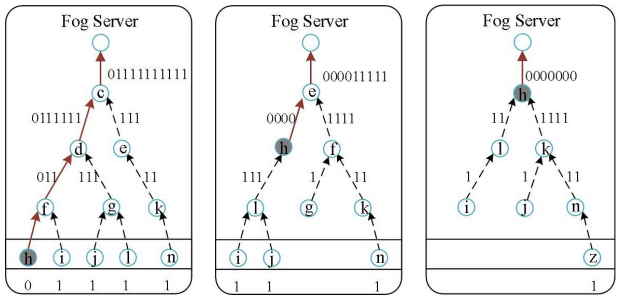
[4]This logical data is "true or false" or "yes or no".

accident has occurred, some malicious nodes may send the opposite responses to the fog server. As shown in Figure 4, we use a binary sequence to represent an inquired result, where "1" means "yes" and "0" means "no". In the binary sequence, each bit is associated with a vehicle node. Also, we assume that the network topology is also a dynamic tree structure over time, h is a malicious node and the fact is that a traffic accident has occurred. In Figure 4(a), at the time $t = 1$, the node h is a leaf node, the data of h uploading to the next hop is "0". That is to say, there is no accident. The influence of h is the smallest for data acquisition in the tree. So, when the final acquired data is uploaded to the fog server, the final result sequence is "011111111111". Similar to the description of Figure 3, at the time $t = 2$ in Figure 4(b), the node h is at the middle of the tree, the result sequence of h uploading to the next hop is "0000"[5]. So, the final result sequence is "000011111". At the time $t = 3$ in Figure 4(c), the node h is as a cluster head at the top of the tree, thus the final result sequence is "0000000". It can also be seen that the greater the influence of malicious nodes is, the greater the deviation of the final result from the true result is. At the time $t = 3$, the fog server knows from the acquired data that no accident occurred, contrary to the fact.



(a) h is at the bottom of the tree (b) h is at the middle of the tree (c) h is at the top of the tree

Figure 3 Tampering with continuous data



(a) h is at the bottom of the tree (b) h is at the middle of the tree (c) h is at the top of the tree

Figure 4 Tampering with logical data

[5]In fact, h may also discard its received result sequence "111" and only send "0" to the next node. However, for the effectiveness of the attack, the modified result sequence "0000" sent by h can maximally interfere the final result sequence.

In the attack model, we differentiate the processes of the continuous data and the logical data, mainly because of the different ways of data acquisition and data falsification made by the attackers. For example, in the continuous data, the data collectors (fog servers) use the way of aggregation calculation (such as calculating average value or taking maximum value) to acquire data. In this process, each intermediate node performs aggregation calculation on its received data and then sends the aggregated data to its upper layer's node; while some attackers (as the intermediate nodes) can modify the aggregated value (or provide false data value) and then send the value to its upper layer's node, resulting in the deviation of the final aggregated value. In the logical data, each user's answer is expressed as "1" or "0". Then, for the data collectors, the intermediate nodes cannot perform aggregation calculations. The intermediate nodes can only splice the values of 1 or 0 to form a string of binary and then send the data string to the data collectors. The data collectors may determine whether the final answer is "yes" or "no" according to the number of 1 or 0 in the acquired binary string. Therefore, the attackers (as the intermediate nodes) can only directly modify the binary answers (including their own answers and the received answers) to disturb the final submitted results in the logical data. Due to the different ways of data acquisition and data falsification, we need to investigate and analyze the impact of malicious nodes on the final results under the two types of data. Additionally, in the above three attack threats, there may be multiple malicious vehicle nodes to collusively tamper with inquired data. Their behaviours may reduce the credibility of honest vehicle nodes in a network topology and confuse the fog server, and the final result will be more deviated from the true value. Further, there are some nodes in the collusive malicious nodes, which can provide true response. These nodes early try to cover up the future malicious behaviours. Because these malicious nodes behave well in the early stage, it needs to take long time to detect out such malicious nodes.

B. Problem definition

In the section, we show a problem definition that how to detect malicious vehicle nodes based on the correlation of acquired data and network topology, where we use the reputation calculation to score each suspicious node. We set a network topology structure $G(V_m, E_m)$, where V_m is a set of nodes, E_m is a set of edges, $n_i \in V_m$ represents the i-th vehicle node, m represents the number of nodes and $i \leq m$. Also, we assume that G_{t_j} is a dynamic network topology related to the time t_j, where the time $t_j \in T = \{t_1, t_2, ...t_\pi\}$, $j \leq \pi$ and π represents the length of the time sequence T; and the acquired data based on the topology G_{t_j} is represented as $D(G_{t_j})$ at the time t_j. Then the sequence of the acquired data is $D\_set = \{D(G_{t_1}), D(G_{t_2})...D(G_{t_\pi})\}$ in T. Further, we assume that the abnormal data detection result for D_set is a sequence $\{U(G_{t_1}), U(G_{t_2})...U(G_{t_\pi})\}$, and the influence of each node n_i for D_set is represented as $I_{t_j}(n_i)$ at the time t_j. Additionally, we set that R is a correlation calculation function between abnormal data detection result and influence of nodes,

X is a set of suspicious nodes, and ϑ is a threshold value determined as a suspicious node. Therefore, the reputation function RF for each node n_i belonging to X is represented as

$$RF(n_i \in X \mid R(\{(U(G_{t_j}), I_{t_j}(n_i))|t_j \in T\}) > \vartheta),$$

where we need to obtain each $U(G_{t_j})$ and $I_{t_j}(n_i)$, and then build the reputation function RF. In the following sections, we will describe how to obtain $U(G_{t_j})$ and $I_{t_j}(n_i)$, construct the correlation function R and build the reputation calculation for each suspicious node.

IV. Malicious Node Detection Scheme Based on Fog Computing in VANETs

A. Proposed Framework

In the section, we propose a data acquisition framework for dynamic VANETs based on fog computing. In the proposed framework, each vehicle selects one of the nearby vehicles within its communication range as the next hop to transfer data until the data is transferred to the cluster head; finally the cluster head sends the data to the fog server by the related based station (such as RSU). Figure 5 shows a system framework for fog computing-based VANETs, where some fog servers are deployed close to VANETs. The fog servers are responsible for acquiring, storing and analyzing data from VANETs, where each fog server can communicate with multiple based stations. Since the managing range of fog servers is different, if a moving vehicle moves out of the managing range of a fog server, then the current fog server can send the related information of the vehicle to the next fog server. In Figure 5, the red vehicle is a malicious node, its position in the VANET is changing[6].

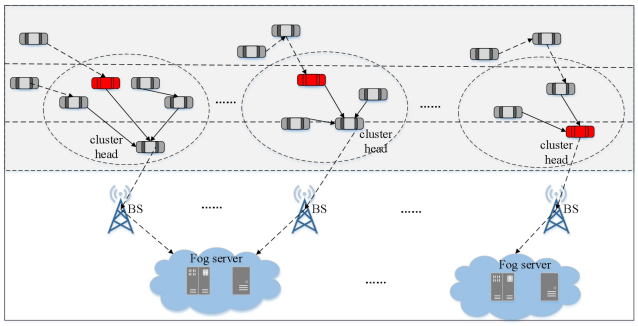


Figure 5 Proposed framework for dynamic fog computing-based VANETs

In our proposed data acquisition framework, the multi-hops communication mode is used to acquire data. It is beneficial to avoid the interaction bottleneck for the based stations and the fog servers in VANETs. For example, if the communication range between the vehicles and the based stations is not considered, and each vehicle node can interact directly with the fog server by the based station, then it may cause communication congestion or interaction difficulty between the vehicles and the fog servers. Further, if there is a malicious vehicle node to launch DOS attacks to the based stations (or the fog servers), then it will cause that other vehicle nodes cannot timely communicate with the based stations. Therefore, the multi-hops communication mode is adopted to our proposed data acquisition framework. Additionally, due to the different speeds of the vehicles in VANETs, the positions of the vehicles are changing. So, the different dynamic network topologies are formed, where the different vehicle nodes are randomly selected as the cluster heads.

[6]In this paper, we consider that each vehicle node used as a fog node is untrusted.

B. Malicious Node Detection Scheme

In the section, according to the defined problem in Section 3, we propose a malicious node detection scheme to detect malicious vehicle nodes based on the correlation of acquired data and network topology under our proposed framework. In the VANET, many different dynamic network topologies are formed, and there are the different nodes involved in different network topologies. Also, as the moving speed of each vehicle is different, the position of each vehicle node is different in the different network topologies. So, the influence of each vehicle node for data acquisition is also different in the different network topologies, as described in Section 3. The acquired data result is related to the position of vehicle nodes in a certain extent. So, if a malicious node is at a position of great influence, then it will lead to that the final acquired result can deviate from the true result. In our proposed scheme, we first construct an approach to build the correlation of acquired data and network topology. It is found that the bigger the influence of a vehicle node in a network topology is, the bigger the data result deviation related to the network topology is. Thus the vehicle node is considered as a suspicious node. Then a reputation mechanism is constructed and used to evaluate each suspicious node. Some important notations are described in Table 2.

