IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 19, 2019, accepted February 7, 2019, date of publication March 18, 2019, date of current version April 5, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2902811

Blockchain Trust Model for Malicious Node Detection in Wireless Sensor Networks

WEI SHE[1,2,3], QI LIU[1], ZHAO TIAN[1], JIAN-SEN CHEN[2], BO WANG[4], (Member, IEEE), AND WEI LIU[1,2]

[1]School of Software Technology, Zhengzhou University, Zhengzhou 450000, China
[2]Collaborative Innovation Center for Internet Healthcare, Zhengzhou University, Zhengzhou 450000, China
[3]Water Environment Governance and Ecological Restoration Academician Workstation of Henan Province, Zhengzhou 450002, China
[4]Department of Computer Science, The State University of New York at Buffalo, Amherst, NY 14260, USA

Corresponding author: Wei Liu (wliu@zzu.edu.cn)

This work was supported in part by the Natural Science Foundation of China under Grant 61602422, in part by the National Key R&D Program of China under Grant SQ2018YFB1201403, in part by the Key Science and Technology Program of Henan Province under Grant 162102310536, and in part by the CERNET Innovation Project under Grant NGII20160705 and Grant NGII20180702.

ABSTRACT The Internet of Things (IoT) has been widely used because of its high efficiency and real-time collaboration. A wireless sensor network is the core technology to support the operation of the IoT, and the security problem is becoming more and more serious. Aiming at the problem that the existing malicious node detection methods in wireless sensor networks cannot be guaranteed by fairness and traceability of detection process, we present a blockchain trust model (BTM) for malicious node detection in wireless sensor networks. First, it gives the whole framework of the trust model. Then, it constructs the blockchain data structure which is used to detect malicious nodes. Finally, it realizes the detection of malicious nodes in 3D space by using the blockchain smart contract and the WSNs' quadrilateral measurement localization method, and the voting consensus results are recorded in the blockchain distributed. The simulation results show that the model can effectively detect malicious nodes in WSNs, and it can also ensure the traceability of the detection process.

INDEX TERMS Wireless sensor networks, blockchain, smart contract, malicious nodes, vote.

I. INTRODUCTION

With the rapid development of Internet and perceptual technology, Internet of Things (IoT) [1] emerged as an important force to promote economic and social development. As the key technology in the IoT architecture, wireless sensor network (WSNs) plays a key role in promoting the development of IoT, which has remarkable practical significance and research value. Wireless sensor networks [2], [3] are mobile ad hoc networks (MANETs) composed of distributed micro-devices embedded with various sensing abilities (call sensors), which have the characteristics of the wide coverage area, high precision monitoring, remote monitoring, rapid deployment, self-organization and high fault tolerance. Now WSNs are widely used in numerous areas, such as military, smart home, commercial and other fields [4]. However, the sensor nodes have certain limitations—they are easily damaged and have limited power, computational ability, memory and transmission range [5]. Moreover, they can be easily compromised by an adversary [6]. According to statistics, the threats faced by WSNs mainly come from two aspects [7], [8]. On the one hand, the external attacker attacks the network, and on the other hand, the internal node is invaded and controlled to become a malicious node to launch an attack from within. Hence, it is an urgent security problem for wireless sensor networks to have the ability to identify and eliminate internal malicious nodes. And how to solve this security problem has a profound impact on the healthy development of the IoT [9]. For this reason, the network security of WSNs has attracted the attention of many researchers.

The problem of malicious node detection in wireless sensor networks has been widely studied. The methods can be divided into two categories: one is based on the trust model and the other is based on WSNs protocol. The former is the most common method.

