Journal of Systems Architecture 119 (2021) 102219

Contents lists available at ScienceDirect

Journal of Systems Architecture

journal homepage: www.elsevier.com/locate/sysarc

A lightweight verifiable trust based data collection approach for sensor–cloud systems

Jiawei Guo [a], Haoyang Wang [a], Wei Liu [b,*], Guosheng Huang [c], Jinsong Gui [a], Shaobo Zhang [d]

[a] School of Computer Science and Engineering, Central South University, ChangSha 410083, China
[b] School of Informatics, Hunan University of Chinese Medicine, Changsha 410208, China
[c] School of Information Science and Engineering, Hunan First Normal University, Changsha 410205, China
[d] School of Computer Science and Engineering, Hunan University of Science and Technology, Xiangtan, Hunan, China

ARTICLE INFO

Keywords:
Sensor–cloud systems
Trustworthiness
Data collection
Mobile vehicles
Unmanned aerial vehicles

ABSTRACT

A Lightweight Verifiable Trust based Data Collection (LVT-DC) approach is proposed to obtain credible data for mobile vehicles network, in which, a large number of IoT devices are deployed in smart city, and Mobile Vehicles (MVs) collect data from IoT devices and report data to cloud to construct various applications. First, some IoT devices are selected as core-IoT device whose data more than others. Then, Unmanned Aerial Vehicles (UAV) delivers a short verification code to each core-IoT devices and collects its data, and it embeds verification code into the data when the MVs collect it in order for cloud to check. Last, a verifiable trust inference method is proposed which includes two types of trust calculations. (a) Direct trust: MVs If the reported data cannot recover the verification code or is inconsistent with the UAV collection result, reduce the trust of the MVs, (b) Trust inference: If the data of a non-core-IoT devices reported by a MVs is inconsistent with the trusted MVs, then reduce its trust. After a large number of experimental results, the LVT-DC approach can quickly and accurately identify the credibility of the data collector and ensure credible data collection.

1. Introduction

With the development of microprocessor technology, the current Internet of Things (IoT) has developed rapidly [1,2]. Currently, IoT devices with communication and sensing capabilities connect to the Internet have exceeded 20 billion units [1] and up to 2.5 quintillion bytes data every day [3] are sensed by IoT devices which facilitates distributed IoT applications such as Vtrack [4] and NoiseTube [5]. Wireless Sensor Networks combined Cloud Computing which is called Sensor–Cloud System (SCS) have received tremendous attention [6–8]. In SCS, a large number of sensing nodes are deployed in the area that needs to be monitored to sense and monitor the objects that need to be observed. Because of the huge number of these sensing devices, they can sense rich data [9–11]. However, these sensing devices have weak storage and computing capabilities [12–14], so they often need to send their sensed data to the cloud with powerful computing capabilities for processing [15–17], However, these sensing devices have weak storage and computing capabilities [12–14], so they often need to send their sensed data to the cloud with powerful computing capabilities for processing [15–17]. For this reason, SCS makes the network and the edge sensing devices are combined with the computing power of the cloud to give full play to their respective advantages, so that SCS has been greatly developed [6].

Mobile vehicles network is one of important SCS, especially for smart city [18–20]. Bonala et al. [21] proposed an opportunistic routing data collection network for mobile vehicles network. In such a network, various sensing devices can be deployed on demand in the areas that need to be monitored. In this regard, the proposed scheme is easier to promote than Wireless Sensor Networks (WSNs) [22,23]. In WSNs, many sensor nodes can only be deployed in a certain area. Then, this requires a base station connected to the Internet, and the based station is generally an active power supply, which requires the deployment of a basic power connection and network connection, so It limits its development [24,25]. In the scheme proposed by Bonola et al. [21], smart sensor nodes are deployed on the infrastructure that needs to be monitored in the city, such as street lights on both sides of the road, trash cans, etc, and data are generated when the state of these facilities changes. When Mobile Vehicles (MVs) pass through their communication range, they can transmit data to mobile vehicles [21]. Not only can a single sensor deivces do this, but multiple sensor nodes can also be organized into a network, The sensor nodes on the side of the road act as a gateway and connect to the Internet through mobile vehicles [26], so this is equivalent to an extension of SCS. Since mobile vehicles can communicate with the cloud using 5G, they can economically transfer

* Corresponding author.
E-mail address: weiliu@csu.edu.cn (W. Liu).

https://doi.org/10.1016/j.sysarc.2021.102219
Received 16 March 2021; Received in revised form 12 June 2021; Accepted 15 June 2021
Available online 7 July 2021

data to the cloud [27–29]. Bonala et al. [21] verified the feasibility of this method through real urban vehicle data. In fact, not only mobile vehicles, but also a huge number of mobile phones can also play the same data collector role. These mobile phones with rich sensing devices can sense a wide range of data and report to the cloud. This type of network is called Mobile Crowdsensing Network (MCN) [27,30]. And this method has huge advantages. These sensing devices eliminate the need for communication hardware with the Internet such as 5G, and more importantly, they can be deployed as needed, without adding infrastructure, and put into use quickly. Therefore, there are many such sensing devices. Class research [21,31].

In such a data collection method, in order to motivate data collector such as mobile vehicles, mobile phone users to actively report data, the usual method is to give a certain reward to the data of each report [27,30,31]. However, many data reporters falsify data or report virtual data in order to obtain rewards [19,26], and these data form the basis of applications. If these data are false, it will affect the quality of the constructed applications. In serious cases, these wrong data will cause decision errors, and bring huge losses to human life and property safety [25,31]. Therefore, how to ensure the credibility of the data collector report data is a challenge issue [19,26,31]. However, verifying the credibility of the data collector in SCS will face difficulties that were not available in previous systems. First of all, in the data acquisition of networks such as SCS, because the interactive behavior of the data collector cannot be recorded, the historical interactive behavior is unavailable, and even the authenticity of the submitted data cannot be verified. In this case, the credibility evaluation of the data collector has difficulties that are not available in any previous trust evaluation system [31]. Secondly, the traditional trust evaluation method is a passive observation method, which mainly makes trust evaluation by observing the behavior of the evaluated object. However, in data collection in SCS, most data reporters only have one data report behavior for a long time, so they cannot make behavior evaluation at all [31]. Finally, this traditional method of observing and evaluating behavior often takes a long time to obtain trust and evaluation, and it is difficult to obtain information and is difficult to implement, and the effect is not as expected. Although it is theoretically feasible, it is often not feasible in practice. A Lightweight Verifiable Trust based Data Collection (LVT-DC) approach is proposed to obtain credible data for mobile vehicles network to ensure that the cloud obtains credible data with low cost. Compared with the previous trustbased data collection methods, the LVT-DC approach has the following innovations. The LVT-DC approach is first an active trust evaluation method, which uses UAV to obtain the data collected by the evaluation object and mark it with a verification code to verify whether the MVs are credible; secondly, the LVT-DC approach is a kind of trust evaluation based on content comparison. The previous method is to make trust evaluation by observing the behavior of the evaluated object, and the external behavior often has the possibility of multiple internal reasons for the same behavior, so it is inherently uncertain of trust evaluation. The LVT-DC approach is an accurate method for determining trust based on the comparison of the data content of the report. It has a good advanced nature. In summary, the innovations of the work in this paper are as follows:

(1) We first proposed a lightweight verifiable trust acquisition framework. In our proposed trust acquisition framework has the following components: (a) IoT deivces. IoT deivces performs data sensing, and reports the sensed data to the cloud by MVs passing through it. It is considered credible by application deployment; (b) Mobile vehicles. It is used for data collection. The credibility of MVs is unknown, and it is possible to report false data in order to obtain rewards. (c) Cloud. Process the collected data to form an application. (d) Unmanned Aerial Vehicles. UAV is dispatched to pass a short verification code to each core-IoT device and collect the data of the core-IoT device. IoT deivces with verification code embed the verification code into the data when MVs collect their data, so that the cloud can inspect it. Under the framework proposed above, the trustworthiness of MVs can be evaluated.

(2) A verifiable trust inference method is proposed to quickly and accurately verify the credibility of MVs which includes 2 types of trust calculations. (a) Direct trust calculation: If the data of the core-IoT device of the MVs report is inconsistent with the data collected by the UAV, or the verification code cannot be recovered in the data reported, it means that the data of the MVs report is forged, thus reducing its trust. (b) Trust inference calculation: The trust value of some MVs is calculated through direct trust, and those highly credible MVs can be selected to perform trust evaluation on other MVs, thereby enriching the trust relationship. The reasoning relationship is: Trusted MVs and MVs with uncertain trustworthiness report the same IoT device data. Because the data reported by the trusted MVs is believed to be true. Therefore, the data submitted by MVs that need to be evaluated is compared with the data reported by credible MVs to infer the trust of other MVs.

(3) After a large number of experimental results, the proposed LVT-DC approach can quickly and accurately identify the credibility of the data collector and guarantee credible data collection. In comparison with our newly proposed BD-VTE scheme, the LVT-DC approach identified The time required for the same proportion of malicious MVs is reduced by 12.5%, the recognition accuracy is improved by 23.40% in the same time, and the accuracy of judgment trust is improved by 48.60%, and the recognition cost is reduced by 34.62%.

The rest of this paper is organized as follows: The related works is introduced in Section 2. In Section 3, the system model and problem statement are presented. The design of LVT-DC approach is proposed in Section 4. Then, performance analysis of LVT-DC approach presented in Section 5. Finally, Section 6 provides conclusions and future work.

2. Related work

Sensor–Cloud System (SCS) have received extensive attention from researchers. Wang et al. [6] gave the 3-layer architecture of SCS, namely (a) Wireless sensor networks layer; (b) Cloud layer; (c) User layer. Wireless sensor networks layer is located at the lowest layer of SCS. It is composed of numerous sensing devices to realize the perception of the physical world. And upload the sensed data to the cloud layer. Due to the storage, communication and computing capabilities of the sensing devices in the wireless sensor networks layer are weak, uploading its data to the cloud with strong computing capabilities can perform indepth data processes. At the same time, the cloud processes the data to form services, which can be used by multiple users [32], which shields the differences in the physical sensor node, so that users can effectively pay attention to their applications.

Bonala et al. [21] proposed an SCS based on mobile vehicle network. In such a network, a large number of smart sensing devices are deployed on the facilities that need to be monitored. These sensing devices have very simple hardware and sensing devices, which can sense specific physical phenomena, and can only carry out wireless communication in short distance with very low cost. For example, smart sensing devices are deployed on the street lights on both sides of the road to sense the status of the street lights. When the status of the street lights is perceived to be abnormal and needs to be repaired, it cannot directly communicate with the Internet [21]. Therefore, when MVs pass through their communication range, their data can be sent to MVs. Because MVs have good communication capabilities, they can transmit data to the cloud through the 5G network. Bonala et al. [21] confirmed that there are so many MVs in the city. Therefore, the trajectory of MVs can cover the entire city, and thus the data in the city can be effectively collected. Huang et al. [31] further found that, in fact, many sensing devices can also self-organize into a network, and those sensing devices on both sides of the road can act as gateways, other nodes send their data to the gateway through single-hop or multi-hop

routing. The gateway can upload the data to the cloud through MVs. This enables economic data collection [31].

In such a network, sensing devices are the perceivers of data, and MVs are the collectors of data [25,26,31]. Generally speaking, sensing devices are deployed for specific applications and are subject to controlling. Therefore, it can be considered that these sensing devices are credible [25]. In order to encourage MVs to collect data in the process of opportunity routing, the method is often used to reward the submitted data [31]. This incentive method for data collection has been widely adopted, especially in Mobile Crowdsensing Network (MCN) [27,30]. Examples of such applications include Vtrack [4] and NoiseTube [5]. NoiseTube needs to give a noise distribution map in the city. Therefore, when publishing tasks that need to sense noise, many mobile phone users perceive noise through their phones and report the data to the cloud. Cloud forms a noise distribution map based on the data collected comprehensively. However, in such a network, data collection MVs, or mobile phone users are third parties to the application, not controlled by the application, and their trustworthiness is unknown. Therefore, there are some untrustworthy MVs report false data to obtain rewards. Some malicious MVs even submit malicious data to achieve the purpose of attacking applications. Therefore, the most critical issue in such a system is how to ensure the authenticity of the data submitted by the data collector. Using a trust evaluation method to ensure the authenticity of the data submitted by the data collector is an effective method. In this method, the system conducts a trust evaluation on the data collector such as MVs in advance, and selects those MVs with a high degree of trust to report the data. Since credible MVs will honestly receive data from sensing devices and report to the cloud, choosing credible MVs for data collection can effectively ensure that the data is authentic and credible. In this way, the key to the problem is how to effectively identify and evaluate the trustworthiness of MVs [25,31].