TABLE II
Related Symbols

Related symbols	Description
Ad_*	Aggregated data
Rd_*	Network topology
$I_{t_j}(n_i)$	The influence of the suspicious node n_i at time t_j
$TRAN(r_{t_j})$	The transformed continuous data of logical data at time t_j
$D(G_{t_j})$	The acquired data point based on the network topology G_{t_j} at time t_j
$U(G_{t_j})$	The abnormal level value of continuous data based on the network topology G_{t_j} at time t_j
R	Correlation value
$S_j(f, n_i)$	The satisfaction degree of the current fog server to the j-th data report from the suspicious node n_i
RF_{n_i}	The reputation value of the suspicious node n_i

1) Data acquisition mode based on dynamic network topology:

In the section, we show a data acquisition mode based on dynamic network topology. In our data acquisition mode, fog servers can send data requests to vehicles within a certain communication range through base stations (or vehicles can actively and regularly send data to fog servers). Each vehicle

periodically selects one of the nearby vehicles within its communication range as the next hop to transfer data (including related path message) until the data is transferred to the cluster head. Finally the cluster head sends the required data and the corresponding network topology to the fog server by the related based station. So, for a period of time, the fog server can acquire the aggregated data transferred by some vehicles based on the different network topologies. The fog server can store these data results, the corresponding network topologies and the corresponding times[7].

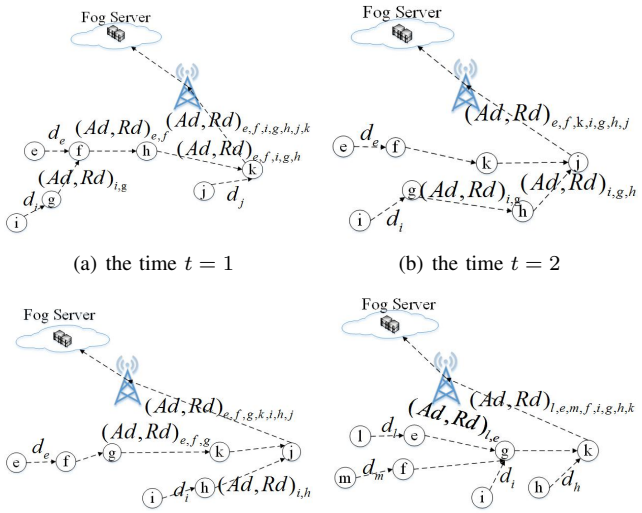
Figure 6 shows how the fog server acquires the data and the corresponding network topology through a dynamic network structure composed of multiple vehicle nodes. In Figure 6(a), d_e is the original data of the node e and transferred to the node f, $Ad_{e,f}$ is an aggregated data based on d_e and d_f (d_f is the original data of the node f)[8], $Rd_{e,f}$ records the related path through which the corresponding data is transferred. Then $Ad_{e,f}$ and $Rd_{e,f}$ are both sent to the node h. Finally, an aggregated data $Ad_{e,f,i,g,h,j,k}$ and a related complete path $Rd_{e,f,i,g,h,j,k}$ are formed and sent to the fog server by the based station, where $Rd_{e,f,i,g,h,j,k}$ records a network topology for this data acquisition. As can be seen from Figure 6, all the vehicle nodes move constantly, thus the positions of the vehicles are rapidly changing and some new vehicle nodes join (or some vehicle nodes leave). In Figure 6(d), the fog server acquires the data based on the network topology composed of 8 nodes. In our proposed data acquisition mode, the cluster head is randomly selected and the network topology for data acquisition is randomly formed even if the VANET topology is relatively stable in a period of time.

2) Calculation of node influence:

In the section, we show how to calculate node influence in a network topology. If a VANET topology is regarded as a dynamic directed graph structure, then the degree centrality method [40] is used to judge node influence related to time in a directed graph structure, by calculating the degree of nodes or the shortest path between nodes. As the complexity of the degree centrality method is low, this method is widely used in the structural analysis of social networks. According to the degree centrality method, node influence is mainly measured by the shortest path in a network topology. Since all data is transferred to cluster head node under our proposed data

[7]In our proposed data acquisition framework, the fog servers are responsible for acquiring, storing and analyzing data from VANETs, thus the fog servers are trusted in our framework. Although these acquired data results may bring some privacy issues (such as node trajectory), the function that provides inquired data to the fog servers is not mandatory for vehicle nodes. Of course, some privacy protection methods may protect the acquired data when the vehicle nodes submit these data to the fog servers. Because the target of our proposed scheme is to detect the malicious nodes, some data privacy protection methods may bring about the interference to discover the malicious nodes. Thus, we consider that the data privacy protection is contradictory to our proposed malicious node detection scheme. It is a challenge for us to provide the detection of malicious nodes and to provide the data privacy protection in a fog computing-based VANET. In the future work, we consider that it is worth researching that how to provide the detection of malicious nodes while providing the data privacy protection in VANETs.

[8]The aggregation procedure is similar to Figures 3 and 4.



(a) the time $t = 1$ (b) the time $t = 2$

(c) the time $t = 3$ (d) the time $t = 4$

Figure 6 Data acquisition based on a dynamic network topology

acquisition framework, we mainly focus on node influence for data acquisition between all nodes and cluster head node.

Definition 1. *Node influence for data acquisition: based on the degree centrality, the metrics that a node n_i at the time t_j can modify transferred data is defined as follows:*

$$I_{t_j}(n_i) = \frac{\sum_{v \in V, v \neq n_i, v \neq c} \frac{P_{v,c}(n_i)}{P_{v,c}}}{m \cdot (m-1)}, \tag{1}$$

where $P_{v,c}(n_i)$ represents the number of the shortest paths passing the node n_i between the node v and the cluster head node c, $P_{v,c}$ represents the number of all the shortest paths between the node v and the cluster head node c, m is the number of nodes in the network topology.

In the above Equation 1, the numerator $\sum_{v \in V, v \neq n_i, v \neq c} \frac{P_{v,c}(n_i)}{P_{v,c}}$ denotes the sum of all the ratios between $P_{v,c}(n_i)$ and $P_{v,c}$ for all node vs with $v \in V$; the denominator $m \cdot (m-1)$ represents the maximum number of edges when there are m nodes in a directed graph, where we further use the maximum value generated by the denominator as a measure to uniformly quantify the value generated by the numerator[9]. It can show the importance of the node n_i when all node vs send the messages to the cluster head node c. Namely this formula shows how many messages must pass through the node n_i when all node vs send the messages to c. Therefore, the node influence $I_{t_j}(n_i)$ reflects the importance of the position of the node n_i in a network topology. It is a measure of the impact of n_i on transferred messages between all other nodes and cluster head node at the time t_j. The greater the influence of the node n_i is, the greater the amount of messages passing through the node n_i is. Then the node n_i has the greater influence for data acquisition.

Lemma 1. *The value of $I_{t_j}(n_i)$ belongs to (0, 1), where close*

[9]For the dynamic network, the further quantifying of the numerator can generate a correlation with the size of network, so that the evaluation of node influence is more comprehensive and accurate.

to 1 denotes the node n_i has the biggest influence and close to 0 denotes the node n_i has the smallest influence.

Proof. Based on $I_{t_j}(n_i) = \frac{\sum_{v \in V, v \neq n_i, v \neq c} \frac{P_{v,c}(n_i)}{P_{v,c}}}{m \cdot (m-1)}$, we can know $P_{v,c}(n_i) \leq P_{v,c}$ and $m > 2$, thus $0 < I_{t_j}(n_i) < 1$. When $P_{v,c}(n_i) = P_{v,c}$, $I_{t_j}(n_i)$ has the biggest value. It denotes that all the shortest paths both pass the node n_i between the node v and the cluster head node c, thus n_i has the biggest influence. Other, when $P_{v,c}(n_i) \ll P_{v,c}$, $I_{t_j}(n_i)$ has the smallest value, which denotes that n_i has the smallest influence. □

We set that $G(V, E)$ is a network topology based on a VANET, V is a set of the nodes, E is a set of the edges, $|V|$ is the size of V and $I(n)$ is the influence of the node $n \in V$. Therefore, the influence $I(n)$ of the vehicle node n in a network topology for data acquisition is calculated as follows:

- Step 1: Compute the number $P_{v,c}$ of the shortest paths between the node v and the cluster head node c, and save $P_{v,c}$;
- Step 2: Compute the number $P_{v,c}(n)$ of the shortest paths passing the node n between the node v and the cluster head node c, and save $P_{v,c}$ where $v \neq c \neq n$;
- Step 3: Repeat Steps 1 and 2 until each $v \in V$ with $v \neq c$ is computed;
- Step 4: Compute the influence $I(n)$ of n according to Equation 1.

The degree centrality method for computing the influence of nodes in a network topology is described as Algorithm 1, where we set that c is a cluster head node, n is the assigned node passed in the related shortest paths.

Algorithm 1 Calculating Node Influence

Input: $G(V, E)$, c, n
Output: $I(n)$

Node-Influence($G(V, E)$, c, n)
Begin
  $Ratio\_Sum = 0$;
  while $v \in V \& v \neq c \& v \neq n$ **do**
    $P_{v,c}$ =**ShortestPathNum**(v, c);
    $P_{v,c}(n)$ =**ShortestPathNumByAssignedNode**(v, c, n);
    $Ratio\_Sum = Ratio\_Sum + \frac{P_{v,c}(n)}{P_{v,c}}$;
  End while
  $I(n) = \frac{Ratio\_Sum}{|V| \cdot (|V|-1)}$;
  return $I(n)$;
End

3) Abnormal data detection:

In the section, we show an outlier detection approach to detect abnormal data for continuous data in VANETs. As logical data is discrete, we must equivalently transform logical data to continuous data.