In order to solve the trust problem of the malicious node detection in WSN, Zawaideh *et al.* [10] improved the algorithm of determining based on neighbor weight trust (NWTD). The algorithm periodically updated the trust degree of nodes and set the minimum threshold of acceptable

The associate editor coordinating the review of this manuscript and approving it for publication was Tie Qiu.

trust for nodes. Thus, it can realize the separated malicious nodes. Aiming at the problem of malicious node detection in WSNs, Zeng *et al.* [11] provided a trust mechanism based on D-S (Dempster-Shafer) evidence theory to consider the indirect and direct trust of third-party nodes. It ensured the robustness of the network and the validity of the data packet. Compared with [11], Su *et al.* [12] proposed a trust model based on the calculation of trust degree and considered both direct and indirect trust levels with the considering the internal attacks faced by wireless sensor networks. This model reduced the network energy consumption and set the trust threshold to assist the decision through the periodically updating the trust degree. And it can also effectively distinguish the malicious nodes to ensure the security and reliability of the network from the aging nodes. In order to solve the uncertainty of the decision made by the traditional trust mechanism, it regarded the message success rate, node delay, correctness and fairness as trust measures. Prabha and Latha [13] proposed a multi-attribute trust model which was based on fuzzy processing to calculate the final trust value of each node. It proved the accuracy of the decision. Zhang *et al.* [14] proposed a new trust management scheme based on D-S evidence theory. Firstly, considering the spatio-temporal correlation of data collected by adjacent sensor nodes, and then according to D-S theory, the trust model was established. Finally, the whole trust degree was calculated to identify the malicious nodes. To solve the problems of the single detection function by malicious node identification system and the inability, Yang *et al.* [15] proposed a new malicious node recognition model to resist malicious libel behavior of high-reputation nodes in existing WSNs. This model presented the indirect credibility of the third-party nodes and the reputation distribution by using Beta Distribution, and integrated the trust values corresponding to various attack types to ensure the accurate identification of malicious nodes.

Besides the malicious node method based on the trust mechanism, another kind of protocol detection method based on WSNs is also widely used. In order to solve the low processing capacity of WSNs, Das and Das [16] proposed an enhanced LEACH protocol, which can detect energy consumption, distance and malicious nodes to ensure the robustness of wireless sensor networks. To detect malicious nodes and enhance network security, Chen *et al.* [17] presented a new multi-valued trust routing protocol (MTR) that is based on sensor trust and the number of hops from sensor nodes to base station (BS) to ensure network stability. To solve the problem of malicious nodes spoofing their identity and location in WSNs, Atassi *et al.* [18] proposed a decentralized malicious node detection technique based on received signal strength indicator (RSSI). To solve the problem of identity attacks by malicious nodes through wireless signals in wireless network communications, Pinto *et al.* [19] proposed a new strategy based on machine learning. It used two classifiers to process and analyze real-time samples of received signal strength, and optimize the case of legitimate nodes and attack nodes whose landmarks are close to each other. Althunibat *et al.* [20] proved that both dependent and independent malicious nodes have the same effect on the overall performance of WSNs in terms of detection and false alarm rate. Uddin *et al.* [21] proposed a model for detecting power distribution side fault points by using WSNs technology. Based on current sensor and wireless protocols, the model realizes the process of wireless monitoring, detecting and locating fault nodes.

The above literatures proposed effective methods to the problem of malicious node detection in wireless sensor networks. However, they did not mention how to record the detection process of malicious nodes, nor the mechanism to safely save the original data for later traceability.

The emergence of Blockchain technology [22], [23] and smart contract [24], [25] provides a new way for the detection of malicious nodes in wireless sensor networks. Smart contract can automatically and distributedly perform predefined operations when abnormal behavior occurs or boundary conditions are triggered, and all relevant data are formed into data blocks that can be traced, verified, and with timing characteristics. Ellul and Pace [26] proposed a smart contract for detecting virtual machines, which realized the interaction between the blockchain system and the IoT device under limited conditions, and promoted the automation of verifiable blockchain transactions. Islam and Kundu [27] proposed a smart contract based on blockchain to guarantee personal privacy and information security in the home-sharing economy. Kang *et al.* [28] combined smart contract with blockchains and proposed a renewable energy trading platform that can automatically implement transactions. Likewise, Pan *et al.* [29] proposed an Edge Chain framework for edge devices in the IoT [30] by using smart contract to implement blockchain management of behavior, resources and accounts of IoT edge devices. Zhang *et al.* [31] proposed a framework based on smart contract to solve the problems of trusted access control and distributed in the IoT. In this paper, it proposes a blockchain trust model for malicious node detection in WSNs. It provides a more secure, credible and reliable solution for malicious node detection and traceability of detection process in WSNs. It not only restrains the interference of malicious nodes to the normal operation of the network, but also ensures the transparency and traceability of the detection process. At the same time, it avoids the space limitation to some extent.