In fact, the evaluation of credibility has always been a key issue in the field of network research [33–35]. Researchers have proposed quite a few studies on evaluation of credibility [36–39]. At the very beginning, the researchers proposed a single-rating trust evaluation system [35]. The specific method is that when the evaluated object completes an interaction, the evaluated object conducts a trust evaluation of the evaluated object, which generally integrates all the evaluations of the evaluated object over a period of time into a comprehensive evaluation. Obviously, the closer the evaluation is to the current time, the more reference value it has for the evaluated object, so its weight is greater in the comprehensive evaluation. The longer the evaluation is from the current time, the lower its reference value, and the lower its weight [36]. A comprehensive evaluation is formed by weighting all evaluations over a period of time. This type of evaluation method is widely used in real life. For example, most service windows, such as banks, and government service windows are mostly equipped with evaluators, and the clerks can evaluate the service quality of the staff. However, this single-rating evaluation system is vulnerable to malicious evaluation by malicious customers. Therefore, some researchers have proposed a dual-rating evaluation system. In such a system, both interacting parties evaluate the current interaction behavior, and if the evaluations of both interacting parties are consistent, the system adopts this evaluation. If the evaluations of the two interacting parties are inconsistent, the evaluation of a certain aspect is considered to be unreliable. At this time, the evaluation conclusion is not adopted, or both parties of the interaction are punished (that is, the trust is reduced). More trust evaluation methods are to observe the interaction behavior of the evaluated object, if the interaction behavior meets expectations, increase its trust degree, otherwise reduce its trust degree [38,40,41]. A set of trust reasoning and calculation methods are also formed in the trust evaluation research. In this method, trust is mainly divided into two categories: One is direct trust. Direct trust is generally the trust evaluation of the other party after the object directly interacts with the evaluated object [34]. Obviously, direct trust It is relatively accurate, and the reliability is relatively high. The other type is indirect trust [35]. In practice, most objects have no direct interaction, so only a direct trust relationship will cause many objects to fail to establish trust evaluation.[42] Therefore, in this case, indirect trust It will work. Multiple objects can form a trust chain through direct trust between each other, and indirect trust can be calculated between objects that do not directly interact. The general method is to multiply the trust on the trust chain to obtain indirect trust degree. It can be seen that the degree of indirect trust will decrease as the trust chain grows, that is, after too much trust inference, the trust relationship becomes weaker. Indirect trust can also have multiple trust chains, and finally the system will integrate direct trust and indirect trust to form a unified trust evaluation [31,43].

In SCS, for the trust of sensor nodes, Wang et al. [27] proposed a way to monitor the interactive behavior of these nodes by sending mobile edge users to the area of sensor nodes, and to conduct trust evaluation based on the interactive behavior of these sensor nodes. In this way, the trust of these nodes can be obtained. Obviously, this method can more accurately evaluate the trust of sensor nodes, and has good application prospects [27]. However, using this kind of method of observing the interactive behavior of sensor nodes to gain trust will still be insufficient because the accuracy of trust evaluation is not very high. Because some malicious nodes can deceive observers through false behaviors [44]. For example, in the select forwarding attack in the sensor network, sensor nodes intentionally drop the data packets forwarded by them to cause damage to the network [24]. And if it is unable to determine whether it has dropped a data packet through its external data forwarding action, because it can indicate that it is sending a data packet through a false communication signal, but it is not actually sending a data packet. In addition, even if it really sends a data packet, it sends a false data packet also has a destructive effect, and this cannot be identified based on external behavior observations. In addition, even if sensor nodes do not intentionally initiate some false actions, but in fact the same action has different reasons, it is different to determine whether the action is credible. This will cause inaccuracy in trust evaluation.

Therefore, the best method is to compare the data submitted by MVs with the data delivered to it by IoT devices. If the data reported by MVs is consistent with the data delivered to it by IoT devices, then the MVs are credible. On the contrary, it can fully explain that MVs submitted false data, which can reduce their trust. Obviously, this method is a deterministic and accurate trust evaluation method. We have already proposed this trust evaluation method in Ref. [31]. In our proposed method, we send UAVs to collect data from some IoT devices. Since the UAV is sent by the system, it is credible, and the data collected by it is also credible. Then, the data packets collected by these UAVs are compared with the data of the MVs report, if the two data are consistent, the trust of MVs can be improved [31]. On the contrary, it means that MVs report false data, thereby reducing the trust of MVs. Obviously, this method is an effective method.

However, the shortcomings of this method we proposed earlier are: This method requires sending UAVs to collect data from IoT devices for verification. Because the data collected by UAVs requires cost, therefore, the data is always limited, which leads to a small number of verified MVs that can be obtained, and the coverage of trust evaluation is not large. Moreover, since the trust level of MVs is also dynamically changing, it is necessary to maintain the amount of data collected by the UAV and the collection frequency above a certain level to achieve good results. In this way, the system will pay a higher price. Therefore, in this article, we have improved the method proposed in the previous stage, so that the trust evaluation obtained by the system at a much less cost is faster and more accurate.

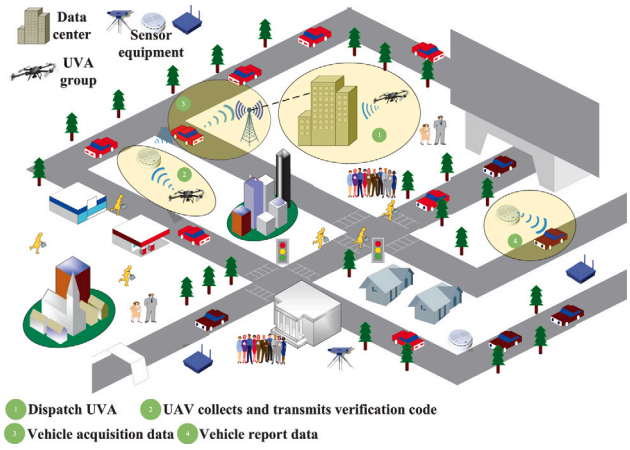


Fig. 1. Network model.

3. System model and problem statement

3.1. Edge computing network model

Fig. 1 shows the network model of this article, which consists of three parts, followed by Internet of things device, data center, and data collector.

IoT device, here mainly refers to the sensor device, which is used to sense the surrounding environment and store data in the memory. Due to the limitation of its energy, it is necessary to send the data forwarded to the mobile node within communication range. We use $S = \{s_1, s_2, \ldots, s_n\}$ to represent these devices that need to send data to the mobile node, where s_i represents the IoT device numbered i, and n represents the total number of IoT devices in the network.

Data center, the IoT device data reported by MVs (Mobile Vehicles) through the 5G network or the data collected by UAVs (Unmanned Aerial Vehicles) will be aggregated in the data center. The data center conducts comparison, evaluation and other tasks, it can also send UAVs to actively collect IoT device data for verification. Data collector, it is an important means for data center to collect data in IoT device, the data collector takes the data stored in the IoT device and reports it to the data center. In most researches, MVs, pedestrians and so on generally can be used as data collectors. The data collectors in this article are mainly MVs and UAVs (see Table 1).

3.2. Problem statement

In a series of research centers proved the feasibility of MVs as a data acquisition tool. The wide range of MVs activities and the large amount of collected data are its advantages as a data collection tool. The disadvantage is that if the system is completely dependent on MVs, then the reliability of the data also depends entirely on the behavior of the MVs themselves. Malicious MVs report false data to defraud rewards through fraud, which will make it difficult for us to collect correct data.

This means that the key to the problem is to know that we can verify the correctness of the received data. We summarize the problems as follows: 1. Generally speaking, normal MVs with high trust will report reliable data, while malicious MVs with low trust will often report false data. In order to obtain the data center to data quality, system want to choose a high degree of trust in the data collected by the MVs. In other words, distinguishing between normal MVs and malicious MVs is the key to the problem. 2. The data center needs to compare the data reported by the MVs in order to conduct a trust assessment of the vehicle. Some MVs have fewer interactions. In the case of a low accuracy of judgment trust, their trust cannot be evaluated, and even malicious MVs are not verified when they conduct malicious interactions. When they engage in joint fraud, they are mistakenly evaluated as normal vehicle. Therefore, increasing the accuracy of judgment trust is also conducive to improving the safety and stability of the system. 3. UAV, as a tool for proactively verifying data, are due to their high usage costs. Therefore, it is not only necessary to optimize the path of UAVs to save time, but also to obtain as much data of IoT devices as possible every time it flies, so as to verify more MVs.

Table 1
Notation table.

Notation	Meaning
n	Number of IoT devices
s_i	IoT device i
S	Set of IoT devices
S^V	Set of verifiable IoT devices
$\mathcal{A}$	Recognition accuracy
$\mathcal{J}$	Accuracy of judgment trust
C	Recognition cost
$\bar{T}_{\text{nor}}$ /$\bar{T}_{\text{mal}}$	Average trust value of normal/malicious MVs
$\mathfrak{A}_i/\mathfrak{B}_i$	Number of verifiable/unverifiable data reported by MV i
$\mathfrak{C}_i$	Cost of using UAV i
R_i^j	Flag of MV i reporting data from IoT device j
u_i/f_i	Successful/Failed verification results of MV i
λ	Trust penalty factor
T_{act}^i	Active trust value of MV i
$e_{i,j}$	Interaction situation between MV i and j
$p_{i,j}/q_{i,j}$	Successful/Failed interactions between MV i and j
$\mathscr{E}_{i,j}$	Flag of interaction between MV i and j
T_{rec}^i	Recommendation trust value of MV i
T_{com}^i	Comprehensive trust value of MV i
V	Trust weights of different periods
W	Time weighting factor
t	Periodic time
S_{path}	Set of UAV target acquisition IoT devices
SP_i	The number of the IoT device i in the S_{path}
$info_i$	Number of times the IoT device i was reported
τ	UAV's cost factor
d_i	Total flight distance of UAV i
v_i	Flight speed of UAV i
ω	Path time penalty factor
φ	Broken line penalty factor
L	Flight path length of UAV
Pb	Number of IoT devices for original path
Ps	Number of IoT devices for improved path
$\alpha_1, \alpha_2, \alpha_3$	Weight of reward and punishment function

Definition 1 (*Recognition Accuracy*)**.** Recognition accuracy $\mathcal{A}$, which is strictly defined as shown in Eq. (1). We use $\bar{T}_{\text{nor}}$ and $\bar{T}_{\text{mal}}$ to denote the average trust values of normal MVs and malicious MVs, respectively, $\mathcal{A}$ represents the difference between their average trust values. The higher the recognition accuracy between the normal MVs and the malicious MVs in the system, the higher the recognition accuracy, and the more accurately the reliable MVs can be distinguished. Therefore, the target of the system is to achieve *Max*($\mathcal{A}$).

$$\mathcal{A} = \bar{T}_{\text{nor}} - \bar{T}_{\text{mal}} \tag{1}$$

Definition 2 (*Accuracy of Judgment Trust*)**.** Accuracy of judgment trust $\mathcal{J}$, which is strictly defined as shown in Eq. (2). The accuracy of judgment trust refers to the ratio of the number of data that can be verified by the MVs report within the current cycle and the ratio of the total number of MVs reported data. $\mathfrak{A}_i$ represents that the number of data that can be verified by the data center of MV_i report, $\mathfrak{B}_i$ represents the number that cannot be verified. The larger the value of $\mathcal{J}$, it means that the system can judge more data currently received, thus making the judgment of trust more accurate. Therefore, the target of the system is to achieve *Max*($\mathcal{J}$).

$$\mathcal{J} = \sum_{i=1} \frac{\mathfrak{A}_i}{\mathfrak{A}_i + \mathfrak{B}_i} \tag{2}$$

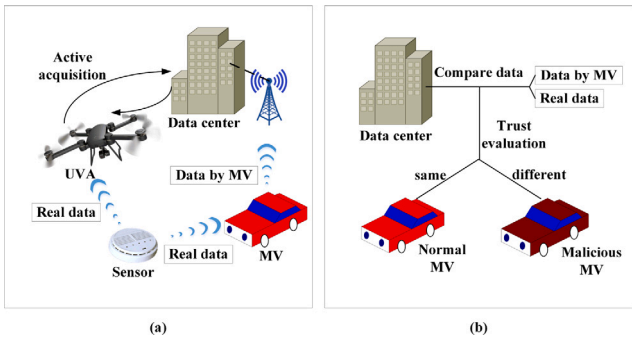


Fig. 2. BD-VTE scheme.

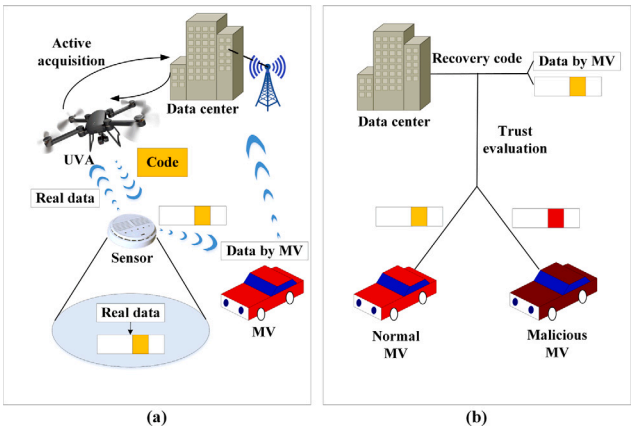


Fig. 3. LVT-DC approach.

Definition 3 (*Recognition Cost*)**.** Recognition cost C, which is strictly defined as shown in Eq. (3). UAVs are used as a tool for active verification in approach, and the cost of system recognition can be represented by the cost of UAVs. Therefore, the cost of recognition is defined as the ratio of the cost of UAVs to the accuracy of judgment trust. Its meaning is the cost of UAVs needed to realize the unit's judgment of trust accuracy. The smaller its value is, it means that the trust judgment on MVs has been realized with a smaller cost in the current cycle. $\mathfrak{C}_i$ represents the price of UAV_i and $\mathcal{J}$ represents the accuracy of judgment trust defined in Eq. (2). In order to reduce the cost of system recognition, therefore, the target system is $Min(C)$.

$$C = \frac{\mathcal{J}}{\sum_{i=1} \mathfrak{C}_i} \tag{3}$$

In summary, the objectives are as follows the Eq. (4):

$$\begin{cases} Max(\mathcal{A}) = Max\left(\bar{T}_{\text{nor}} - \bar{T}_{\text{mal}}\right) \\ Max(\mathcal{J}) = Max\left(\sum_{i=1} \frac{\mathfrak{A}_i}{\mathfrak{A}_i + \mathfrak{B}_i}\right) \\ Min(C) = Min\left(\frac{\mathcal{J}}{\sum_{i=1} \mathfrak{C}_i}\right) \end{cases} \tag{4}$$

4. The design of LVT-DCA approach

4.1. Research motivation

Our research motivation is as follows:

In order to realize the efficient communication of edge node networks, the method widely used in a series of research is to establish a trust mechanism to solve this problem. Its purpose is to increase the trust of normal nodes, increasing effective and reliable interactions. When the malicious node has fraudulent behavior, reducing trust and interaction, increasing system security, reducing unreliable interaction.