- Transforming of logical data
  We use a binary system to represent the logical data, where "1" means true and "0" means false. Then we use a binary sequence to represent an inquired result. We set r_{t_j} is a binary sequence composed of $b_1 b_2 ... b_m$ at the time t_j, where b_i is a bit denoting the answer of the vehicle node i with $i \leq m$, m is the number of nodes in a network topology. For example, the binary sequence r_{t_j}="111" means that the answers of three vehicle nodes are both true. So, in order to measure the deviation of a binary sequence, it needs to be transformed to continuous data by a normalized Euclidean distance. The transforming function is defined as follows:

$$TRAN(r_{t_j}) = \frac{\sqrt{\sum_{i=1}^{|r_{t_j}|} (b_i)^2}}{\sqrt{|r_{t_j}|}}, \quad (2)$$

  where $|r_{t_j}|$ is the length of the binary sequence, namely $|r_{t_j}| = m$. For example, a binary sequence r_{t_j}="1100", its transformed continuous data is $\frac{\sqrt{1^2+1^2+0^2+0^2}}{\sqrt{4}}$ =0.707.
- Outlier detection of continuous data
  For the processing of continuous data, if one or several continuous data points acquired by the fog server at different times in a period of time are very different from others, then these data points are considered as outliers. Namely they are suspected to be caused by some malicious behaviors. For outlier detection, our processing is based on the assumption that most of the data acquired by the fog server over a period of time are true[10]. In this paper, the outlier detection method [41] is based on density-based local outlier detection. The outlier detection is used to identify the possible outlier data points in the data acquired by the fog server. Because there may be relatively few vehicle nodes in a certain area managed by a fog server, local outlier detection is more suitable. The outlier detection method can be used in many security fields, such as network intrusion detection. We assume that D_set ={$D(G_{t_1})$,$D(G_{t_2})$...$D(G_{t_\pi})$} is a set of the acquired data points, where $D(G_{t_j})$ is the data acquired by the fog server based on the network topology G_{t_j} at the time $t_j \in T = \{t_1, t_2, ...t_\pi\}$, $j \leq \pi$ and π represents the length of the time sequence. Also, we set that the distance of $D(G_{t_j})$ to $D(G_{t_i})$ is defined as $Dist(D(G_{t_j}), D(G_{t_i})) = \sqrt{(D(G_{t_j}) - D(G_{t_i}))^2}$; the k-th distance of $D(G_{t_j})$ is represented as $Dist_k(D(G_{t_j}))$, and the k-th distance field of $D(G_{t_j})$ is represented as $Ner_k(D(G_{t_j}))$, which is a set of all neighbor data points within the k-th distance $Dist_k(D(G_{t_j}))$. The k-th reachable distance of the data point $D(G_{t_j})$ to the data point $D(G_{t_i})$ is defined as:

$$R\_Dist_k(D(G_{t_j}), D(G_{t_i})) = \textbf{MAX}\{Dist_k(D(G_{t_j})), Dist(D(G_{t_j}), D(G_{t_i}))\},$$

  where the k-th reachable distance $R\_Dist_k(D(G_{t_j}), D(G_{t_i}))$ is the maximal value between the distance of $D(G_{t_j})$ to the k-th farthest data point and the distance of $D(G_{t_j})$ to $D(G_{t_i})$. The local reachable density of the

[10]If there is an extreme situation that most of vehicle nodes are malicious in a certain period of time, then the outlier detection may be based on the data acquired in a longer period of time.

data point $D(G_{t_j})$ is expressed as:

$$Ld_k(D(G_{t_j})) = \frac{1}{\frac{\sum_{t_i \in Ner_k(D(G_{t_j}))} R\_Dist_k(D(G_{t_j}),D(G_{t_i}))}{|Ner_k(D(G_{t_j}))|}} \quad (3)$$

where $|Ner_k(D(G_{t_j}))|$ is the number of data points in the set $Ner_k(D(G_{t_j}))$. The above formula represents the reciprocal of the average reachable distance from all the data points in the set $Ner_k(D(G_{t_j}))$ to the data point $D(G_{t_j})$. The local reachable density represents the density of a data point. The higher the density $Ld_k(D(G_{t_j}))$ of a data point $D(G_{t_j})$ is, the more likely $D(G_{t_j})$ and its surrounding data points belong to the same cluster. On the contrary, the lower the density of a data point is, the more likely the data point is an outlier. Then the local outlier factor of the data point $D(G_{t_j})$ is defined as:

$$U(G_{t_j}) = \frac{\sum_{t_i \in Ner_k(D(G_{t_j}))} \frac{Ld_k(D(G_{t_i}))}{Ld_k(D(G_{t_j}))}}{|Ner_k(D(G_{t_j}))|} = \frac{\sum_{t_i \in Ner_k(D(G_{t_j}))} Ld_k(D(G_{t_i}))}{|Ner_k(D(G_{t_j}))| \cdot Ld_k(D(G_{t_j}))} \quad (4)$$

where $U(G_{t_j})$ represents the ratio average of the local reachable density of each data point belonging to $Ner_k(D(G_{t_j}))$ to the local reachable density of the point $D(G_{t_j})$. If the ratio average is close to 1, then it means that the local reachable density of $D(G_{t_j})$ is similar to those of the neighbor data point $D(G_{t_i})$s belonging to $Ner_k(D(G_{t_j}))$. Thus $D(G_{t_j})$ and its neighbor data points belonging to $Ner_k(D(G_{t_j}))$ may belong to the same cluster. If the ratio average is smaller than 1, then it means that the local reachable density of $D(G_{t_j})$ is higher than those of its neighbor data points, thus $D(G_{t_j})$ is a dense point; otherwise, $D(G_{t_j})$ is considered as an abnormal data point when $U(G_{t_j}) > 1$.

According to the above formulas, we may use the outlier detection method to detect abnormal data points from the acquired continuous data in VANETs. The process is described as follows:

- For a set of the acquired data points, D_set =$\{D(G_{t_1}),D(G_{t_2})...D(G_{t_\pi})\}$, computing the k-th distance $Dist_k(D(G_{t_j}))$ of each data point $D(G_{t_j})$, with $j \in \{1, 2...\pi\}$;
- For each data point $D(G_{t_j})$ with $j \in \{1, 2...\pi\}$, computing the k-th reachable distance $R\_Dist_k(D(G_{t_j}), D(G_{t_i}))$ of $D(G_{t_j})$ to other data point $D(G_{t_i})$s with $i \in \{1, 2...\pi\}$ and $i \neq j$, and getting its k-th distance field $Ner_k(D(G_{t_j}))$;
- For each data point $D(G_{t_j})$ with $j \in \{1, 2...\pi\}$, computing its local reachable density $Ld_k(D(G_{t_j}))$;
- For each data point $D(G_{t_j})$ with $j \in \{1, 2...\pi\}$, computing its local outlier factor $U(G_{t_j})$;
- Evaluating each data point $D(G_{t_j})$ with $j \in \{1, 2...\pi\}$ according to $U(G_{t_j})$: if $U(G_{t_j})$ is higher than 1, then $D(G_{t_j})$ is considered as an abnormal data point.

4) Reputation-based detection of malicious nodes:

In the section, we propose a reputation mechanism to score each suspicious node, based on the correlation of acquired data and network topology. First, we build an approach to construct the correlation between outlier detection $U(G_{t_j})$ of acquired data and influence $I_{t_j}(n_i)$ of node. Then, we make reputation calculation to score each suspicious node for the detection of malicious vehicle nodes. The specific steps are as follows:

- **Step1** The current fog server acquires required data and network topology information generated by different network topologies at different times. Then the fog server makes outlier detection of data acquired in a certain period of time, and records these detection results of abnormal data.
- **Step2** According to the detection results of abnormal data, the correlation detection is performed for each node in the corresponding network topology. We set that $\{I_{t_1}(n_i),I_{t_2}(n_i)...I_{t_\pi}(n_i)\}$ is a set of influences of the node n_i related to the different time t_j in a period of time T, and $\{U(G_{t_1}),U(G_{t_2})...U(G_{t_\pi})\}$ is a set of outlier detection values of abnormal data related to the node n_i, with $t_j \in T = \{t_1, t_2, ...t_\pi\}$ and $j \leq \pi$. If it is found that a node in different network topologies has the following correlation: the detection value of abnormal data increases or decreases with the influence of the node when the node appears in the different network topologies, then the node can be considered as a suspicious node.

  Definition 2. *Suspicious nodes in VANETs: according to outlier detection $U(G_{t_j})$ and node influence $I_{t_j}(n_i)$ where the node n_i is related to the network topology G_{t_j}, the node n_i is detected whether it is suspicious according to the following formula [42]:*

$$R = 1 - 6 \cdot \sum_{j=1}^{\pi} \frac{(U(G_{t_j}) - I_{t_j}(n_i))^2}{\pi \cdot (\pi - 1)}. \quad (5)$$

  When the correlation value R of the node n_i is more than a preset value ϑ, we consider that the node n_i is a suspicious node.

  The correlation detection between node influences and detection values of data outliers is described as Algorithm 2. We set that $VNode[]$ is an array of nodes where each node includes its node influence In and its detection value Ug of abnormal data, and π is the size of $VNode[]$.

- **Step3** According to the detection of suspicious nodes in Step2, the current fog server makes reputation calculation to score each suspicious node in real time. We set that $X = \{n_1, n_2, ...n_m\}$ is a set of suspicious nodes, and RF_{n_i} is a reputation value for the suspicious node n_i with $1 \leq i \leq m$. RF_{n_i} is used to evaluate the trust value of each suspicious node in VANETs. The reputation value RF_{n_i} is calculated according to the abnormal detection of acquired data related to the suspicious node n_i and the satisfaction of the current fog server to acquired data.