The paper is structured as follows. In Sec. II, we propose a blockchain trust model for malicious node detection in WSNs; Sec. III carries out simulation experiments and analysis of the model; and Sec. IV includes the summary of the paper and future work.

II. BTM FOR MALICIOUS NODE DETECTION IN WSNs

In this section, we propose a blockchain trust model for malicious node detection in wireless sensor networks. First of all, we introduce the overall structure involved in the blockchain trust model in Sec. II-A. The data structure of the model is constructed in Sec. II-B. The smart contract

of the model is designed in Sec. II-C where we present the formal expression of the smart contract and discuss indicators of evaluating malicious nodes by smart contract, then we give WSN location method in smart contract.

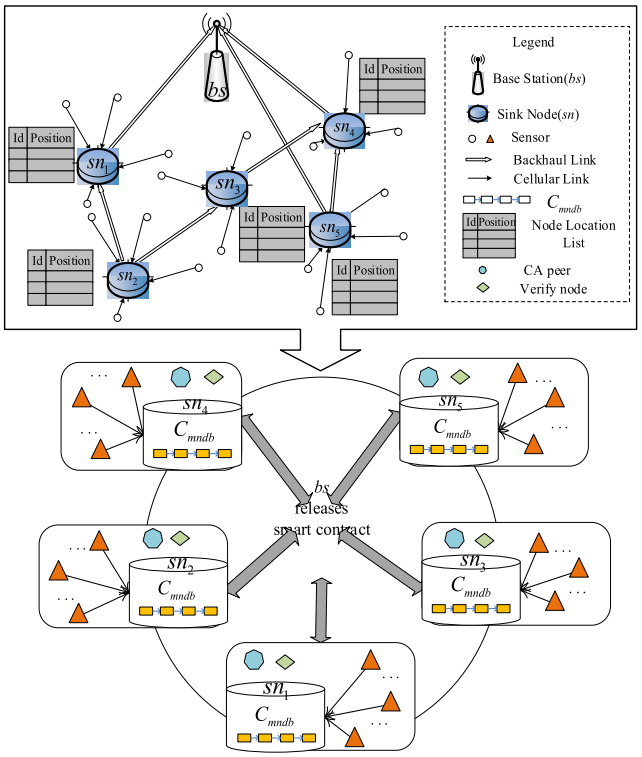

FIGURE 1. An illustration of BTM for malicious node detection in WSNs.

A. MODEL ARCHITECTURE

As shown in the top half of Fig. 1, in wireless sensor networks, we consider dividing the nodes into *bs*, *sn* and *sensor* [32]. The *sensor* monitors the indicators of the area in real time, collects monitoring data, understands the running status, and uploads the integrated data to the associated *sn* through the cellular link; *sn* collects all the monitoring data uploaded by the *sensor* in the transmission range, analyzes the running status of the *sensor* in real time, and collects the results to the *bs* center through the backhaul network.

As shown in the lower part of Fig. 1, we map the operating framework of the wireless sensor network into the Consortium Blockchain. There are four main types of nodes in the Consortium Blockchain: contract issuing node, CA node, verification node and common nodes. *bs* is the contract publishing node, responsible for publishing intelligent contracts, as the issuer of activities.

The *sn* serves as both CA node and verification node. CA node provides digital certificate-based identity information to members of the Consortium Blockchain community (Each *sn* and the *sensor* in the communication range is an alliance), and can generate or cancel a member's identity certificate. On the basis of clear membership, the organization can implement the management of authority control. The verification node is served by the pre-selected *sn*. It is mainly responsible for receiving the monitoring data collected by the common node, processing the smart contract, checking the legality of the transaction data, and updating and maintaining the node data and the account status in the blockchain organization. Among them, the smart contract is a piece of code that is deployed on the distributed ledger, which can control the received external information. In particular, the generation of each block is determined by all pre-selected nodes, and stored in C_{mndb}. The sensor is a normal node and only uploads the collected monitoring data, regardless of the accounting process.