The scheme of trust mechanism is mainly divided into two types, respectively, which is active trust and indirect trust. S. Huang et al. proposed the use of UAVs as the main collection tool in the approach in [31], and the data in the IoT device collected by the UAVs is used as baseline data for verification. This mainly solves the problem of the lack of authoritative verification results to maintain the trust relationship when establishing the trust mechanism in the past research. For example, a malicious MVs launches a joint fraud attack on the IoT device node for a specific purpose, causing the system to be unable to distinguish between real data and false data.

We show the approach of UAVs as an active verification tool in Fig. 2, hereinafter referred to as "BD-VTE" scheme. Fig. 2(a) shows the process of UAVs acquisition. Their approach is that in the current cycle, the IoT device transmits real data to the MVs, and the MVs reports the data to the data center via the network. The data center receives the data reported by the MVs, and before proceeding with the evaluation work, it sends UAVs to select a certain number of IoT devices with the most interactions for collection work. The purpose of this is to make the UAVs within the limited flight capability, to verify as many MVs as possible. The data collected by UAVs is the real data of IoT device, which is regarded as credible baseline data. Fig. 2(b) shows the verification process for the data center to distinguish between normal MVs and malicious MVs. T Fig. 2(b) shows the verification process for the data center to distinguish between normal MVs and malicious MVs. The data center conducts trust evaluation on MVs by comparing the baseline data collected by UAVs and the data reported by each MVs. The advantage of this is that there are authoritative and credible verification results, the trust relationship of the system is more secure and reliable, and the problem of joint fraud by malicious MVs is effectively suppressed. However, this approach also has the following shortcomings:

(1) The recognition accuracy is low.

The recognition accuracy represents the system's ability to distinguish MVs. When a certain MVs interacts in only one of the cycles in a continuous cycle, if the data reported by the MVs cannot be verified by the data center in the current cycle, the data center cannot perform it trust evaluation. This will reduce the update speed of the average trust value of MVs, thus reducing the degree of discrimination between normal MVs and malicious MVs.

(2) The accuracy of judgment trust is low.

The accuracy of the judgment trust represents the amount that the system can judge from the data currently received. The higher the value, the higher the authority of the system to judge the trust. The active acquisition capability of UAVs is limited. In a larger edge node network, a large amount of interactive data will be generated in each cycle. In the case of a low proportion of verifiable data, it will cause a waste of interactive data and affect the accuracy of judging trust

(3) Recognition cost is high.

The low cost of recognition means that the cost of UAVs required to achieve the accuracy of the current cycle unit judgment trust is lower, and it represents the ability of the system to utilize the cost. For example, when an IoT device has been reported by MVs at a high level in multiple cycles, the UAVs needs to collect the IoT device repeatedly, and cannot collect other IoT devices. Therefore, other IoT device data reported from MVs cannot be verified, which causes a waste of UAVs' cost. In our model.

In order to solve the shortcomings listed in the above approach, we proposed the "LVT-DC" approach.

4.1.1. Analysis of LVT-DC approach

Our approach process is shown in Fig. 3. The innovation of our "LVT-DC" approach lies in the use of UAV as a collection tool like the approach shown in Fig. 2. While collecting, the UAV transmits the verification code to the IoT device. The specific method is that in each cycle, we also use UAVs to select a certain number of IoT devices that have been reported the most by MVs for interaction. UAV flies to the selected IoT device to collect data. While collecting data, it uses the interaction characteristics of UAV and IoT device to transmit a time-sensitive verification code to the IoT device. After the IoT device gets the verification code, it will be added to the data message to be sent to the MVs. When the MVs are close to the communication range of the IoT device to report data, the IoT device will send a data message containing the verification code to the MVs, and the MVs will report it to the data center. When the data center obtains the data reported by MVs, it will restore the verification code in this piece of data according to the agreed rules. If it can be successfully restored, it means that the MVs report is the real data. If the verification code cannot be successfully recovered from this piece of data. It means that what MVs report is false data, through this method to establish a trust relationship.

The advantage of the "LVT-DC" approach is that it can not only verify the IoT device that has been reported the most by MVs in the current cycle, but also the IoT device that has transmitted the verification code during UAVs collection in the previous cycle. When UAVs collect data from IoT device without verification code for the first time, the verification code is transmitted to it, so that within the time that the verification code is valid, the data reported by MVs will contain the verification code. UAV no longer needs to fly to this IoT device to collect data again in the period when the verification code is valid later Until the verification code expires, if the device still needs to be collected in the current cycle, when the UAV needs to collect the invalid IoT device again, retransmit the verification code to it.

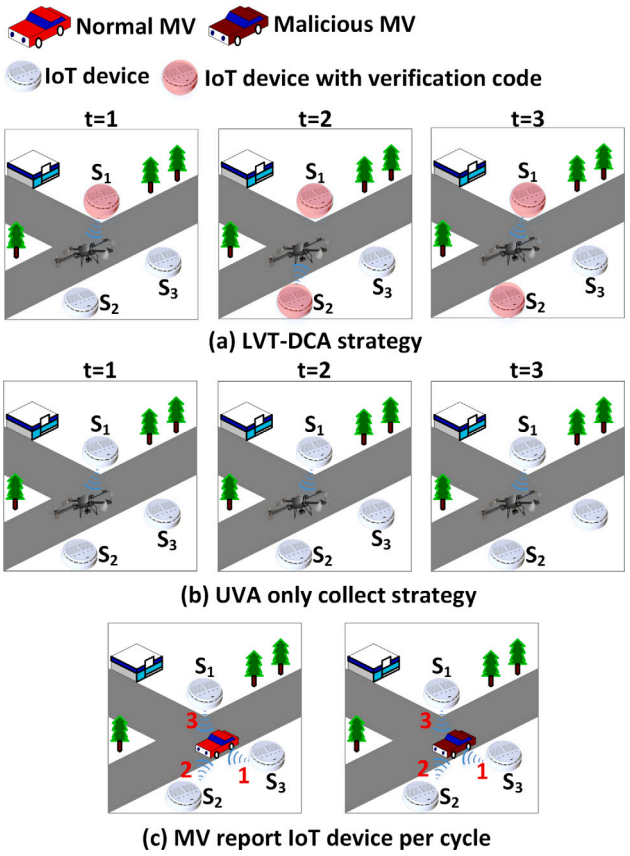


Fig. 4. Case of comparison of two approaches. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

4.1.2. Case of approach comparison

Next, we will briefly compare the differences between the two approaches through the simple case shown in Fig. 4. We suppose deployed 3 IoT devices, denoted by S_1, S_2, S_3, respectively, and t denotes the cycle time. A normal MV that continuously reports real data and a malicious MV that continuously report false data are set up in this case. We set the validity period of the verification code to 2 cycles, and the UAV can only interact with 1 IoT device per cycle.

Figs. 4(a) and 4(b) show the situation of UAV collecting data under two approaches. The normal MV and malicious MV report the same to each IoT device in each cycle, Fig. 4(c) shows the number of times that two MVs report to each IoT device in each cycle. They report 3, 2, and 1 times for S_1, S_2, S_3 respectively in each cycle. That is to say, in 4 cycles, each MV was reported 24 times, so the total number of two MVs reported by the data center is 48. In order to make the comparison result more direct and simple, the initial trust value of MV is 0.5. Every time MV reports real data, the trust is increased by 0.01. On the contrary, if false data is reported, the trust in MV is reduced by 0.01. We denote the trustworthiness of normal MV as T^n, and the trustworthiness of malicious MV as T^m.

Fig. 4(a) shows our "LVT-DC" approach, The IoT device that is reported the most times in each cycle is S_1, so when the initial $t = 1$, we will transmit the verification code to it while collecting the data of S_1. When $t = 2$, the data from S_1 can be verified by means of a verification code, so we collect and transmit S_2 in the same way. When $t = 3$, the verification code in S_1 becomes invalid because the verification code is valid for 2 cycles. Because S_1 is still the IoT device with the most reports in the current cycle, we will start collecting and transmitting it again. Therefore, the total number of data that can be verified in 4 cycles of our approach is 26. The number of real data reported by normal MV is 13($S_1 : 9, S_2 : 4, S_3 : 0$), and its trust value $T^n = 0.63$. Similarly, the number of false data reported by malicious MV is also 13, and the trust value $T^m = 0.37$.

The "BD-VTE" scheme is shown in Fig. 4(b). Because the IoT device that is reported the most times in each cycle is S_1, only S_1 data can be collected in each cycle. This means that the number of data that can be verified in 4 cycles is 18. The number of true data reported by a normal MV is 9($S_1 : 9, S_2 : 0, S_3 : 0$), and its trust value $T^n = 0.59$. Similarly, the number of false data reported by malicious MVs is 9, and the trust value $T^m = 0.41$.

The red nodes in Fig. 4 are all IoT devices that can be verified in the current cycle. It can be seen that our approach is only in the initial cycle (all IoT devices do not contain verification codes) and the verification capability of the "BD-VTE" scheme is the same, The latter can only maintain the same verification capability in consecutive cycles. As the cycle goes on, our approach verification capabilities will gradually improve. We use two specific values to illustrate the difference between the two approaches, namely the average accuracy of judgment trust $\bar{J}$ and the recognition accuracy $\mathcal{A}$. Among them, $\mathcal{A}$ is calculated by Eq. (1), using the final result of all cycles, and $\bar{J}$ is the result calculated by Eq. (2), which represents the average value of all cycles. In the previous simple case, the average accuracy of judgment trust of the "BD-VTE" approach is $\bar{J} = 37.5\%$, and the recognition accuracy $\mathcal{A} = 0.18$. Similarly, in our "LVT-DC" approach, the average accuracy of judgment $\bar{J} = 54.17\%$, and the recognition accuracy $\mathcal{A} = 0.26$.

Through this case, it is proved that the "LVT-DC" approach has increased average accuracy of judgment trust $\bar{J}$ and the recognition accuracy $\mathcal{A}$ by 44.45% and 44.44% respectively, and in the two approaches, under the same energy consumption of the UAVs, what we can verify the number of IoT devices is significantly higher and will remain at a level after stabilizing.

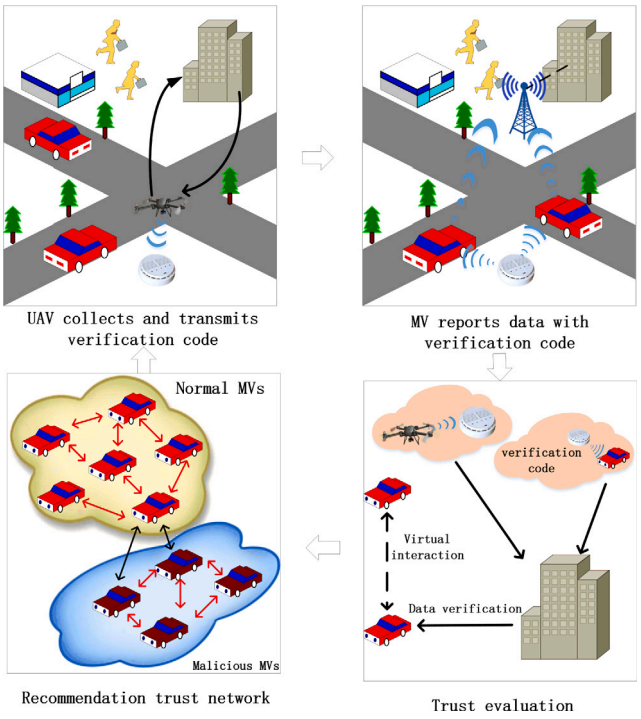


Fig. 5. Trust mechanism framework.

4.2. Data collection and trust evaluation

Fig. 5 shows the trust mechanism we have established. Just as the "LVT-DC" approach described in the motivation, in each cycle, the IoT device data is collected through UAVs for comparison, and the verification code is added to the collected IoT device. After the verification code is valid for a continuous period, the MVs will report the data of the IoT device within the communication range. We evaluate the trust of the MVs by restoring the verification code in the data reported by the MVs.

We establish trust relationships by comparing the data collected by UAVs with the data reported by MVs, or by restoring the verification code contained in the MVs report data. Therefore, active trust is also the most reliable trust. There will be several vehicles reporting the same node between MVs. We believe that in this case, there are virtual interactions between MVs, and they can form a recommendation network in the graph to establish an indirect trust relationship.

4.3. Trust evaluation mechanism

This part of the work mainly introduces the detailed algorithm for establishing trust evaluation for MVs. In the previous section, we have shown the trust framework. The trust methods are active trust and recommended trust, in order to prevent malicious vehicles from fraud in a certain period. Behavior, the trust value of the vehicle is synthesized according to the different proportions of each cycle. Finally, we will use these two types of trusts to obtain the comprehensive trust of each vehicle by weighted summation.

4.3.1. Active trust

The establishment of active trust comes from our direct verification of the data reported by MVs, which is mainly achieved through UAVs. Within the communication range, UAVs obtain their data by interacting with IoT devices, and use the collected data as our credible baseline data.

As we described in the motivation, for the data reported in the current cycle of MV, if it reports an IoT device that does not contain a verification code, and this IoT device is within the flight target of the UAVs in the current cycle, by comparing with the data collected by UAVs, assess whether the MV has reported real data. If it reports an IoT device that cannot be collected currently, but the reported data contains a verification code and is within timeliness, the MV will be evaluated by restoring the verification code in the MV report data. If the data reported by the MV is verified as real data, then its trust value should be increased, otherwise, if the data reported by the MV is verified as false data, its trust value should be lowered.