Algorithm 2 Correlation Detection

Input: $VNode[]$, π
Output: R

Correlation-Detection($VNode[]$, π)
Begin
for $j = 0$ **to** $\pi - 1$ **do**
$key1 = VNode[j].In$;
$key2 = VNode[j].Ug$;
$i = j - 1$;
while $i \geq 0$ & $VNode[i].In > key1$ **do**
$VNode[i+1].In = VNode[i].In$;
$i = i - 1$;
End while
$VNode[i+1].In = key1$;
$i = j - 1$;
while $i \geq 0$ & $VNode[i].Ug > key2$ **do**
$VNode[i+1].Ug = VNode[i].Ug$;
$i = i - 1$;
End while
$VNode[i+1].Ug = key2$;
End for
$sum = 0$;
for $j = 0$ **to** $\pi - 1$ **do**
$d[j] = VNode[j].In - VNode[j].Ug$;
$sum+ = (d[j])^2$;
End for
$R = 1 - 6 \cdot \frac{sum}{\pi \cdot (\pi - 1)}$;
return R;
End

The calculation formula is described as follows:

$$RF_{n_i} = \sum_{j=1}^{N(n_i)} \left(\frac{S_j(f, n_i) \cdot (1 - I_{t_j}(n_i))}{N(n_i)}\right) - \frac{\omega_b \cdot ack_b + 1}{\omega_g \cdot ack_g + \omega_b \cdot ack_b + 1}, \quad (6)$$

where $S_j(f, n_i) = \frac{1}{U(G_{t_j})}$ is the satisfaction degree of the current fog server to the j-th data report from the suspicious node n_i[11], $I_{t_j}(n_i)$ is the influence of the suspicious node n_i at the time t_j, $N(n_i)$ is the number of data reports made by the suspicious node n_i (namely $N(n_i)$ is the number of received data reports of the current fog server from n_i), ack_b indicates the newest collaborative number of detections that a node is detected as the suspicious node[12], ack_g indicates the newest collaborative number of detections that a node is not detected as the suspicious node, $\omega_b = \frac{1}{1+e^{-ack_b}}$ is an acceleration factor and ω_g is a preset influence factor. This acceleration factor ω_b may cause the reputation value of the suspicious node to drop rapidly with increasing of the number that the node is considered to be suspicious. Thus it can prevent excessive punishment for a node due to one or two unintentional error data reports made by this node.

Theorem 1. *The value of RF_{n_i} belongs to $(-1, 1)$, where close to 1 denotes the node n_i has the highest reputation and close to -1 denotes the node n_i has the lowest reputation.*

Proof. Based on Lemma 1, we know $I_{t_j}(n_i) \in (0, 1)$. Also, we only compute the reputation of a suspicious node, thus we can obtain $U(G_{t_j}) \in (1, +\infty)$ related to the suspicious node n_i. Based on the Formula 6,

$$RF_{n_i} = \sum_{j=1}^{N(n_i)} \left(\frac{S_j(f, n_i) \cdot (1 - I_{t_j}(n_i))}{N(n_i)}\right) - \frac{\omega_b \cdot ack_b + 1}{\omega_g \cdot ack_g + \omega_b \cdot ack_b + 1},$$

we may get that[13]

– when $I_{t_j}(n_i) \to 1$,

$$\lim_{U(G_{t_j}) \to +\infty} \frac{S_j(f, n_i) \cdot (1 - I_{t_j}(n_i))}{N(n_i)} = \lim_{U(G_{t_j}) \to +\infty} \frac{\frac{1 - I_{t_j}(n_i)}{U(G_{t_j})}}{N(n_i)} = 0,$$

and

$$\lim_{ack_g \to 0, ack_b \to +\infty} \frac{\omega_b \cdot ack_b + 1}{\omega_g \cdot ack_g + \omega_b \cdot ack_b + 1} = 1,$$

thus $RF_{n_i} \to -1$;

– when $U(G_{t_j}) \to 1$,

$$\lim_{I_{t_j}(n_i) \to 0} \frac{S_j(f, n_i) \cdot (1 - I_{t_j}(n_i))}{N(n_i)} = \lim_{I_{t_j}(n_i) \to 0} \frac{\frac{1 - I_{t_j}(n_i)}{U(G_{t_j})}}{N(n_i)} = 1,$$

and

$$\lim_{ack_b \to 0, ack_g \to +\infty} \frac{\omega_b \cdot ack_b + 1}{\omega_g \cdot ack_g + \omega_b \cdot ack_b + 1} = 0,$$

thus $RF_{n_i} \to 1$.

□

Lemma 2. *If the node n_i is a suspicious node, then the change of its influence $I_{t_j}(n_i)$ is related to its reputation RF_{n_i}, where the bigger $I_{t_j}(n_i)$ is, the smaller RF_{n_i} is.*

Proof. Based on Theorem 1, if $I_{t_j}(n_i) \to 1$, then $RF_{n_i} \to -1$; otherwise if $I_{t_j}(n_i) \to 0$ and $ack_b \to 0$, then $RF_{n_i} \to 1$. □

Lemma 3. *If the node n_i is a suspicious node, then the change of its abnormal detection value $U(G_{t_j})$*

[11] $U(G_{t_j})$ is the local outlier factor of the data point $D(G_{t_j})$ according to Formula 4.

[12] When a node is detected as the suspicious node, the newest number of detections is based on collaborative detection of multiple fog servers in a certain time range. For example, a node is currently considered to be suspicious for a number of 2 by the last fog server, then the node leaves the management range of the last fog server; and the node is still considered to be suspicious for a number of 3 by the current fog server, thus the newest number that the node is considered to be suspicious is 5.

[13] We set $N(n_i) = 1$ for clearer description.

($U(G_{t_j}) > 1$) is related to its reputation RF_{n_i}, where the bigger $U(G_{t_j})$ is, the smaller RF_{n_i} is.

Proof. Similarly, based on Theorem 1, if $U(G_{t_j}) \rightarrow +\infty$, then $RF_{n_i} \rightarrow -1$. □

Thus, from Lemma 2, we may know if a node has greater influence for final acquired abnormal data, then its calculated reputation becomes smaller. Additionally, if a node has no influence for final acquired abnormal data and was not checked to be a suspicious node, then its calculated reputation becomes the biggest. From Lemma 3, we may know if a node is related to a greater detection value of abnormal data, then its calculated reputation becomes smaller. Lemma 2 and Lemma 3 will be confirmed by the following experiments. Based on the Formula 6, the current fog server makes reputation calculation to score each suspicious node for the detection of malicious vehicle nodes. We set that $N = \{N(x_1), N(x_2), ...N(x_m)\}$ is a set of the numbers of data reports made by the node $x_i \in Node\_set$ with $i \in [1, m]$, where $Node\_set$ is a set of related vehicle nodes in a period of time and m is the size of $Node\_set$. The detailed scoring procedure for each suspicious node is described as follows (and Algorithm 3):

step(a) For the node x_i, the current fog server records the number $N(x_i)$ of data reports made by the node x_i in the current period of time.

step(b) For the node x_i, the outlier detection is performed on the acquired data by the current fog server in real time. For all js with $1 \leq j \leq N(x_i)$, the current fog server computes $S_j(f, x_i) = \frac{1}{U(G_{t_j})}$ as the satisfaction degree of the current fog server to the j-th data report from the node x_i.

step(c) Based on Algorithm 1, for all js with $1 \leq j \leq N(x_i)$, the current fog server obtains the influence $I_{t_j}(x_i)$ of the node x_i at the time t_j.

step(d) Based on Algorithm 2, the current fog server obtains the correlation value R of the node x_i in the current period of time. If R is more than a pre-set value ϑ, then the node x_i is added to the set X of suspicious nodes, and the current fog server computes and records the local detection number $ack_{b_l} = ack_{b_l} + 1$; otherwise the current fog server computes and records the local detection number $ack_{g_l} = ack_{g_l} + 1$[14]. if $x_i \in X$, then the procedure continues; otherwise, the procedure will goto step(b) to check the next node.

step(e) For the current suspicious node $n_i \in X$ ($x_i = n_i$), the current fog server queries the newest numbers ack_{b_w} and ack_{g_w} about the suspicious node n_i from other fog servers if the current fog server did not query the newest numbers ack_{b_w} and ack_{g_w} after the suspicious node n_i enters into the management scope of the current fog server[15]. Then the current fog server computes ack_b=ack_{b_l}+ack_{b_w} and ack_g=ack_{g_l}+ack_{g_w} in the current period of time.

step(f) Repeat the above five steps to calculate the reputation value of each suspicious vehicle node.

Algorithm 3 Calculating Node Reputation

Input: $Node\_set$, m, ω_g
Output: $RF\_List = \{RF_{n_i}\}$

Reputation-Calculation($Node\_set$, m, ω_g)
Begin
  for $i = 1$ **to** m **do**
    for $j = 1$ **to** $N(x_i)$ **do**
      Computing $U(G_{t_j})$ related to x_i;
      Computing $I_{t_j}(x_i)$ based on **Algorithm 1**;
      $S_j(f, x_i) = \frac{1}{U(G_{t_j})}$;
      $U\_TList \leftarrow$**Add**($U(G_{t_j})$);
      $I\_TList \leftarrow$**Add**($I_{t_j}(x_i)$);
      $S\_TList \leftarrow$**Add**($S_j(f, x_i)$);
    End for
    Call Algorithm 2 to compute R based on U_TList and I_TList;
    if $R > \vartheta$ **then**
      $X \leftarrow$**Add**(x_i);
      $ack_{b_l} = ack_{b_l} + 1$;
    else
      $ack_{g_l} = ack_{g_l} + 1$;
      Break;
    End if
    if Is_queried(ack_{b_w}, ack_{g_w})==True **then**
      ack_b=ack_{b_l}+ack_{b_w};
      ack_g=ack_{g_l}+ack_{g_w};
    else
      Querying ack_{b_w}s and ack_{g_w}s of n_i from other fog servers;
      Obtaining the newest ack_{b_w} and ack_{g_w} according to the time;
      ack_b=ack_{b_l}+ack_{b_w};
      ack_g=ack_{g_l}+ack_{g_w};
    End if
    $\omega_b = \frac{1}{1+e^{-ack_b}}$;
    $RF_{x_i} = 0$;
    for $j = 1$ **to** $N(x_i)$ **do**
      $I_{t_j}(x_i) \leftarrow$**Get**($I\_TList[j]$);
      $S_j(f, x_i) \leftarrow$**Get**($S\_TList[j]$);
      $RF_{x_i} = RF_{x_i} + \frac{S_j(f,x_i)\cdot(1-I_{t_j}(n_i))}{N(x_i)}$;
    End for
    $RF_{x_i} = RF_{x_i} - \frac{\omega_b \cdot ack_b + 1}{\omega_g \cdot ack_g + \omega_b \cdot ack_b + 1}$;
    $RF\_List \leftarrow$**Add**(RF_{x_i});
  End for
  return RF_List;
End

[14]The local detection numbers ack_{b_l} and ack_{g_l} are initially set to 0.

[15]In this paper, we decide whether the numbers ack_{b_w} and ack_{g_w} are the newest according to the time.

- **Step4** Each node is detected in real-time in a period of time. When a node is considered as a suspicious node, the reputation of the suspicious node is scored in real-time through the Algorithm 3. From Theorem 1, we may know that the value of RF_{n_i} belongs to $(-1, 1)$. Thus, if the reputation of a suspicious node is lower than a preset threshold value α (such as $\alpha = 0$) in a fixed duration[16], then the node is determined to be a malicious node; otherwise, it is still considered to be trusted.

V. Experimental analysis

A. Experiment environment

We make some experiments to test the performance of our proposed scheme through the CloudSim simulation platform, where we use JAVA to code our proposed algorithms. In the experiments, the test environment is under Win10, Intel i5 CPU and 8G RAM. We use the extended CloudSim platform to simulate a VANET environment based on fog computing. Each vehicle node in the simulated environment is used as a fog node to send (and forward) continuous data or logical data to the fog server. Also, for the dynamicity of VANETs, we change the configure of network topology according to the time. Based on the behaviors of the vehicle nodes, these nodes are classified into the following types:

- Honest nodes: such nodes send (or forward) true data to the fog server;
- Dishonest nodes: such nodes are further divided into the following subclasses:
  - Common malicious nodes: such nodes always and only provide untrue data to the fog server in a period of time;
  - Colluded malicious nodes: such nodes form a malicious group to reduce the credibility of the honest nodes and confuse the fog server when the final data is transferred to the fog server;
  - Variant malicious nodes[17]: such nodes alteratively provide untrue data to the fog server in a period of time, namely such nodes sometimes provide untrue data to the fog server and sometimes provide true data to the fog server. Because the detection of malicious nodes based on the correlation of acquired data and network topology is related to time, the different behaviors of such nodes in a period of time may influence the accuracy of detection.

The related experimental parameters are set as the following Table 3.

[16]In our proposed scheme, the value of the reputation RF_{n_i} is mapped into the interval $(-1, 1)$, thus we use the middle value of this interval as the preset threshold value. Namely the value of 0 is used as a boundary point. In fact, we consider that the preset threshold value may be adjusted according to the actual security requirements in different applications. If some applications require high security, then the preset threshold value may be set to a value greater than 0, such as 0.1.

[17]In our proposed experiment, when the experimental program simulates the fog servers to collect data, these malicious nodes can randomly provide the preset standard data (correct data) to the fog servers, or can randomly provide the wrong data to the fog servers in the dynamic network topology at the next time.

TABLE III
Related experimental parameters

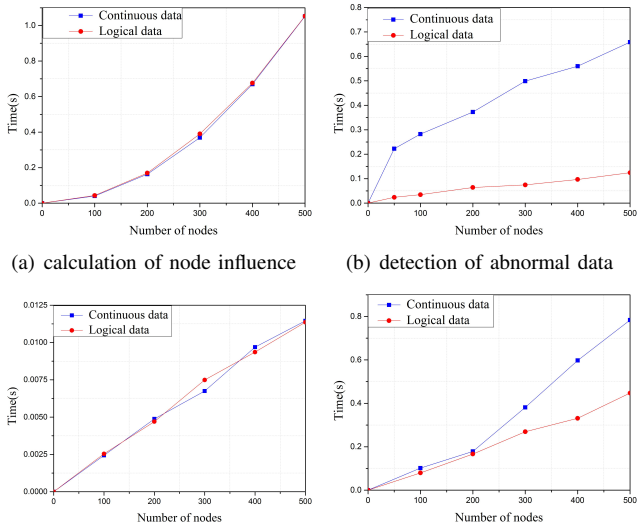
Parameters	Value range
Initial reputation of node	0
Preset correlation value (ϑ)	0.96
Historical honest factor (ω_g)	0.5
Number of nodes	20, 50, 100, 200,...500
Proportion of dishonest nodes	0.1, 0.2, 0.3, 0.4, 0.5

In order to check the efficiency of our proposed scheme, we first test the running times of calculation of node influence, detection of abnormal data, detection of suspicious nodes and detection of malicious nodes under different number of vehicle nodes. Second, for the effectiveness of our proposed scheme, we test the correlation between node influence and abnormal data and the correlation between node reputation and node influence. Further, we test the reputation change of the different types of nodes and the detection accuracy of malicious nodes (and suspicious nodes) under the different proportion of dishonest nodes. Also, based on the detection rate of malicious nodes, we compare our proposed scheme with other related works [7,8,25].

B. Simulation test and analysis

1) Running time:

In the section, we test the running times of calculation of node influence, detection of abnormal data, detection of suspicious nodes and detection of malicious nodes under different number of vehicle nodes. In the experiment, the number of vehicle nodes is changed from 0 to 500. Also, we make the tests on the two different types of data (logical data and continuous data), where we mainly observe the effect of different types of data on the running times. The experimental results are shown in Figure 7.



(a) calculation of node influence (b) detection of abnormal data

(c) detection of suspicious nodes (d) detection of malicious nodes

Figure 7 The running times of the proposed algorithms

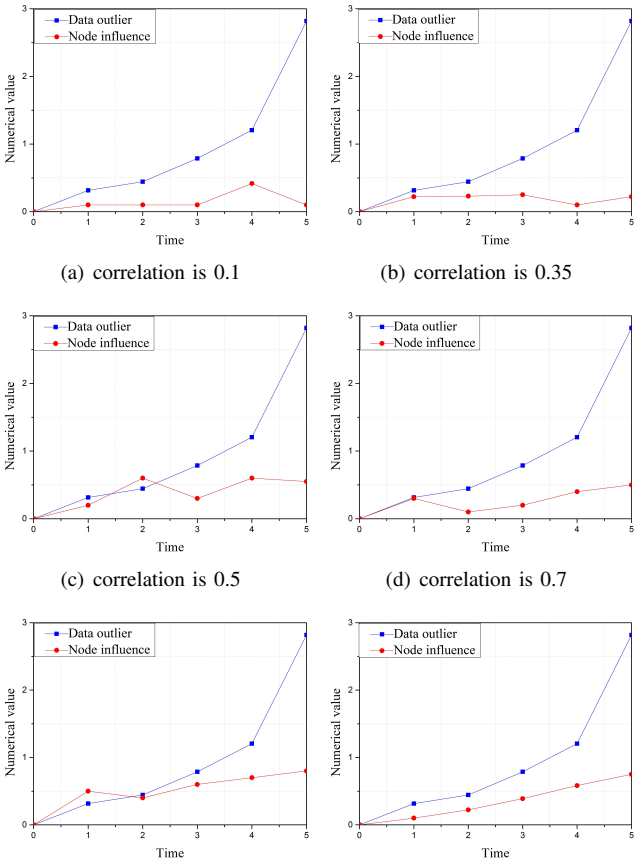
In the Figure 7, all the running times are increasing with the number of vehicle nodes. Figure 7(a) shows the running times

of calculation of node influence for the two different types of data. It can be seen from Figure 7(a) that the two running times are almost identical. That is because calculation of node influence is related to network topology, not to data type. Figure 7(b) shows the running times of detection of abnormal data for the two different types of data. It can be seen from Figure 7(b) that there is a big gap between the two running times. That is because logical data is represented as a binary system in which a binary sequence represents an inquired result. Figure 7(c) shows the running times of detection of suspicious nodes for the two different types of data. Similarly, it can be seen from Figure 7(c) that the two running times are almost identical. Figure 7(d) shows the running times of detection of malicious nodes for the two different types of data. It can be seen from Figure 7(d) that there is a small gap between the two running times. We consider this is because the value change of continuous data is more diverse compared with logical data. So, the detection time of continuous data is bigger than that of logical data. On the whole, we may know all the running times of the proposed algorithms are very small.