We use Beta distributed applications to establish active trust evaluation. An important application of the Beta distribution is to estimate the probability of an experiment's success. Its probability density function is shown in Eq. (5). It contains two parameters. Generally, α, β represent the number of successes and failures, which is also called shape parameter. We refer to the explanation in [45], assuming that an experiment satisfies the binomial distribution, and the probability of success is set to θ. Since θ can only be taken from a value in the interval [0,1], the prior distribution $\theta \sim U(0,1)$ is reasonable. Through n times of experimental observation, set the number of successes to k, and the number of failures to be $n-k$. This means that we need to calculate the conditional probability of θ. The condition is the number of observed successes and failures. The result of its calculation is the result of the Beta distribution, which is the probability of predicting the success of the experimental result.

$$f(\theta;\alpha,\beta) = \frac{1}{B(\alpha,\beta)}\theta^{\alpha-1}(1-\theta)^{\beta-1} \tag{5}$$

In our model, we judge whether the data reported by MVs is real data by recovering the verification code or comparing with the results collected by UAVs. When an MV reports real data, it is deemed a success, otherwise, if it reports false data, it is deemed a failure. Therefore, we use the predicted MV success rate as its trust value, that is, the more successes reported in the past MV, the higher the predicted success probability will be, and the trust value will increase accordingly.

We show this application of Beta distribution through a simple case. Assuming that the probability of MV reporting real data is θ, and this probability obeys a certain distribution, a total of n reports have been made, in which the number of times of reporting real data is set to x, and the sample result is set to y. Each MV report is independent of each other. Therefore, the entire process of reporting data is subject to a binomial distribution. According to the knowledge in statistics, we should give the data a prior expectation. The success rate is taken from a value in the interval [0,1] under the same possible circumstances, so the prior distribution is $\theta \sim Beta(1,1)$ (that is, uniform distribution). Our purpose is to predict the distribution of the probability of success θ and estimate its value. This is changed to the problem of finding the posterior probability in Bayesian inference, which is represented in Eq. (6b). $p(y)$ represents the result of the data sample, and the result has nothing to do with θ. By calculating $h(y)$ represented by Eq. (6b) and $g(\theta, y)$ represented by Eq. (6b),it can be found that the result of $h(y)\cdot g(\theta, y)$ is the same as the molecular result of the in Eq. (6b), which proves that the posterior distribution satisfies $Beta(\alpha+k, \beta+(n-k))$The distribution of this parameter. This shows that we can predict the success rate of MV report data through the expected result of the Beta distribution. Suppose that when a MV reports 3 times of data, of which 2 times are real data and 1 time is false data, the predicted success rate should be the expectation of $Beta(3,2)$.

$$p(\theta|y,\alpha,\beta) = \frac{p(y \mid \theta)p(\theta \mid \alpha,\beta)}{p(y)} \tag{6}$$

$$p(y|\theta)p(\theta|\alpha,\beta) = \binom{n}{k}\theta^{k}(1-\theta)^{n-k}\frac{1}{B(\alpha,\beta)}\theta^{\alpha-1}(1-\theta)^{\beta-1} \tag{6b}$$

$$h(y) = \binom{n}{k}\frac{B(\alpha+k,\beta+(n-k))}{B(\alpha,\beta)} \tag{6b}$$

$$g(\theta,y) = \frac{1}{B(\alpha+k,\beta+(n-k))}\theta^{\alpha+k-1}(1-\theta)^{\beta+n-k-1} \tag{6c}$$

However, in reality, trust is a slow accumulation process. We will not greatly increase the trust of MVs just because they report real data once. However, for the malicious behavior of MVs reporting false data, penalties should be appropriately increased. Therefore, we set a penalty factor $\lambda(\lambda > 1)$ for the expectations of the Beta distribution. When the MV conducts malicious behaviors of reporting false data, its trust will drop faster. As shown in Eq. (7). Where u_i is the number of times that the vehicle MV_i reports real data, and f_i is the number of times that the vehicle MV_i reports false data.

$$T_{act}^{i} = \frac{u_i + 1}{u_i + \lambda f_i + 2} \tag{7}$$

4.3.2. Recommendation trust

Recommended trust, as a common way of trust, can help us distinguish normal MVs from malicious MVs. The data center will receive data from different IoT devices reported by MVs. For MVs that report the same IoT device, we assume that they have established a virtual interaction.

In our model, for MVs that report data from IoT devices that cannot be verified, we determine the recommendation trust by comparing the similarities and differences of the data reported between MVs. By comparing with MVs with high trust, if they are the same, it is considered that a successful virtual interaction is established, and if they are different, it is considered that a failed virtual interaction is established. For example, the IoT device numbered 1 cannot be verified. Both a normal MV and a malicious MV report the data of this device in the current cycle. After the data center receives it and passes the comparison, it is considered that they have established a failed virtual interaction.

For each IoT device jointly reported between vehicles, we use the same formula as Eq. (8) to express their interaction success rate. The difference in (8) is that $p_{i,j}$ represents the successful establishment of the vehicle The number of virtual interactions, $q_{i,j}$ represents the number of failed virtual interactions established by the vehicle, and a recommendation network between vehicles is established through the interaction.

$$e_{i,j} = \frac{p_{i,j} + 1}{p_{i,j} + q_{i,j} + 2} \tag{8}$$

In the recommendation network of MVs, for MVs that report the same IoT device, the recommended trust value of MV_i is generated by the joint recommendation of MVs with a higher comprehensive trust level. This also means that there will be an authoritative MVs with the highest trust in the recommendation network. Malicious MVs will often send incorrect data to report, which is different from normal MVs. Therefore, their interaction success rate will be at a relatively low level, and the recommendation trust they will receive will be lower. On the contrary, the recommendation trust of normal MVs will increase. This helps us distinguish between normal MVs and malicious MVs.

$$T_{rec}^{i} = \frac{\sum_{j=1} \ell_{i,j} \left(T_{com}^{i}\right)^2 e_{i,j}}{\sum_{j=1} \ell_{i,j} T_{com}^{i}} \tag{9}$$

Eq. (9) represents the way we calculate the recommended trust of MV_i. We use the comprehensive trust value T_{com}^{i} of each MVs to represent the recommendation ability of each MVs. The purpose is to ensure that high trust MVs have a higher recommendation ability, on the other hand, to prevent the fraudulent behavior of malicious MVs, to give a lower recommendation value to disrupt the system balance. $\ell_{i,j}$ indicates whether a virtual interaction is established between MV_i and MV_j, that is, data of the same IoT device has been reported. That is, $\ell_{i,j} = \{0|1\}$.

Algorithm 1: Comprehensive trust computing strategy.

Input: $T_{\text{com}}^{t-1}, S^V, S, R$
Output: T_{com}^{t}

```
for each MV_i ∈ MVs : do
    for each s_j ∈ S do
        if s_j ∈ S^V and R_i^j = 1 then
            Calculate T_act^i according to Eq. (7)
        end if
    end for
end for
for each MV_i ∈ MVs : do
    for each s_k ∈ S do
        if R_i^k = 1 then
            for each MV_j ∈ MVs do
                if each R_j^k = 1 and T_com^j > T_com^i then
                    Calculate e_{i,j} according to Eq. (8)
                end if
            end for
        end if
    end for
    Calculate T_rec^i according to Eq. (9)
end for
Calculate T̂_act according to Eq. (10)
Calculate T̂_rec according to Eq. (10)
Calculate T̂_com according to Eq. (11)
```

4.3.3. Comprehensive trust computing strategy

Our approach is to calculate the comprehensive trust of MVs through the above two types of trust obtained. In the initial stage of trust establishment, whether it is the data collected through UAVs or the data verified by restoring the verification code, it is still in a small scale state. By calculating the active trust Eq. (7), it can be seen that if malicious MVs obtain a higher trust value through fraudulent behavior in the initial stage and report several correct data consecutively, it will damage the stability of our system. In order to maintain the stability of the system and ensure the fairness and reasonableness of the trust mechanism, we adopt the method of proportionally averaging the trust of MVs in the last $\mathcal{N}$ cycles. This approach avoids the impact of MVs gaining excessive trust in a single cycle.

We use $V = \{V_1, \ldots, V_{\mathcal{N}}\}$ to represent the different proportions of each cycle, V_1 represents the most recent cycle, and $V_{\mathcal{N}}$ represents the farthest cycle from the current. The closer the information to the current period is, the more recent information should be given. now represents the current cycle. Eq. (10) represents our method of obtaining the average trust value.

$$\hat{T}_l = \sum_{t=1}^{\mathcal{N}} V_t T_i^{now-t+1} \tag{10}$$

After the two types of trust are obtained by calculation, we calculate the comprehensive trust by weighted summation. The calculation method is expressed by Eq. (11). W represents the weight of trust. Active trust is obtained by direct inspection, so it is the most reliable way of trust and should be given a major weight.

$$T_{com}^{i} = W\hat{T}_{act}^{i} + (1 - W)\hat{T}_{rec}^{i} \tag{11}$$

Algorithm 1 contains the process of calculating active trust. By comparing with the data collected by UAVs or recovering the verification code, and comparing with the reported data, we can judge whether it has reported real data. $R_j^i = 1$ means that MV_i reports the data of IoT device node s_j. If $R_j^i = 0$, it means that MV_i has not reported the data of IoT device node s_j. S is the set of IoT device nodes indicated in 3.1, and S^V represents the set of IoT device that we can check. It also includes the calculation process of recommendation

trust. Each MV that needs to be evaluated has reported the same IoT device with other MVs, and its recommendation trust is derived from the recommendations of these MVs. Finally, the calculation process of comprehensive trust, t represents the current cycle. After obtaining active trust and recommendation trust respectively, they are calculated by the method in Eq. (10). Obtain the comprehensive trust of the current cycle through Eq. (11).

4.4. UAV's flight target and path

After the data center receives the data reported by the MVs, it will dispatch UAVs to the selected main IoT device set to interact with each other to verify the data reported by the MVs. In our "LVT-DC" approach, UAVs also transmit the verification code to the corresponding IoT device node while collecting. In this section, the work we mainly introduce is (1) Sort the IoT device routing times in descending order, and select the core-IoT device collection. This is mainly to be able to verify more data, thereby assessing more MVs. (2) After obtaining the IoT device set, we convert the flight problem into a TSP problem, and use the simulated annealing algorithm to make the UAV's flight path as close to the optimal solution as possible, and find the shortest flight path of the UAV. (3) the UAV flight path, there will be some IoT device node, which we call non-core IoT device, which is a short distance communication range of the UAV, by interacting with them, you may be increased with our accuracy of judgment trust and verification speed. In order to further improve the system performance, we adjust the path so that the UAV can collect more IoT devices while adding very little price.

4.4.1. Select the core-IoT device

The IoT devices deployed on the map are in different locations [46], so the number of times they are reported by the MVs is also different. Generally, in areas with dense MVs trajectories, IoT devices are reported more often. UAVs can evaluate a larger number of MVs by interacting with these IoT devices. We call this type of IoT device a core-IoT device.

When dispatching UAVs for collection and transmission, we use core-IoT device as the main target. We use Eq. (12) to sort the IoT device target set to be collected. $info_i$ represents the number of IoT device s_j reported by MV_i, $R_j^i = \{0|1\}$ means MV_i reported IoT device s_j. Then b_i represents the number of times s_i was reported by MVs. By sorting b_i from largest to smallest, we set a fixed number num as the size of the IoT device collection.

$S_{path} = \{SP_1, \ldots, SP_k, SP_{num}\}$ is used to represent the set of paths we can get, and SP_k represents the IoT device number after sorting according to the rules.

$$b_i = \sum_{j=1} R_j^i info_i \tag{12}$$

4.4.2. Path planning based on simulated annealing

Relative to the UAV to perform the collection work, or the work of transmitting the verification code to the IoT device. UAV in flight time spent is much larger than this part of the work. This shows that the energy consumption of the UAV mainly depends on the time it takes to fly. Therefore, in our model, we use Eq. (13) to represent the price of the UAV to perform work. It is assumed that the UAV flies at a constant speed v and follows the planned path. d represent the distance traveled by the UAV. tau is the price coefficient of the UAVs' price.

$$\mathfrak{C} = \tau \frac{d}{v} \tag{13}$$

The energy consumption of a UAV is proportional to the distance it travels. Therefore, how to plan a path to shorten the flight distance of the UAV is the focus of our approach. The UAV is dispatched by an advanced data center to collect information from core-IoT devices and transmit a verification code before returning. The idea is similar to the classic traveling salesman problem in dynamic programming. In our approach, we solve this problem by using the idea based on the simulated annealing algorithm. The key improvement is that compared with the general greedy strategy, simulated annealing accepts solutions that fail to reach the local optimum with a certain probability, so that the final solution can be closer to the optimal solution. Our approach is to randomly exchange the positions of two points each time at different temperatures in the simulated annealing algorithm, and iterate repeatedly to find a better solution in this process.

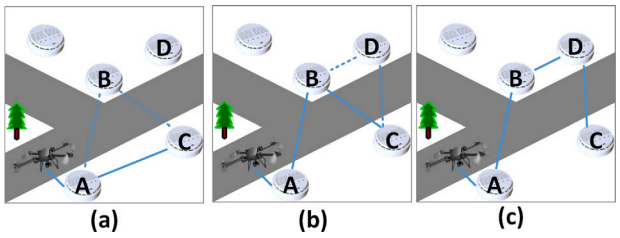


Fig. 6. UAV path adjustment strategy.