2) Correlation between node influence and data outlier:

In the section, we mainly evaluate the correlation between node influence and data outlier generated by node. Also, based on the different experiment datasets, we test the correlation for the two different types of data. Figures 8 and 9 show the correlation changes, where x-axis denotes the time duration[18] and y-axis denotes the values of node influence and data outlier. Figure 8 shows the correlations for continuous data between node influence and data outlier over time. Figure 8(a) shows the correlation change between node influence and data outlier when the correlation is 0.1. In Figure 8(a), the changes of node influence and data outlier only have the same tendency at the times 0-1 and 3-4, and there is no same tendency at other times. For example, at the time 4-5, node influence is reducing while the value of data outlier is increasing. Figure 8(b) shows the correlation change when the correlation is 0.35. In Figure 8(b), the changes of node influence and data outlier only have the same tendency at the times 0-1 and 4-5, and there is no same tendency at other times. Figure 8(c) shows the correlation change when the correlation is 0.5. In Figure 8(c), the changes of node influence and data outlier only have the same tendency at the times 0-2 and 3-4, and there is no same tendency at other times. Obviously, the time durations of the same tendency between node influence and data outlier are increasing. Figure 8(d) shows the correlation change when the correlation is 0.7. In Figure 8(d), the changes of node influence and data outlier have the same tendency at the times 0-1 and 2-5. Figure 8(e) shows the changes of node influence and data outlier have the same tendency at the times 0-1 and 2-5 when the correlation is 0.9. Figure 8(f) shows the changes of node influence and data outlier have the same tendency at all the times 0-5 when the correlation is 1. Because the changes of node influence and data outlier have the same tendency, the corresponding node is suspicious when the correlation is 1.

Figure 9 shows the correlations for logical data between



(a) correlation is 0.1 (b) correlation is 0.35

(c) correlation is 0.5 (d) correlation is 0.7

(e) correlation is 0.9 (f) correlation is 1

Figure 8 The correlation changes on continuous data

node influence and data outlier over time. Figure 9(a) shows the correlation change between node influence and data outlier when the correlation is 0.05. In Figure 9(a), the changes of node influence and data outlier only have the same tendency at the times 0-2 and 3-4, and there is no same tendency at other times. For example, at the time 2-3, node influence suddenly drops while the value of data outlier is still increasing. Figure 9(c) shows the changes of node influence and data outlier only have the same tendency at the times 0-1 and 4-5 when the correlation is 0.55, and there is no same tendency at other times. Figure 9(d) shows the changes of node influence and data outlier have the same tendency at the times 0-2 and 3-5 when the correlation is 0.75. Similarly, from Figure 9(a) to Figure 9(d), the time durations of the same tendency between node influence and data outlier are increasing. Figure 9(f) shows that the changes of node influence and data outlier have the same tendency at all the times 0-5 when the correlation is 1. So, the corresponding node is suspicious when the correlation is 1.

Based on the above Figures 8 and 9, we may evaluate the correlations between node influence and data outlier for the two different types of data. When the correlation reaches 1, the changes of node influence and data outlier have the same tendency at all the times, thus we may consider the corresponding node is suspicious.

Additionally, we show some detailed values for the different correlations under the two different types of data in Tables 4

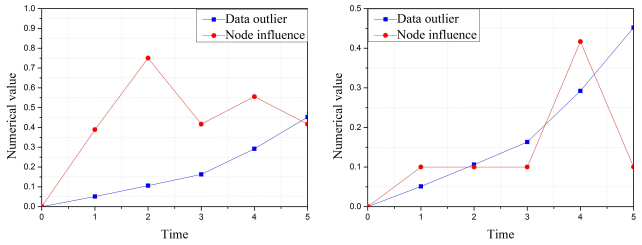
[18]We downsize the time duration as 0-5 and each time point is an integer index.

TABLE IV
DETAILED VALUES FOR THE DIFFERENT CORRELATIONS (< 1)

	Continuous data			Logical data		
Time	Correlation	Node influence	Data outlier	Correlation	Node influence	Data outlier
1	0.1	0.1	0.315	0.05	0.1	0.051
2		0.1	0.444		0.1	0.106
3		0.1	0.787		0.1	0.163
4		0.41667	1.205		0.41667	0.292
5		0.1	2.819		0.1	0.452
1	0.35	0.22222	0.315	0.1	0.38889	0.051
2		0.23	0.444		0.75	0.106
3		0.25	0.787		0.41667	0.163
4		0.1	1.205		0.55556	0.292
5		0.22222	2.819		0.41667	0.452

TABLE V
DETAILED VALUES FOR THE CORRELATION$= 1$

	Continuous data			Logical data		
Time	Correlation	Node influence	Data outlier	Correlation	Node influence	Data outlier
1	1.0	0.1	0.315	1.0	0.1	0.051
2		0.22222	0.444		0.22222	0.106
3		0.38889	0.787		0.38889	0.163
4		0.58333	1.205		0.58333	0.292
5		0.75	2.819		0.75	0.452
6		0.88	10.433		0.8	3.17
7		0.75	8.047		0.6	2.25
8		0.616	1.205		0.55555	1.27
9		0.555	0.686		0.2222	0.83
10		0.416666	0.444		0.1	0.09



(a) correlation is 0.05 (b) correlation is 0.1

(c) correlation is 0.55 (d) correlation is 0.75

(e) correlation is 0.9 (f) correlation is 1

Figure 9 The correlation changes on logical data

and 5. Table 4 shows the value changes of node influence and data outlier on continuous data when the correlations respectively are 0.1 and 0.35, and the value changes of node influence and data outlier on logical data when the correlations respectively are 0.05 and 0.1. In Table 4, the change of node influence is not obviously related to that of data outlier. For example, the influence values 0.1, 0.1, 0.1, 0.41667 and 0.1 related to the same node in the corresponding network topologies are not correlated with the corresponding values of data outliers, thus the correlation is 0.1 for continuous data. Similarly, when the correlation is 0.35, the influence values related to the other same node are also not correlated with the corresponding values of data outliers. Table 5 shows the changes of node influence and data outlier on continuous data and logical data when the correlation is 1. In Table 5, the change of node influence is consistent to that of data outlier.

3) Relation among node influence, node reputation and detection number of suspicious nodes:

In the section, we change node influence in a time duration downsized as 0-5, and then observe the relation change among node influence, node reputation and detection number of suspicious nodes. Figures 10 and 11 show the tested average results based on continuous data and logical data respectively. In the figures, with the increase of node influence and detection number of suspicious nodes, node reputation will decrease.

For dishonest nodes, when they are at different positions in a dynamic network, they have different influences to interfere final results. So, the greater node influence is, the greater abnormal detection value is. The previous experiment has confirmed this change. Also, as long as the influence of a node

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCC.2020.2985050, IEEE Transactions on Cloud Computing

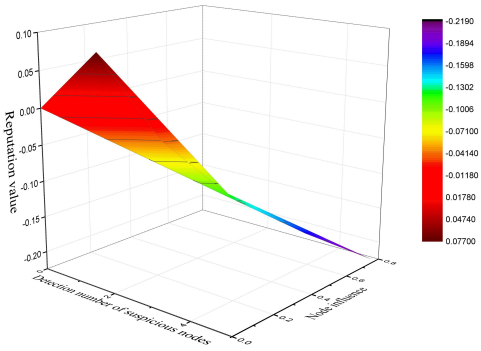


Figure 10 The change of node reputation related with node influence and detection number of suspicious nodes on continuous data

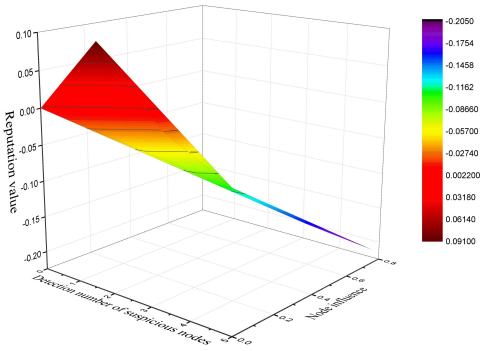


Figure 11 The change of node reputation related with node influence and detection number of suspicious nodes on logical data

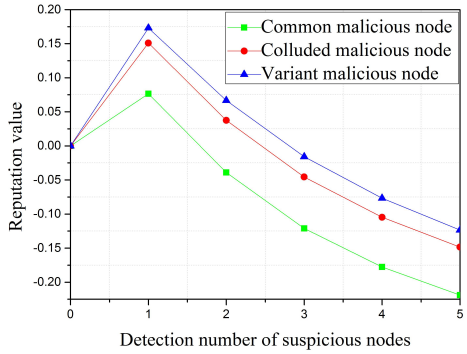


Figure 12 The reputation change of three types of dishonest node on continuous data

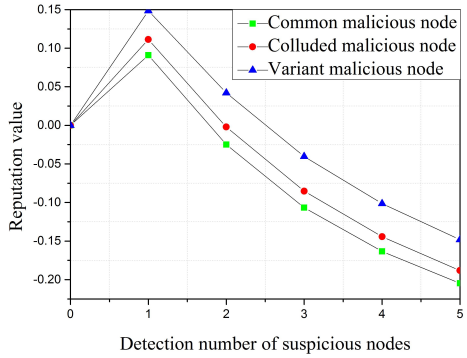


Figure 13 The reputation change of three types of dishonest node on logical data

is checked to be correlated with its abnormal detection value, the node can be considered to be suspicious. Further, when node influence and detection number of suspicious nodes both become large, node reputation becomes small. So, this change coincides with Lemma 2 and Lemma 3. Additionally, with the increase of node influence and detection number of suspicious nodes, node reputation is gradually decreasing and fluctuates in a lower reputation range. For example, Figure 10 shows that node reputation is gradually changing to negative value (less than -0.2). Therefore, when the reputation of a node is always less than 0 in a time duration, we may consider it is a malicious node.

4) Reputation changes for three types of dishonest node:

In the section, we test the reputation changes for three types of dishonest node, which include common malicious node, colluded malicious node and variant malicious node. Because the three types of dishonest node have different behaviours to damage transferred data in VANETs, it is necessary to test the effectiveness of our proposed scheme for detecting the three types of dishonest node. Figures 12 and 13 show the reputation changes based on continuous data and logical data respectively. It can be seen from the figures that the reputations of three types of dishonest node are both decreasing along with the corresponding detection numbers of suspicious nodes. Since variant malicious nodes sometimes provide untrue data to the fog server and sometimes provide true data to the fog server, this behaviour inhibits the real evaluation of variant malicious nodes to a certain extent. So, the overall reputations of such nodes are higher than those of common malicious nodes and colluded malicious nodes. However, as time goes by, the detection number of suspicious nodes for such nodes will increase, then the reputations of such nodes are decreasing to a lower reputation range.