4.4.3. UAV path adjustment strategy

The first path planning in our approach can only perceive a small number of non-core IoT devices. In order to perceive as many such nodes as possible, by adjusting the flight direction of the drone, collecting more IoT device data and transmitting the verification code to more IoT devices with a very small increase in price. This also further utilizes every flight of the UAV.

The strategy to adjust the path is as follows: Take several IoT device nodes in Fig. 6 as an example. As shown in Fig. 6(a), we set the length between the two IoT devices A and C as $L_{A,C}$. By adjusting the flying direction of the UAV, we can sense more IoT devices and change the flight route from A-C to A-B-C. In order to prevent the UAV from flying to farther nodes without restriction. There is a restriction on the change of the route: $\varphi * L_{A,C} > L_{A,B} + L_{B,C}$

After changing from a straight flight path to a polyline path, compared to the original path, we can collect and transmit to more IoT devices that were not in range before, and we can also perceive more IoT devices reported by the vehicle, accelerate the verification speed and the coverage of our collection and transmission. However, if the UAV does not impose restrictions during the flight, it will result in an IoT device reported by very few or no MVs for collection and transmission. This has minimal impact on followup work, and the UAV causes a waste of energy. Therefore, we use a reward and punishment function F in the strategy, as Eq. (14) indicates that the UAV can better plan a suitable route, where Pb_i and Ps_i^c represent the modified path of the drone and the unreported path of the original path of IoT devices. Pb_i^s and Ps_i^s represents the number of IoT devices that have been reported by the vehicle but not collected by the drone. L_i^b and L_i^s respectively represent the length of the improved path and the length of the original path.

$$F = \alpha_1 \frac{Pb_i^c - Ps_i^c}{Ps_i^c} + \alpha_2 \frac{Pb_i^S - Ps_i^S}{Ps_i^S} - \alpha_3 \frac{L_i^b - L_i^S}{L_i^S} \tag{14}$$

In order to prevent the UAV from constantly changing its flight direction and accumulating a large amount of flight distance, we also set the path attenuation coefficient ω so that the UAV will not constantly change the flight direction. The improved restriction conditions are shown in Eq. (15). We illustrate by adjusting the path between the two points of AC from Fig. 6(a), the path from AC to ABC is $\omega^0 * \varphi * L_{A.C} > L_{A.B} + L_{B.C}$, when the path becomes ABDC in Fig. 6(b), $\omega^1 * \varphi * L_{B.C} > L_{B.D} + L_{D.C}$. The path will not be adjusted after the restriction conditions are met between two points, so the IoT device above will not be accessed. The final adjustment result is shown

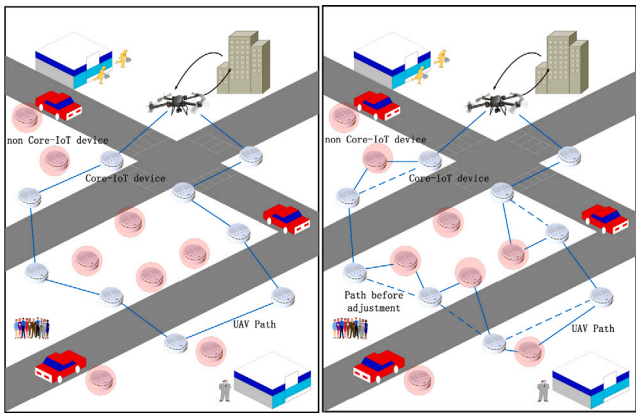


Fig. 7. Path optimization comparison.

Algorithm 2: Path planning based on simulated annealing.

Input: $L, \varphi, \alpha_1, \alpha_2, \alpha_3, num, t_0, T_0, N_0, a$
Output: NS_{path}

1: Calculate b_i according to Eq. (12) and get the S_{path}
2: Set $k = 0$, $tp = tp_0$, $TP = TP_0$, $N = N_0$
3: **while** $tp < TP$ **do**
4: **for** k in range$(1, N)$ **do**
5: Exchange two positions randomly in tmp
6: Obtain start point:Sp and end point:Ep from tmp
7: Calculate $\mathfrak{C}^{\text{now}}$ using $L_{Sp,\ldots,Ep}$
8: Set $prob = \exp\left(\left(\mathfrak{C}^{\text{now}} - \mathfrak{C}^{\text{last}}\right)/t\right)$
9: Set $Rprob$ = Random Probability
10: **if** $\mathfrak{C}^{\text{now}} < \mathfrak{C}^{\text{last}}$ or $prob < Rprob$ **then**
11: $NS_{path} = tmp$
12: **end if**
13: **end for**
14: $tp = tp * a$
15: **end while**
16: **for** each $edge \in NS_{path}$ **do**
17: Obtain start point:Sp and end point:Ep from $edge$
18: p = UVA path adjustment$(\varphi, \omega, Sp, Ep, cnt)$
19: Insert p in $edge$
20: **end for**

in Fig. 6(c), and Fig. 7 shows the final effect of the path adjustment strategy.

$$\omega^{cnt} * \varphi * L_{A,C} > L_{A,B} + L_{B,C} \tag{15}$$

The algorithm 2 of "UPA" path adjustment is as follows: (a) Obtain the core-IoT device set S_{path}, set the iteration temperature tp, the final temperature TP, the temperature attenuation factor a and the number of iterations N (lines 1–2). (b) A new path is obtained by randomly exchanging the positions of two points in each iteration, and the price of each time is compared with the optimal price. (lines 3–15). (c) After the final path NS_{path} is obtained, algorithm 3 is called recursively to adjust the strategy for each edge in the path. (lines 16–20).

5. Performance analysis

In this part, we will introduce the data set we use. We will conduct simulation experiments based on our approach under the data set. The results of simulation experiments are presented, analyzed and compare.

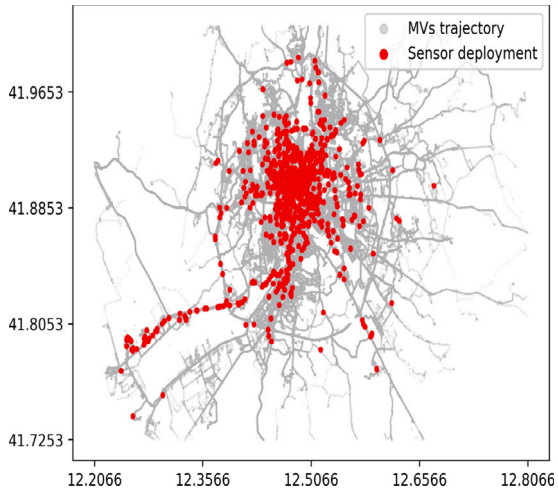


Fig. 8. Deployment diagram of IoT device.

Algorithm 3: UAVs path adjustment.

Input: $\varphi, \omega, Sp, Ep, cnt$
Output: (L_p, Mp, R_p)

1: **for** each $s_i \in S$ **do**
2: **if** $\omega^{cnt} * \varphi * L_{Sp,Ep} > L_{Sp,Mp} + L_{Mp,Ep}$ **then**
3: Calculate F_i according to Eq. (14)
4: **if** $F_i > MaxF$ **then** $MaxF = F_i$
5: **end if**
6: **end if**
7: **end for**
8: $Mp = argmax(F_i)$
9: L_p =UVA path adjustment$(\varphi, \omega, Sp, Mp, cnt + 1)$
10: R_p =UVA path adjustment$(\varphi, \omega, Mp, Ep, cnt + 1)$
11: **return** (L_p, Mp, R_p)

5.1. Experimental precondition description

In order for our simulation experiment to be in a real data environment to evaluate the effects of our approach, we obtained the 30-day trajectory in Rome, Italy through the website provided in [42].

After cleaning, analyzing and filtering the data, we finally got the latitude and longitude coordinates of the MV's trajectory. Most of their coordinates are concentrated in the longitude range of $12.21° - 13.01°$, and the latitude range is $41.72° - 42.21°$. We used a deployment method that divides the area and uses the density of the motion trajectory coordinates in the area as a probability deployment method, and randomly deploys 1000 IoT devices among these coordinate nodes, as shown in Fig. 8.

We randomly select 25% of the MVs provided in the data set as malicious MVs, and they have a 60%–80% probability to report false data. For other MVs, we set them as normal MVs and set them in the experiment. Report correct data during the process.

After the MVs collects the data in the IoT device, it sends the data back to the data center, and data center dispatches the UAVs to collect the IoT device and transmit the verification code.

In the following content, we guarantee to compare the two approaches under the same condition. we show a comparison of the three indicators of the system under "LVT-DC" and the "BD-VTE" scheme, respectively, Recognition accuracy, accuracy of judgment trust, and recognition cost.

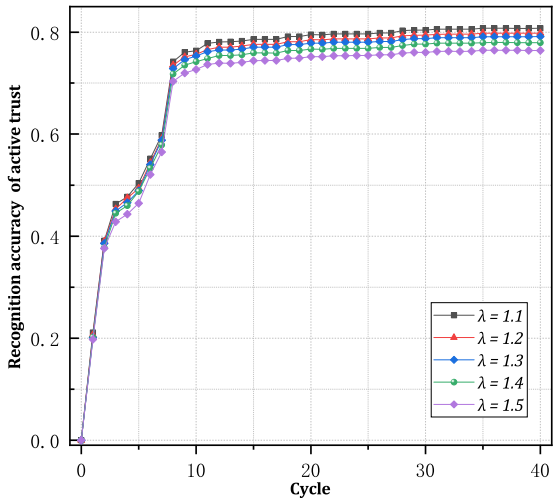


Fig. 9. The effect of λ on $\mathcal{A}_{act}$.

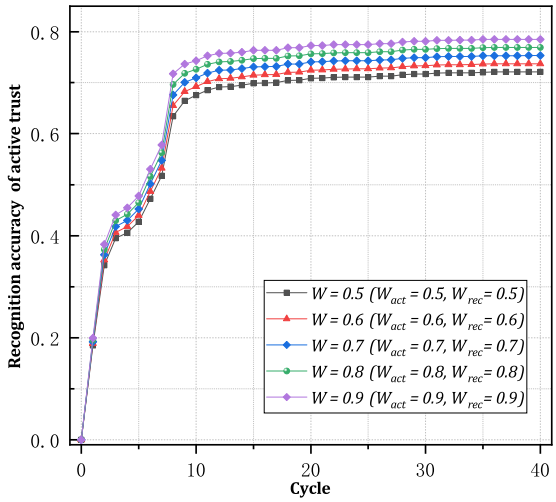


Fig. 10. $\mathcal{A}_{act}$ under different W.

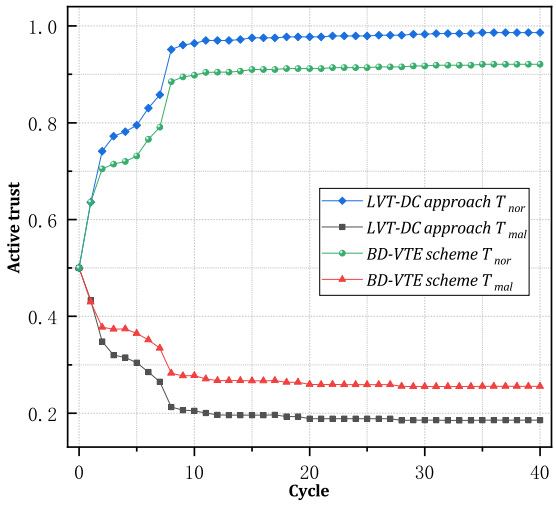


Fig. 11. Comparison of T_{act}.

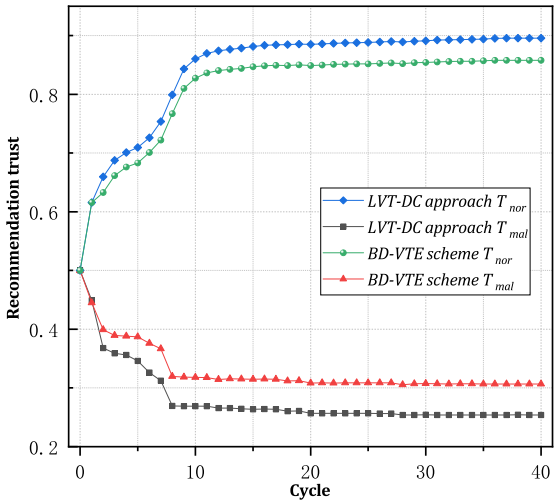


Fig. 12. Comparison of T_{rec}.

5.2. Recognition accuracy

Fig. 9 shows the impact of the penalty factor λ for MVs reporting false data on the recognition accuracy of active trust under different values. The figure shows that as the value of λ increases, the recognition accuracy of active trust is greater. This is mainly due to the increase in the value of λ, which reduces the trust of malicious vehicles. In order to ensure the rationality of the trust evaluation, we set $\lambda = 1.5$ in the experiment.

Fig. 10 shows the recognition accuracy of comprehensive trust under different values of W. The comprehensive trust degree is calculated by weighting the active trust and recommendation trust through Eq. (11). The figure shows that the greater the weight of active trust, the higher the recognition accuracy of comprehensive trust, which also confirms our discussion. As a component of the trust mechanism, recommendation trust should also be given a certain weight. Therefore, we set the weight factor $W = 0.8$, that is, $W_{act} = 0.8$, $W_{rec} = 0.2$.

Fig. 11 shows the average trust values of normal MVs and malicious MVs in each cycle of the "LVT-DC" approach and the "BD-VTE" scheme in the active trust assessment. The active trust between normal MVs and malicious MVs is calculated by Eq. (7). In the same evaluation method, because our "LVT-DC" approach has more data that can be verified for more MVs, or for the same MV, it can also verify its interaction behavior more times. Therefore, under the "LVT-DC" approach, the average trust value of normal MVs and malicious MVs is better than the "BD-VTE" scheme.