Additionally, the reputations of colluded malicious nodes are also higher than those of common malicious nodes because of the existence of colluded "partners". Compared with common malicious nodes, colluded malicious nodes can provide more false data because of their "partners", thus the detection values of abnormal data are lower so that the reputations of such nodes can maintain relatively high in a period of time. However, as the detection number of suspicious nodes increases, the reputations of such nodes are gradually decreasing to a certain area (about -0.14 in Figure 12 and -0.18 in Figure 13). Also, compared with other two types of dishonest node, the overall reputations of common malicious nodes are lower. That is because no colluded "partners" and masquerading behaviours exist for such nodes.

5) Detection accuracy:

In the section, we respectively test the accuracy of detecting suspicious nodes and malicious nodes. Also, we compare our

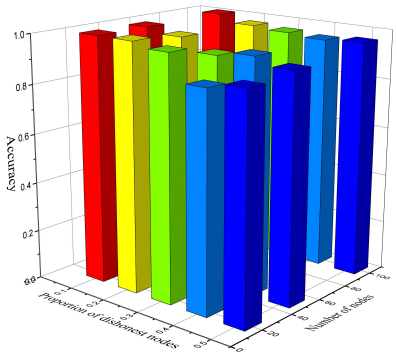


Figure 14 The detection accuracy of suspicious nodes on continuous data

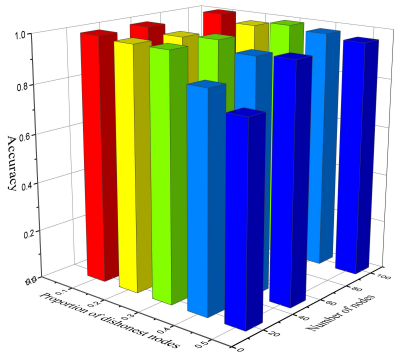


Figure 15 The detection accuracy of suspicious nodes on logical data

proposed scheme with other related works [7,8,25]. In the experiment, we set three different network topologies composed of 20, 50, 100 vehicle nodes respectively. Additionally, we test 30 times for each network topology under the settings of 10%, 20%, 30%, 40% and 50% proportions of dishonest nodes, where the proportion of common malicious nodes, colluded malicious nodes and variant malicious nodes in all dishonest nodes is 2:1:1. Figures 14 and 15 show the tested results for detecting suspicious nodes on continuous data and logical data respectively. In the Figures 14 and 15, the accuracy of the proposed detection algorithm can be affected by the proportion of dishonest nodes. Thus the larger the proportion of dishonest nodes is, the lower the detection accuracy of suspicious nodes is. Because the proportion of dishonest nodes becomes larger, more dishonest nodes may provide false (abnormal) data[19]. Then, because the detection value of abnormal data gradually decreases, the fluctuation of abnormal value becomes small so that the correlation coefficient becomes small. Figures 14 and 15 show that the detection accuracy of suspicious nodes decreases with the increase of the number of dishonest nodes. On the whole, the detection rate of suspicious nodes based on our proposed algorithm is high. Especially when the proportion of dishonest nodes is less than 30%, the overall detection rate remains around 90%.

Figures 16 and 17 show the tested results for detecting malicious nodes on continuous data and logical data respectively. Similarly, in the Figures 16 and 17, the accuracy of the proposed detection algorithm can also be affected by the proportion of dishonest nodes. The detection accuracy of malicious nodes decreases with the increase of the number of dishonest nodes. It can be seen from Figures 16 and 17 that when the proportion of dishonest nodes is less than 30%, the overall detection rate of malicious nodes remains around 90%. Compared with the detection rate of suspicious nodes, the detection rate of malicious nodes relatively decreases.

[19]In our work, if the proportion of dishonest nodes is close to 45%, then the detection of abnormal data needs to be based on the context of related data. Namely we need to assume that the detection of abnormal data can be finished based on a large amounts of data generated over a longer period of time when the proportion of dishonest nodes becomes larger.

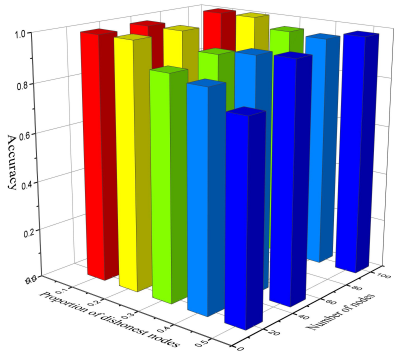


Figure 16 The detection accuracy of malicious nodes on continuous data

Because the number of dishonest nodes increases, it will affect the detection of abnormal data. When the detection value of abnormal data becomes small, the calculated reputation value will directly become high according to Formula 6. So, the detection rate of malicious nodes decreases. However, our proposed scheme first judges whether a node is suspicious by computing correlation, then the suspicious node can be recorded. So, the suspicious node still has high probability to be detected as a malicious node.

Additionally, for the detection accuracy of malicious nodes, we compare our proposed scheme with the HMM scheme proposed by [25], the iTrust scheme proposed by [7] and the ART scheme proposed by [8]. In the experiment, we unify the test parameters for the four schemes, and we set the network composed of 50 vehicle nodes. Also, we test 30 times for the different network topology under the settings of 10%, 30% and 50% proportions of dishonest nodes. The tested results are shown in Figure 18. Figure 18 shows the detection accuracies of the four schemes. From the experimental results, we may know that the accuracies of the four schemes are high. With the increase of the proportion of dishonest nodes (changing 10% to 50%), their accuracies will both decrease. It can be seen from Figure 18 that the detection accuracy of the iTrust scheme is

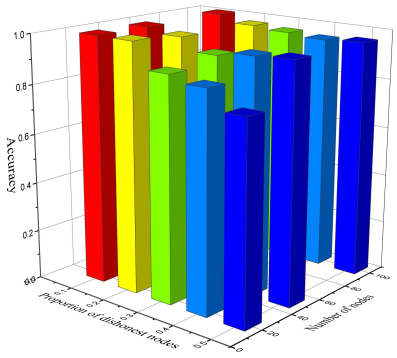


Figure 17 The detection accuracy of malicious nodes on logical data

the highest and the detection accuracy of our proposed scheme is second, where the detection accuracy of the HMM scheme is not stable. Also, we further consider the iTrust scheme is not suitable for VANETs because more test data and time cost are needed for checking wrong behaviors. On the whole, when the proportion of dishonest nodes is less than 30%, the accuracies of the four schemes both remain around 80%, where our scheme can remain around 90%. So, our proposed scheme is effective for detecting malicious nodes.

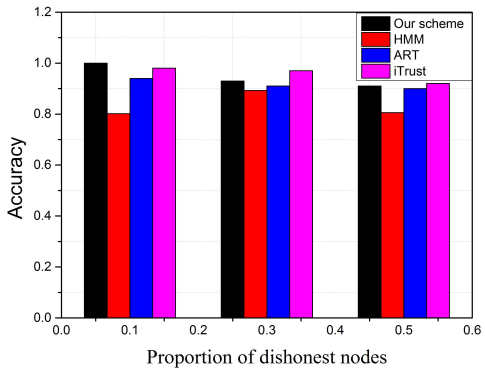


Figure 18 The detection accuracy comparison of four schemes

VI. Conclusions

The behaviours of malicious nodes may cause many serious traffic accidents in VANETs. Therefore, how to detect out malicious vehicle nodes by some lightweight methods needs to be researched in VANETs. In this paper, we propose a malicious node detection scheme in fog computing-based VANETs, where the fog server uses the reputation calculation to score each suspicious node based on the correlation of acquired data and network topology. In our scheme, we show how to calculate node influence in a network topology and propose an outlier detection approach to detect abnormal data for continuous data in VANETs. Further, based on the correlation between outlier detection of acquired data and influence of nodes, we propose a reputation mechanism to score each suspicious node. According to our experimental results, our proposed scheme can effectively detect out malicious vehicle nodes so that fog server can acquire more true data. However, because our scheme needs a period of real-time data and a small amount of historical data, it is difficult to fast detect malicious nodes in real time. Additionally, the fog server needs to acquire data and related dynamic network topology at the same time, thus the vehicle nodes need to pay extra cost for transferring their data and network positions to the fog server. For the previous mentioned disadvantages, we still need to do more in-depth research to solve them in the future work.

Acknowledgment

This study was funded by the National Natural Science Foundation of China (No.61402055, No.61902040, No.61972056), the Natural Science Foundation of Hunan Province (No.2018JJ2445), the FDCT Project (No.0007/2018/A1), the DCT-MoST Joint Project of SAR Macau (No.FDCT/025/2015/AMJ), the Funds of University of Macau (No.CPG2018-00032-FST, No.SRG2018-00111-FST), the National Natural Science Key Foundation of China (No.61532013), the National China 973 Project (No.2015CB352401) and the Shanghai Scientific Innovation Act of STCSM (No.15JC1402400).