Fig. 12 shows the performance of the "LVT-DC" approach and "BD-VTE" scheme in the recommendation trust evaluation. Among them, the average trust values of normal MVs and malicious MVs under the recommendation trust evaluation are respectively shown, which are calculated by Eq. (9). Under our "LVT-DC" approach, more verification results are obtained than the latter, which means that the MV has a more reliable recommendation ability and a higher trust value. Therefore, under the same data set, compared to the "BD-VTE" scheme, the recommended trust value is better than the latter.

Fig. 13 shows the final average comprehensive trust of normal MVs and malicious MVs. Comprehensive trust is calculated by active trust and recommended trust through weighted calculation. From Figure 11 and Fig. 12, it can be seen that the ""LVT-DC"" approach is better than the ""BD-VTE"" scheme in terms of active trust and recommended trust. Therefore, the comprehensive trust obtained through calculation will also be in an advantageous state. This value is obtained from Eq. (11), which clearly shows from the figure that the "LVT-DC" approach achieves a better average trust degree of MVs and a better trust update speed.

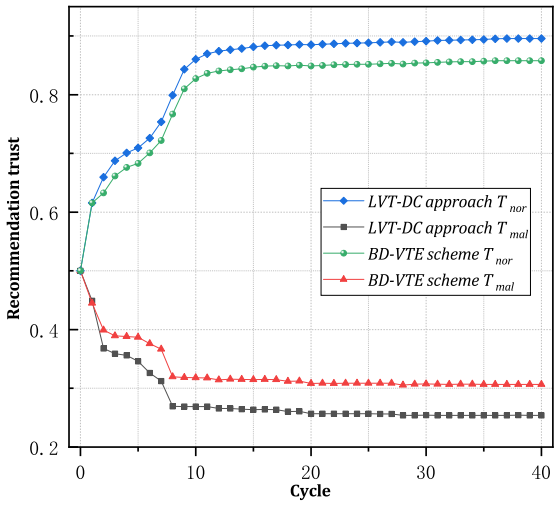


Fig. 13. Comparison of T_{com}.

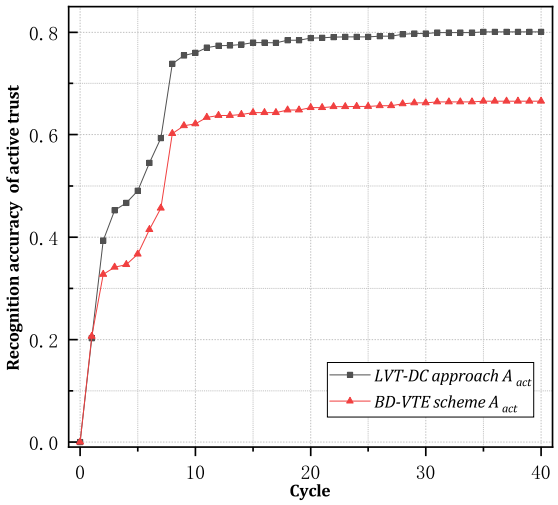


Fig. 14. Comparison of $\mathcal{A}_{act}$.

Fig. 14 shows a comparison of the recognition accuracy in active trust between the two approaches. The recognition accuracy is derived from Eq. (1), which represents the difference between the average trust value of normal MVs and malicious MVs. The higher the recognition accuracy, it means that the distinction between normal MVs and malicious MVs is more obvious. The figure shows that the "LVT-DC" approach achieves a higher difference in active trust. It can also be seen that active trust achieves a higher recognition accuracy in trust, which is suitable for assigning a more dominant weight.

Fig. 15 shows the recognition accuracy in recommendation trust between the two approaches. The figure shows that the "LVT-DC" approach achieves a higher recognition accuracy. Recommendation trust can also provide a certain reference after it is stable, and its weight should not be too low.

Fig. 16 shows the recognition accuracy of comprehensive trust under the two approaches. The figure shows that our approach achieves a higher recognition accuracy. The recognition accuracy of comprehensive trust also represents our final comparison result.

The above results indicate that the "LVT-DC" approach has a better performance in the system's recognition accuracy index than the "BD-VTE" scheme. The experimental results show that our approach achieves the same recognition accuracy rate as 8 cycles of ""BD-VTE"" scheme in 7 cycles.

According to the above experimental results, our "LVT-DC" approach has improved the value of recognition accuracy by 23.40%, and the time to achieve the same recognition accuracy has been reduced by 12.5%.

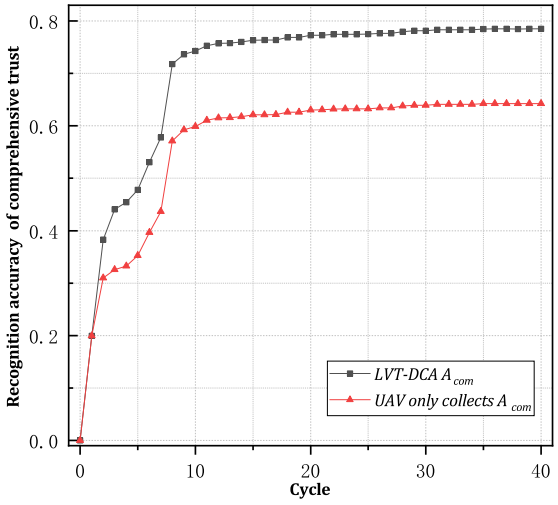


Fig. 15. Comparison of $\mathcal{A}_{rec}$.

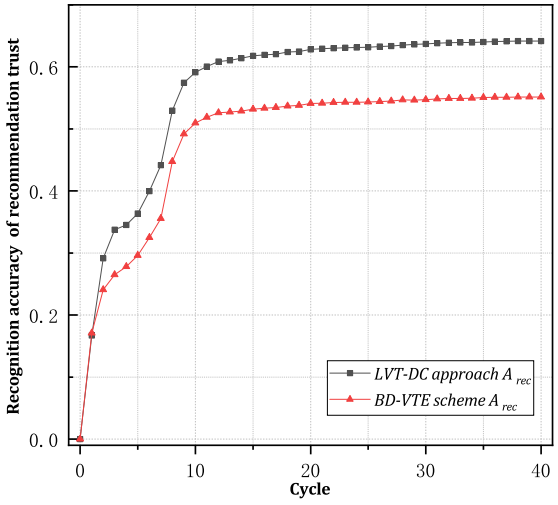


Fig. 16. Comparison of $\mathcal{A}_{com}$.

5.3. Accuracy of judgment trust

Accuracy of judgment trust represents the ability of the system to judge the trustworthiness of MVs. For the data received from MVs in the current cycle, the more authenticity can be verified, the stronger the system's ability to judge, and the more authoritative the result. Its definition and calculation method are shown in Eq. (2)

In our "LVT-DCA" approach, the number of verifiable data is increased by adding a verification code to the data of the IoT device. During the valid time of the verification code, the data containing the verification code can be verified all the time.

In the experiment, the valid time of the verification code is represented by Vt. Fig. 17 shows the influence of Vt on the accuracy of judgment trust under different conditions. It can be clearly seen in the figure that as the valid time of the verification code increases, the accuracy of the system's judgment of trust will also increase. This is

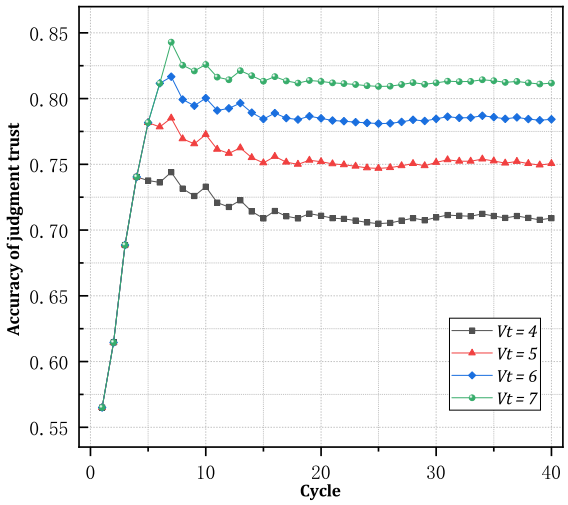


Fig. 17. Effect of different Vt.

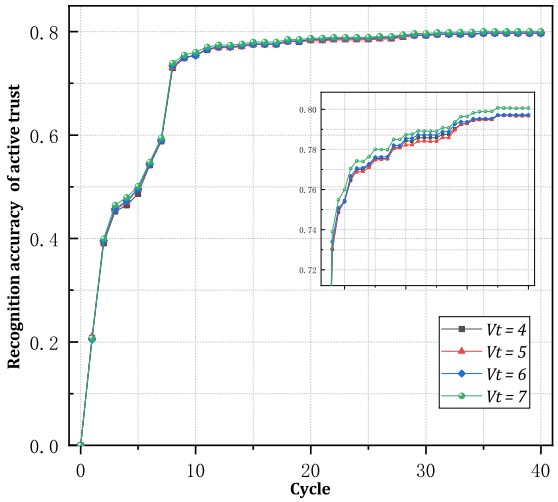


Fig. 18. Effect on $\mathcal{A}_{rec}$ under different Vt.

because the d has obtained more verifiable data in the same time. The reason for the fluctuations in the results in the figure is that in the real data set, the total number of data is not uniform in each cycle, and the number of data reported by each MV is also uneven.

Fig. 18 shows the effect of Vt on the recognition accuracy rate under different values. It can be seen in the figure that the longer the verification code is valid, the recognition accuracy rate will also be improved. However, it also reflects that the scope of the improvement brought by this is limited, and considering that in practice, the verification code time is too long and there will be security problems, so we set the valid time of the verification code to $Vt = 6$.

Fig. 19 shows the initial stage of the collection work under the "LVT-DC" approach (cycle=1). cyc represents the current cycle time, and there will be different results in different cycles. Due to the limited flight capability of the UAV, the IoT devices that can collect and transmit are still very small. As time goes by, the IoT device node that UAVs last transmitted no longer needs to repeatedly fly to the node that it has reached before, and instead goes to the core-IoT device that does not contain the verification code for transmission and collection.

We set $Vt = 6$. According to the approach of "LVT-DC" approach, the verification capability of the system will enter a relatively stable state at this time. Therefore, Fig. 20 shows the IoT device node containing the verification code at the end of the sixth cycle. At the beginning

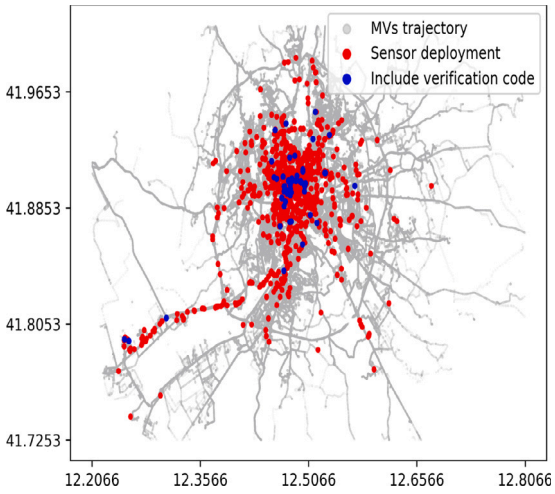


Fig. 19. Data collection of LVT-DC(c-yc=1).

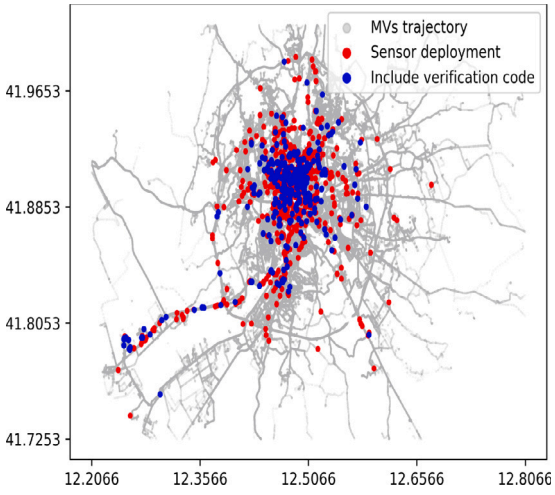


Fig. 20. Data collection of LVT-DC(c-yc=6).

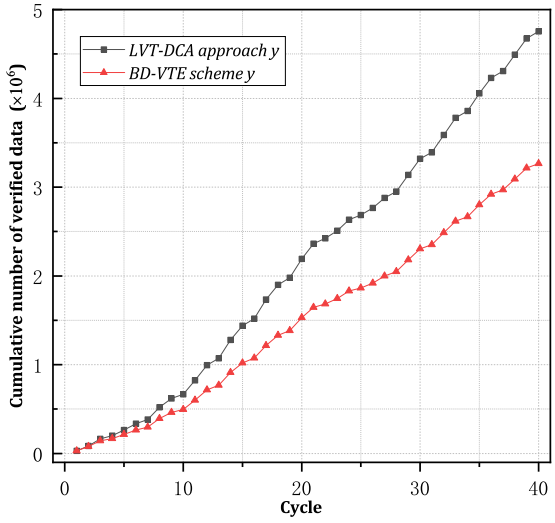


Fig. 21. Comparison of y.

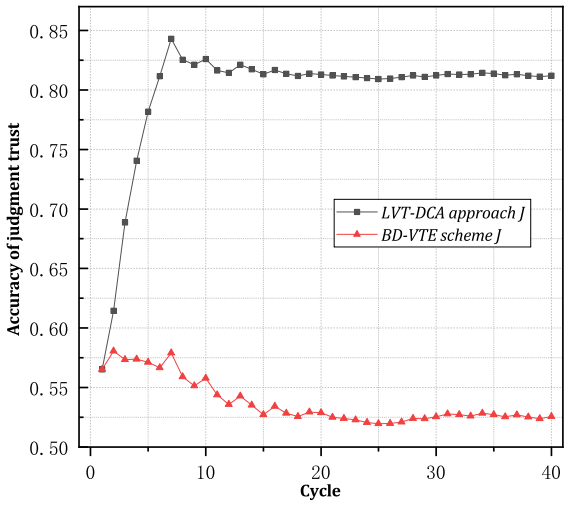


Fig. 22. Comparison of $\mathcal{J}$.