References

[1] M. Sichitiu, M. Kihl. Inter-vehicle communication systems: A survey[J]. IEEE Communications Surveys & Tutorials, 2008,10(2): 88-105.
[2] E. Schoch, F. Karge, M. Weber, T. Leinmuler. Communication patterns in VANETs[J]. IEEE Communication Magazine,2008,46(11):119-125.
[3] M. Raya, JP. Hubaux. The security of vehicular ad hoc networks[C]//Proceedings of the 3rd ACM workshop on Security of ad hoc and sensor networks. ACM, 2005: 11-21.
[4] M. Mukherjee, R. Matam, L. Shu, et al. Security and privacy in fog computing: Challenges[J]. IEEE Access, 2017, 5: 19293-19304.
[5] F. Bonomi, Connected vehicles, the internet of things, and fog computing[C], The eighth ACM international workshop on vehicular inter-networking (VANET), Las Vegas, USA. 2011: 13-15.
[6] C. Huang, R. Lu, and KKR. Choo Vehicular Fog Computing: Architecture, Use Case, and Security and Forensic Challenges[J]. IEEE Communications Magazine, 2017, 55(11): 105-111.
[7] H. Zhu, S. Du, Z. Gao, et al. A probabilistic misbehavior detection scheme toward efficient trust establishment in delay-tolerant networks[J]. IEEE Transactions on Parallel and Distributed Systems, 2014, 25(1): 22-32.
[8] W. Li, H. Song. ART: An attack-resistant trust management scheme for securing vehicular ad hoc networks [J]. IEEE Transactions on Intelligent Transportation Systems, 2016, 17(4): 960-969.
[9] F. Wang, C. Huang, J. Zhang, and C. Rong, IDMTM: A novel intrusion detection mechanism based on trust model for ad hoc networks[C], 22nd IEEE International Conference on Advanced Information Networking and Applications, 2008, pp. 978-84.
[10] P. Ebinger and N. Bibmeyer, TEREC: Trust evaluation and reputation exchange for cooperative intrusion detection in MANETs[C], 7th Annual Comm. Networks and Services Research Conf.2009, pp. 378-385.
[11] S. Ganeriwal, LK. Balzano, and MB. Srivastava, Reputation-based framework for high integrity sensor networks[J], ACM Transitions on Sensor Network, vol. 4, no. 3, May 2008.
[12] K. Liu, N. Abu-Ghazaleh, and KD. Kang, Location verification and trust management for resilient geographic routing[J], J. Parallel and Distributed Computing, vol. 67, no. 2, 2007, pp. 215-28.
[13] R. A. Shaikh, H. Jameel, B. J. dAuriol, H. Lee, S. Lee, and Y. J. Song, Group-based trust management scheme for clustered wireless sensor networks[J], IEEE Transactions on Parallel and Distributed Systems, vol. 20, no. 11, Nov. 2009, pp. 1698-1712.
[14] S. Althunibat, A. Antonopoulos, E. Kartsakli. Countering Intelligent Dependent Malicious Nodes in Target Detection Wireless Sensor Networks[J]. IEEE Sensors Journal, 2016:1-13.

[15] M. Abdelhakim, L. Lightfoot, J. Ren. Distributed Detection in Mobile Access Wireless Sensor Networks under Byzantine Attacks[J]. IEEE Transactions on Parallel and Distributed Systems, 2014, 25(4):950-959.
[16] A. Vempaty, O. Ozdemir, K. Agrawal. Localization in wireless sensor networks: Byzantines and mitigation techniques[J]. IEEE Transactions on Signal Processing, 2012, 61(6): 1495-1508.
[17] M. Raya, P. Papadimitratos, I. Aad, D. Jungels, and J. P. Hubaux. Eviction of Misbehaving and Faulty nodes in Vehicular Networks[J]. IEEE Journal on Selected Areas in Communications, Special Issue on Vehicular Networks, 25(8):1557-1568, 2007.
[18] T. Leinmuller, E. Schoch, and F. Kargl. POSITION VERIFICATION APPROACHES FOR VEHICULAR AD HOC NETWORKS[J]. Wireless Communications, IEEE, 13(5):16-21, october 2006.
[19] T. Leinmuller, E. Schoch, F. Kargl, and C. Maihfer. Influence of falsified position data on geographic ad-hoc routing[C]. In2nd European Workshop on Security and Privacy in Ad hoc and Sensor Networks (ESAS 2005), pages 102-112, 2005.
[20] G. Yan, S. Olariu, MC. Weigle. Providing VANET Security Through Active Position Detection[J]. Computer communications, 2008, 31(12): 2883-2897.
[21] G. Yan, S. Olariu, and M. C. Weigle. Providing location security in vehicular Ad Hoc networks. Wireless Communications[J], 16(6):48-55, December 2009.
[22] B. Xiao, B. Yu, and C. Gao. Detection and localization of Sybil nodes in VANETs. [C]//Proceedings of the 2006 workshop on Dependability issues in wireless ad hoc networks and sensor networks. ACM, 2006: 1-8.
[23] S. Lv, X. Wang, X. Zhao, and X. Zhou. Detecting the Sybil Attack Cooperatively in Wireless Sensor Networks[C]. InCIS 08: Proceedings of the 2008 International Conference on Computational Intelligence and Security, pages 442-446, Washington, DC, USA, 2008. IEEE Computer Society.
[24] S. Dhurandher, M. Obaidat, A. Jaiswal. Vehicular Security Through Reputation and Plausibility Checks[J]. IEEE Systems Journal, 2014, 8(2):384-394.
[25] MGH. Al Zamil, S. Samarah, M. Rawashdeh. False Alarm Detection in Fog-Based Internet of Connected Vehicles[J]. IEEE Transactions on Vehicular Technology, 2019, 68(7): 7035-7044.
[26] J. Pereira, L. Ricardo, M. Luís, C. Senna, S. Sargento. Assessing the reliability of fog computing for smart mobility applications in VANETs[J]. Future Generation Computer Systems, 2019, 94: 317-332.
[27] S. A. Soleymani, A. H. Abdullah, M. Zareei, et al. A Secure Trust Model based on Fuzzy Logic in Vehicular Ad Hoc Networks with Fog Computing[J]. IEEE Access, 2017, 5:15619-15629.
[28] J. Ni, A. Zhang, X. Lin, et al. Security, Privacy, and Fairness in Fog-Based Vehicular Crowdsensing[J]. IEEE Communications Magazine, 2017, 55(6): 146-152.
[29] S. Yi, Z. Qin, Q. Li. Security and Privacy Issues of Fog Computing: A Survey[M]// Wireless Algorithms, Systems, and Applications. Springer International Publishing, 2015:685-695.
[30] K. Lee, D. Kim, D. Ha, et al. On security and privacy issues of fog computing supported Internet of Things environment[C]// Network of the Future. IEEE, 2015:1-3.
[31] I. Stojmenovic, S. Wen. The Fog computing paradigm: Scenarios and security issues[C]// Computer Science and Information Systems. IEEE, 2014:1-8.
[32] P. Hu, H. Ning, T. Qiu, et al. Security and Privacy Preservation Scheme of Face Identification and Resolution Framework Using Fog Computing in Internet of Things[J]. IEEE Internet of Things Journal, 2017,4:1143-1155.
[33] P. Hu, H. Ning, T. Qiu, et al. Fog Computing-Based Face Identification and Resolution Scheme in Internet of Things[J]. IEEE Transactions on Industrial Informatics, 2017,13:1910-1920.
[34] O. Osanaiye, S. Chen, Z. Yan, et al. From cloud to fog computing: A review and a conceptual live VM migration framework[J]. IEEE Access, 2017, 5:8284-8300.
[35] K. Zhang, Y. Mao, S. Leng, et al. MOBILE-EDGE COMPUTING FOR VEHICULAR NETWORKS A Promising Network Paradigm with Predictive Off-Loading[J]. IEEE Vehicular Technology Magazine, 2017, 12(2): 36-44.
[36] J. He, J. Wei, K. Chen, et al. Multi-tier fog computing with large-scale IoT data analytics for smart cities[J]. IEEE Internet of Things Journal, 2018, 5(2):677-686.
[37] A. Giordano, G. Spezzano, A. Vinci. Smart agents and fog computing for smart city applications[C]. International Conference on Smart Cities, Springer International Publishing, 2016: 137-146.
[38] A. Daeinabi, A. Rahbar. Detection of malicious vehicles (DMV) through monitoring in Vehicular Ad-Hoc Networks[J]. Multimedia Tools and Applications, 2013, 66(2):325-338.
[39] X. Zhang, C. Lyu, Z. Shi. Reliable Multiservice Delivery in Fog-Enabled VANETs: Integrated Misbehavior Detection and Tolerance[J]. IEEE Access, 2019, 7:95762-95778.
[40] S. P. Borgatti, M. G. Everett. A graph-theoretic perspective on centrality[J]. Social networks, 2006, 28(4): 466-484.
[41] M. M. Breunig, HP. Kriegel, RT. Ng, et al. LOF: identifying density-based local outliers[C]. ACM sigmod record, ACM, 2000, 29(2): 93-104.
[42] S. Yue, P. Pilon, G. Cavadias. Power of the MannCKendall and Spearman's rho tests for detecting monotonic trends in hydrological series[J]. Journal of Hydrology, 2002, 259(1):254-271.

Ke Gu received his Ph.D. degree in School of Information Science and Engineering from Central South University, Changsha, China in 2012. He joined the School of Computer & Communication Engineering at Changsha University of Science and Technology, in 2013. Currently he is an Associate Professor. His research interests include network and information security. He has published more than 60 research papers in journals or conferences.

XinYing Dong is pursuing her Master degree in School of Computer & Communication Engineering at Changsha University of Science and Technology, Changsha, China. Her research interests include network and information security.

Weijia Jia received the Ph.D. degree in computer science from the Polytechnic Faculty of Mons, Mons, Belgium in 1993. He is currently a Full Professor with Department of Computer and Information Science, University of Macau. His research interests include next-generation wireless communication, distributed systems, and multicast and anycast routing protocols. He has published more than 100 research papers in famous journals or conferences.