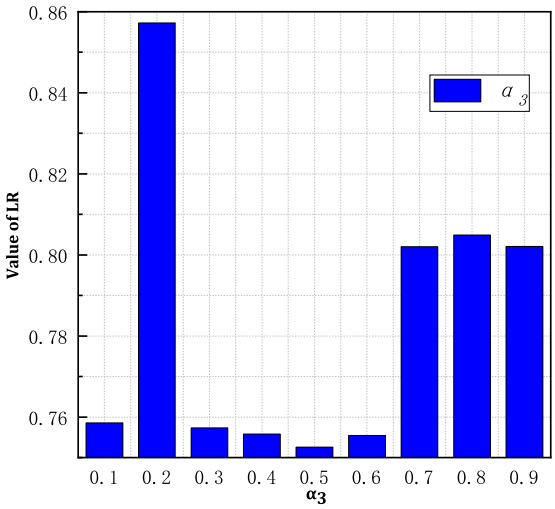


Fig. 23. Effect of Different α_3 on LR.

Table 2
Influence of different ω and φ on LR.

ω	φ			
	1.05	1.10	1.15	1.20
0.8	0.673186	0.648249	0.655969	0.651744
0.85	0.751264	0.707210	0.708764	0.707382
0.9	0.808071	0.708168	0.668230	0.800093
0.95	0.800087	0.804413	0.754884	0.804273

of the seventh cycle, the verification code set in the initial cycle has expired, and UAVs need to recollect and transmit these IoT device nodes. The number of IoT devices that can be verified will remain at a relatively stable level in the period after the verification code expires. The IoT devices with verification codes in the figure account for 32.3% of the total.

We use y to represent the number of data verified by the system in the current cycle. The more data verified, the stronger the verification ability. Fig. 21 shows the comparison of the accumulated verification data quantity under the two approaches. The figure shows that with the passage of time under "LVT-DC" approach, this number will be more and higher than the result achieved under "BD-VTE", which also illustrates the superiority of our approach.

Fig. 22 shows the comparison of the judgment trust accuracy between the "LVT-DCA" approach and the "BD-VTE" scheme in each cycle. This result is obtained by Eq. (2), which means that the data that can be verified currently accounts for the total number of data received. The reason for the fluctuation of the results is that the data set is unstable, and the report situation and the total number of data in each cycle are not uniform.

The result shows that under the "LVT-DCA" approach, when the verification code reaches the validity period, the result reaches the apex and remains at a relatively stable level in the subsequent period. Under the "BD-VTE" scheme, the result has been maintained at a relatively stable level. Under the same conditions, the "LVT-DCA" approach improves the accuracy of judgment trust to 48.60%.

5.4. Recognition cost

Recognition cost represents the cost of recognition data work performed by the system. The smaller the value, the better the use of limited resources. It is mainly determined by the cost of UAVs. In order to make better use of UAVs each flight, it covers a larger area when transmitting the verification code to the IoT device. In Section 4.4, we used the "UPA" strategy to plan the UAV's route. In this case, the drone

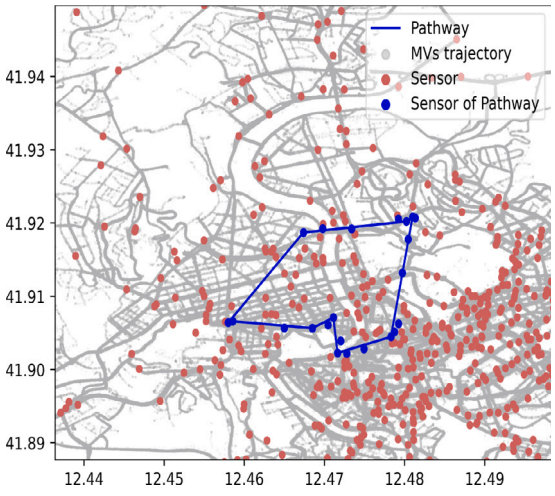


Fig. 24. The original path of UAV of "LTV-DC" approach. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

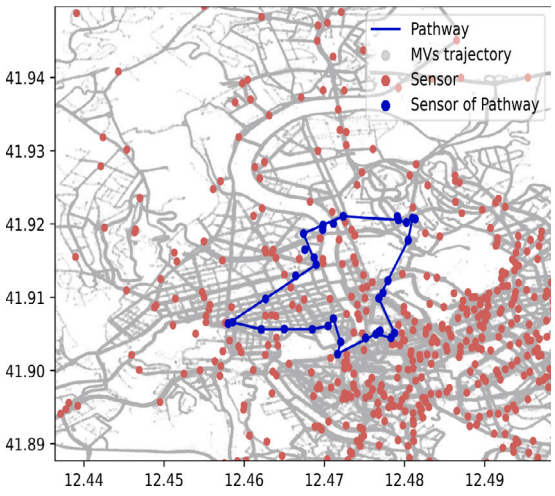


Fig. 25. The improved path of UAV of "LTV-DC" approach. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

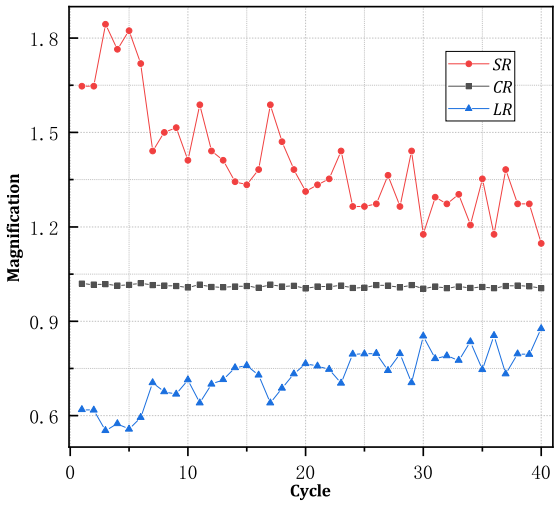


Fig. 26. Changes of LR, Sr and Cr.

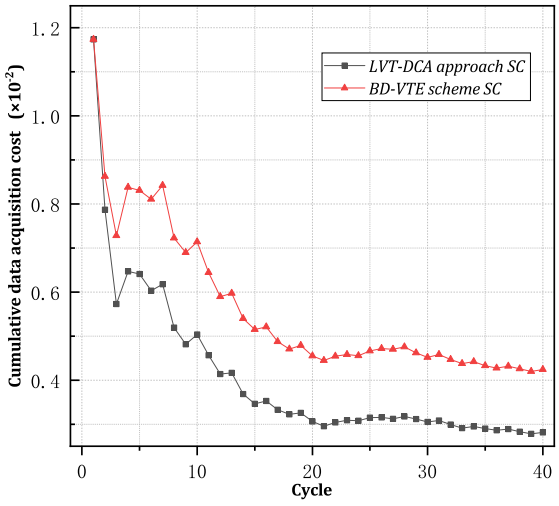


Fig. 27. Comparison of SC.

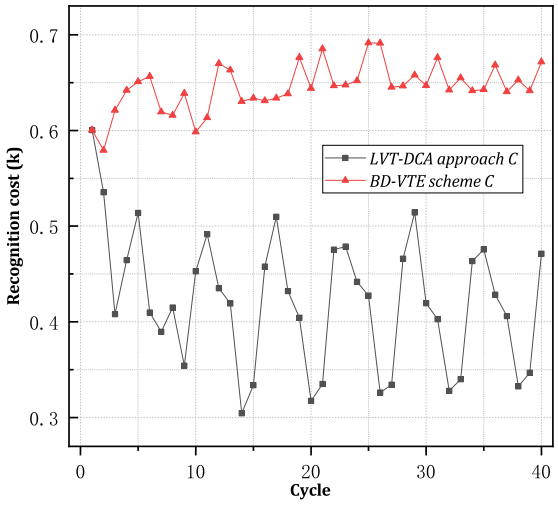


Fig. 28. Comparison of C.

can perceive more IoT device nodes, but it will also slightly increase the cost of UAVs. We set the edge weight factor φ and the broken line factor ω to limit the cost of UAVs.

$$LR = \frac{CR}{SR} \tag{16}$$

We define CR as the ratio of the cost after the improved route to the cost before the improved route, called the cost improvement rate, and SR is the ratio of the number of IoT device nodes sensed after the improved route to the number before the improved route, called the number improvement rate. LR is defined as the improvement cost factor, and its calculation method is the ratio of CR to SR, and the definition formula is given by Eq. (16). From the definition, we can find that by choosing different parameter combinations, the goal is to achieve $Min(LR)$, which means that with as many IoT devices as possible, there is a lower cost increase. By observing the simulation results of the improvement cost factor in Table 2, it can be found that when $\varphi = 1.1, \omega = 0.8$, the improvement effect of the strategy is optimal.

In the work of path adjustment, we also have an important limiting factor. The F set by Eq. (14) is used to select the direction of UAV path change. Among them, α_1 and α_2 are used as reward factors, and we take the same reward proportion as 0.5. α_3 is used as a penalty factor, and its different values will make the improvement cost factor have different results. From the loss rate simulation result of Fig. 23, it can be seen that after $\alpha_3 = 0.3$, the improvement cost factor gradually stabilizes. There is a jump when $\alpha_3 = 0.6$, which proves that the penalty is too high, so we take $\alpha_3 = 0.5$.

Figs. 24 and 25 are the results of UAV simulation flight in the real data set area after determining the parameters of each part of the formula. The red dots represent the sensor nodes in the area, and the blue dots represent the nodes interacting with the UAV.

Fig. 24 is the optimal path planned under the simulated annealing algorithm, and Fig. 25 is the UAV path adjusted by the "UPA" strategy. Through the comparison of the two figures, it can be clearly seen that the number of IoT devices sensed by the UAV has increased.

Fig. 26 shows the change in cost improvement rate, number improvement rate, and improvement cost factor for each cycle in the simulation experiment. SR is the number improvement rate. It can be seen that the number of IoT devices sensed by UAVs has increased significantly. The reason for the fluctuation of SR is that the location of IoT devices is randomly deployed, so different target sets will have different routes. CR is the cost improvement rate. The figure shows that the increased cost of the "UPA" path planning algorithm used is between 0.4% and 2.1%, which also shows the effectiveness of the path planning strategy. The purpose is to perceive more IoT devices by adding very little cost.

We define the cost of UAVs through Eq. (13). The maximum operating speed of the UAV can reach 70 km/h. In order to keep the UAVs working in a stable state, we set the UAV's speed to maintain 35 km/h in the experiment, and the cost factor τ is set to 100.

The cumulative cost SC of acquiring data is defined by the following Eq. (17). The calculation method is the ratio of the cumulative number of verifiable data acquired to the cost of recognition recognizing these data. Where y is the same as the cumulative amount of verifiable data obtained as defined above, and x represents the cumulative cost of UAVs.

Fig. 27 shows a comparison of the "LVT-DC" approach and the "BD-VTE" scheme in this respect. The figure clearly reflects that, as time goes by, because the amount of verifiable data obtained under the "LVT-DC" approach is greater, the cost utilization rate is better. As the results accumulate, our approach can save more costs.

$$SC = \frac{y}{x} \tag{17}$$

Fig. 28 shows the comparison of the recognition cost between the "LVT-DCA" approach and the "BD-VTE" scheme, which is defined in Eq. (3). This value represents the system's ability to utilize the cost

of UAVs. The lower the value, the more verifiable data that can be obtained through limited costs.

As we described before, the target Core-IoT device collected in each cycle is often concentrated in the area with the highest vehicle trajectory density. Under the "BD-VTE" scheme, it is necessary to collect these devices repeatedly in this case. This means that the route of each flight is almost the same, so the cost of UAVs fluctuates in a small range.

Under our "LVT-DCA" approach, because UAVs do not interact with IoT devices that contain verification codes in the current cycle, they fly to those devices that do not contain verification codes. This means that the route of UAVs under our approach is different each time, so the cost of UAVs consumed will fluctuate. This is the reason why the data fluctuates under the "LVT-DC" approach.

The figure shows that, in addition to the initial cycle (at this time, the two approaches are the same in all aspects), the recognition cost in each cycle is at a lower level. Compared with the "BD-VTE" scheme, the "LVT-DCA" approach reduces the recognition cost by 34.62%.

6. Conclusion and future work

The most important function of Sensor–Cloud System (SCS) is to sense and obtain data, which is constructed into various applications in the cloud. Therefore, low cost, reliable data acquisition is one of its most critical issues. We propose a Lightweight Verifiable Trust based Data Collection (LVT-DC) approach to obtain credible data through to effectively evaluate the credibility of MVs. The essential innovation of LVT-DC approach is to send a verifiable verification code to IoT devices of perception data in advance, so that after the cloud obtains the data reported by the data collector, it can be compared, which can effectively determine the trust of MVs, and through a series of trust reasoning to enrich the trust. Since the method of using the verification code can continuously achieve the purpose of verification during the validity period of the verification code, it is more effective than the previous proposed methods, and the cost is lower, which makes the acquisition and trust evaluation more accurate.

In the era of big data, many applications rely on the effectiveness of collected data. Therefore, credible data collection plays a key role in the widespread use of these applications. The method proposed in this article is not only suitable for the data collection application scenarios proposed in this article, but also for data collection such as mobile crowdsensing network, the method of this article can be appropriately improved and applied. The method proposed in this article is not only suitable for the data collection application scenarios proposed in this article, but also for data collection such as mobile crowdsensing network, the method in this article can be appropriately modified and then applied. Our future work is not only limited to the application of this method, but more importantly, our future work intends to research the combination of credibility and privacy in data collection, so as to provide high quality data for future advanced applications.

CRediT authorship contribution statement

Jiawei Guo: Writing - original draft, Software. **Haoyang Wang:** Software. **Wei Liu:** Conceptualization. **Guosheng Huang:** Methodology. **Jinsong Gui:** Data curation. **Shaobo Zhang:** Software.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

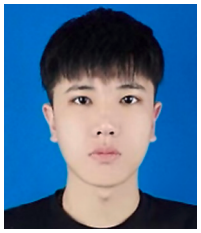
Acknowledgment

This work was supported in part by the National Natural Science Foundation of China (No. 62072475, No. 61772554).

References

[1] J. Sengupta, S. Ruj, S.D. Bit, A secure fog-based architecture for industrial internet of things and industry 4.0, IEEE Trans. Ind. Inf. 17 (4) (2020) 2316–2324.
[2] A.M. Ghosh, K. Grolinger, Edge-cloud computing for internet of things data analytics: Embedding intelligence in the edge with deep learning, IEEE Trans. Ind. Inf. 17 (3) (2021) 2191–2200.
[3] B. Marr, How much data do we create every day? The mind-blowing stats everyone should read, in: Forbes, 2019, pp. 1–5.
[4] A. Thiagarajan, L. Ravindranath, K. LaCurts, S. Madden, H. Balakrishnan, S. Toledo, J. Eriksson, Vtrack: accurate, energy-aware road traffic delay estimation using mobile phones, in: Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems, 2009, pp. 85–98.
[5] N. Maisonneuve, M. Stevens, M.E. Niessen, L. Steels, Noisetube: Measuring and mapping noise pollution with mobile phones, in: Information Technologies in Environmental Engineering, Springer, 2009, pp. 215–228.
[6] T. Wang, G. Zhang, M.Z.A. Bhuiyan, A. Liu, W. Jia, M. Xie, A novel trust mechanism based on fog computing in sensor–cloud system, Future Gener. Comput. Syst. 109 (2020) 573–582.
[7] T. Wang, Y. Lu, J. Wang, H.-N. Dai, X. Zheng, W. Jia, EIHDP: Edge-intelligent hierarchical dynamic pricing based on cloud-edge-client collaboration for IoT systems, IEEE Trans. Comput. (2021) http://dx.doi.org/10.1109/TC.2021.3060484.
[8] H. Teng, M. Dong, Y. Liu, W. Tian, X. Liu, A low-cost physical location discovery scheme for large-scale internet of things in smart city through joint use of vehicles and UAVs, Future Gener. Comput. Syst. 118 (2021) 310–326.
[9] F. Li, G. Huang, Q. Yang, M. Xie, Adaptive contention window MAC protocol in a global view for emerging trends networks, IEEE Access 9 (2021) 18402–18423.
[10] Y. Liu, A. Liu, S. Guo, Z. Li, Y.-J. Choi, H. Sekiya, Context-aware collect data with energy efficient in cyber–physical cloud systems, Future Gener. Comput. Syst. 105 (2021) 932–947.
[11] T. Li, W. Liu, T. Wang, Z. Ming, X. Li, M. Ma, Trust data collections via vehicles joint with unmanned aerial vehicles in the smart internet of things, Trans. Emerg. Telecommun. Technol. (2020) e3956, http://dx.doi.org/10.1002/ett.3956.
[12] M. Yu, A. Liu, N.N. Xiong, T. Wang, An intelligent game based offloading scheme for maximizing benefits of IoT-edge-cloud ecosystems, IEEE Int. Things J. (2020) http://dx.doi.org/10.1109/JIOT.2020.3039828.
[13] X. Zhu, Y. Luo, A. Liu, M.Z.A. Bhuiyan, S. Zhang, Multi-agent deep reinforcement learning for vehicular computation offloading in IoT, IEEE Internet Things J. 8 (2021).
[14] J. Luo, X. Deng, H. Zhang, H. Qi, QoE-driven computation offloading for edge computing, J. Syst. Archit. 97 (2019) 34–39.
[15] S. Huang, Z. Zeng, K. Ota, M. Dong, T. Wang, N.N. Xiong, An intelligent collaboration trust interconnections system for mobile information control in ubiquitous 5G networks, IEEE Trans. Netw. Sci. Eng. 8 (1) (2020) 347–365.
[16] T. Wang, P. Wang, S. Cai, X. Zheng, Y. Ma, W. Jia, G. Wang, Mobile edge-enabled trust evaluation for the Internet of Things, Inf. Fusion 75 (2021) 90–100, http://dx.doi.org/10.1016/j.inffus.2021.04.007.
[17] W. Mo, T. Wang, S. Zhang, J. Zhang, An active and verifiable trust evaluation approach for edge computing, J. Cloud Comput. 9 (1) (2020) 1–19.
[18] T. Li, A. Liu, N.N. Xiong, S. Zhang, T. Wang, A trustworthiness-based vehicular recruitment scheme for information collections in distributed networked systems, Inform. Sci. 545 (2021) 65–81.
[19] S. Huang, J. Gui, T. Wang, X. Li, Joint mobile vehicle–UAV scheme for secure data collection in a smart city, Ann. Telecommun. (2020) 1–22, http://dx.doi.org/10.1007/s12243-020-00798-9.
[20] M. Shen, A. Liu, G. Huang, N.N. Xiong, H. Lu, ATTDC: An active and trace-able trust data collection scheme for industrial security in smart cities, IEEE Internet Things J. (2020).
[21] M. Bonola, L. Bracciale, P. Loreti, R. Amici, A. Rabuffi, G. Bianchi, Opportunistic communication in smart city: Experimental insight with small-scale taxi fleets as data carriers, Ad Hoc Netw. 43 (2016) 43–55.
[22] Y. Ouyang, A. Liu, N. Xiong, T. Wang, An effective early message ahead join adaptive data aggregation scheme for sustainable IoT, IEEE Trans. Netw. Sci. Eng. 8 (1) (2021) 201–219.
[23] C. Huang, G. Huang, W. Liu, R. Wang, M. Xie, A parallel joint optimized relay selection protocol for wake-up radio enabled WSNs, Phys. Commun. (2021) 101320, http://dx.doi.org/10.1016/j.phycom.2021.101320.
[24] X. Liu, A. Liu, T. Wang, K. Ota, M. Dong, Y. Liu, Z. Cai, Adaptive data and verified message disjoint security routing for gathering big data in energy harvesting networks, J. Parallel Distrib. Comput. 135 (2020) 140–155.
[25] J. Guo, A. Liu, K. Ota, M. Dong, X. Deng, N. Xiong, ITCN: An intelligent trust collaboration network system in IoT, IEEE Trans. Netw. Sci. Eng. (2021) http://dx.doi.org/10.1109/TNSE.2021.3057881.
[26] Y. Ouyang, Z. Zeng, X. Li, T. Wang, X. Liu, A verifiable trust evaluation mechanism for ultra-reliable applications in 5G and beyond networks, Comput. Stand. Interfaces 77 (2021) 103519.
[27] T. Wang, Y. Liu, X. Zheng, H.-N. Dai, W. Jia, M. Xie, Edge-based communication optimization for distributed federated learning, IEEE Trans. Netw. Sci. Eng. (2021) http://dx.doi.org/10.1109/TNSE.2021.3083263.

[28] Y. Liu, T. Wang, S. Zhang, X. Liu, X. Liu, Artificial intelligence aware and security-enhanced traceback technique in mobile edge computing, Comput. Commun. 161 (2020) 375–386.
[29] S. Zhu, Z. Cai, H. Hu, Y. Li, W. Li, Zkcrowd: A hybrid blockchain-based crowdsourcing platform, IEEE Trans. Ind. Inf. 16 (6) (2020) 4196–4205.
[30] Y. Ren, Z. Zeng, T. Wang, S. Zhang, G. Zhi, A trust-based minimum cost and quality aware data collection scheme in P2P network, Peer-To-Peer Netw. Appl. (2020) 1–24.
[31] S. Huang, A. Liu, S. Zhang, T. Wang, N. Xiong, BD-VTE: a novel baseline data based verifiable trust evaluation scheme for smart network systems, IEEE Trans. Netw. Sci. Eng. (2020) http://dx.doi.org/10.1109/TNSE.2020.3014455.
[32] X. Deng, J. Li, E. Liu, H. Zhang, Task allocation algorithm and optimization model on edge collaboration, J. Syst. Archit. 110 (2020) 101778.
[33] Z. Cai, X. Zheng, A private and efficient mechanism for data uploading in smart cyber-physical systems, IEEE Trans. Netw. Sci. Eng. 7 (2) (2020) 766–775.
[34] J. Singh, R. Kaur, D. Singh, A survey and taxonomy on energy management schemes in wireless sensor networks, J. Syst. Archit. (2020) 101782.
[35] S. Ahmadian, M. Afsharchi, M. Meghdadi, An effective social recommendation method based on user reputation model and rating profile enhancement, J. Inf. Sci. 45 (5) (2019) 607–642.
[36] T.D. Nguyen, Q. Bai, Enhance trust management in composite services with indirect ratings, Comput. J. 60 (11) (2017) 1619–1632.
[37] X. Zheng, Z. Cai, Privacy-preserved data sharing towards multiple parties in industrial IoTs, IEEE J. Sel. Areas Commun. 38 (5) (2020) 968–979.
[38] K. Renuka, S. Kumari, X. Li, Design of a secure three-factor authentication scheme for smart healthcare, J. Med. Syst. 43 (5) (2019) 133.
[39] S. Kumari, K. Renuka, A provably secure biometrics and ECC-based authentication and key agreement scheme for WSNs, Int. J. Commun. Syst. 33 (3) (2020) e4194.
[40] Y.A. Kim, R. Phalak, A trust prediction framework in rating-based experience sharing social networks without a Web of Trust, Inform. Sci. 191 (2012) 128–145.
[41] M. Rahman, A. Rahman, H.-J. Hong, L.-W. Pan, M.Y. Sarwar Uddin, N. Venkatasubramanian, C.-H. Hsu, An adaptive IoT platform on budgeted 3G data plans, J. Syst. Archit. (ISSN: 1383-7621) 97 (2019) 65–76.
[42] M. Piorkowski, N. Sarafijanovic-Djukic, M. Grossglauser, CRAWDAD data set epfl/mobility (v. 2009-02-24), 2009, https://crawdad.org/epfl/mobility/20090224.
[43] C. Ge, X. Ma, Z. Liu, A semi-autonomous distributed blockchain-based framework for UAVs system, J. Syst. Archit. 107 (2020) 101728.
[44] S. Zhu, W. Li, H. Li, L. Tian, G. Luo, Z. Cai, Coin hopping attack in blockchain-based IoT, IEEE Internet Things J. 6 (3) (2018) 4614–4626.
[45] M. Taboga, Lectures on Probability Theory and Mathematical Statistics, CreateSpace Independent Publishing Platform, 2017, https://www.statlect.com/probability-distributions/beta-distribution.
[46] S. Shekhar, A. Chhokra, H. Sun, A. Gokhale, A. Dubey, X. Koutsoukos, G. Karsai, URMILA: Dynamically trading-off fog and edge resources for performance and mobility-aware IoT services, J. Syst. Archit. 107 (2020) 101710.

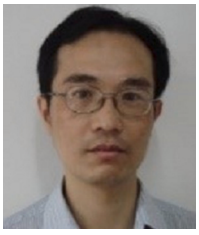


Jiawei Guo is currently studying for a master's degree in computer science and technology at the School of Computer Science, Central South University, China. His research interests include edge computing, crowd sensing networks, and wireless sensor networks. E-mail: jinyi109@csu.edu.cn

Haoyang Wang is currently a student master's degree with the School of Information Science and Engineering, Central South University, China. His research interests include Internet of things and cloud computing. E-mail: wanghaoyang@csu.edu.cn

Wei Liu received the Ph.D. degree in computer application technology from Central South University, Changsha, China, in 2014. He is an Associate Professor and a Senior Engineer with the School of Informatics, Hunan University of Chinese Medicine, Changsha. His research interests include complex network analysis, software engineering and medical informatics. He has published more than 40 papers in related fields. E-mail: weiliu@csu.edu.cn

Guosheng Huang received his M.S. and Ph.D. degrees in computer science from Central South University, China, 2001 and 2010 respectively. He was a visiting scholar with Sun Yat-Sen University, From 2017 to 2018. Currently, he is an associate professor of School of Information Science and Engineering, Hunan First Normal University, China. His major research interests include MIMO techniques, wireless sensor network and mobile computing. E-mail: hgsheng@mail.sysu.edu.cn



Jinsong Gui received the BE from the University of Shanghai for Science and Technology, China, in 1992, and the M.S. and Ph.D. from Central South University, China, in 2004 and 2008, respectively. He is currently a Professor in School of Computer Science and Engineering, Central South University. He is the member of China Computer Federation (CCF) and the IEEE member. He published over 50 international journal papers and over 10 international conference papers. His research interests cover the general area of distributed systems, as well as related fields such as wireless network topology control, cloud and green computing, network trust and security. E-mail: jsgui2010@csu.edu.cn

Shaobo Zhang received the B.Sc. and M.Sc. degree in computer science both from Hunan University of Science and Technology, Xiangtan, China, in 2003 and 2009 respectively, and the Ph.D. degree in computer science from Central South University, Changsha, China, in 2017. He is currently a Lecture at School of Computer Science and Engineering of the Hunan University of Science and Technology, China. His research interests include privacy and security issues in social networks and cloud computing. E-mail: shaobozhang@hnust.edu.cn