Therefore, based on their common characteristics, the overall structure of the wireless sensor network can be mapped into the Consortium Blockchain network, and then the detection of malicious nodes in the blockchain network. The blockchain trust model (BTM) for malicious node detection in WSNs is formalized as follows:

Definition 1: BTM is a 8-tuple set.

$$(BS, AN, SENSOR, C_{mndb}, SC, T, \alpha, \beta)$$

where:

1) $BS = \{bs_i | i \in \mathbb{N}^+\}$ is the finite set of base stations.

2) $SN = \{sn_j | j \in \mathbb{N}^+\}$ is the finite set of cluster heads.

3) $SENSOR = \{sensor_k | k \in \mathbb{N}^+\}$ is the finite set of sensors.

4) C_{mndb} is a malicious node detection blockchain.

5) SC is the smart contract of C_{mndb}. (see Sec.II-C-1).

6) $T = \{t_f | t_f \in SENSOR \times SN \vee SN \times BS, f \in \mathbb{N}^+\}$ is the transaction set of nodes. $SENSOR \times SN$ is a Cartesian set of *sensor* and SN. $SN \times BS$ is a Cartesian set of SN and BS.

7) $\alpha : S \vee SN \rightarrow C_{mndb}$ is the mapping from *sensor* and SN to C_{mndb}.

8) β is the location list of the sensor. (see Sec. II-C-3).

B. THE BLOCKCHAIN DATA STRUCTURE FOR MALICIOUS NODE DETECTION

In order to better describe the C_{mndb} blockchain, this paper proposes a block data structure based on malicious node detection blockchain ($C_{mndb} - BDS$). It is different from the traditional WSNs where it cannot be repeatedly detected. $C_{mndb} - BDS$ records all communication data.

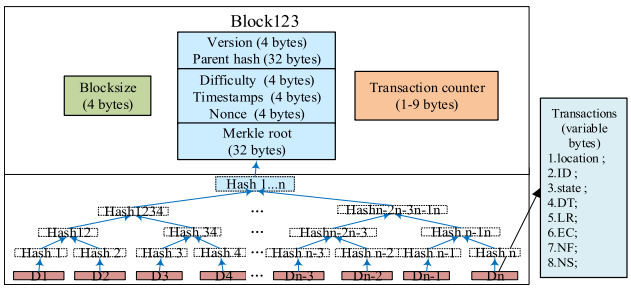

FIGURE 2. An illustration of C_{mndb}-BDS.

As shown in Fig. 2, the data structure is mainly divided into two parts. On the one hand, the block header mainly

contains the hash value of the previous block where the hash value is used to connect the previous block and meet the needs of the integrity of the C_{mndb} blockchain. On the other hand, the block body contains the main information of the wireless sensor node in the block, such as location, ID, $state$, DT, FR, RT, N_S and N_F (see Sec. II-C). This information together with the hash value of the previous block and the random number form the hash value of the block.

Here, D1 - Dn represents each sensor data collected by sensor, Hash1 is the hash pointer of D1 sensor data, Hash12 is the hash pointer of Hash1 + Hash2, so the layer stack is added, and finally the unique Merkle-root is generated.

C_{mndb} not only uses a "block + chain" chain data structure, but also records the information collected by each block in the form of a Merkle Tree formed by a hash pointer. Such a data structure makes it possible to change the hash pointer of the block once the data of any block is modified, thereby ensuring that the data cannot be tampered with. In addition, using the data structure of $C_{mndb} - BDS$, the data is recorded by multiple sensors when the whole network is released to reduce the possibility of malicious manipulation, ensure safety and fairness, and improve the convenience of the detection process.

C. SMART CONTRACT OF THE MODEL

A smart contract is a piece of code that is deployed on a distributed ledger [33] that controls the received outside information. The C_{mndb} blockchain leverages the smart contract platform provided by Decentralized Application (DAPP) to increase its flexibility and operability.

1) MALICIOUS NODE DETECTION METHOD IN SMART CONTRACT

The smart contract of malicious node detection blockchain ($C_{mndb}-SC$) is proposed in this paper that contains the following relationships:

$$C_{mndb}-SC = (bs, sn, sensor, \delta, \eta, C_{mndb} - BDS, QM)$$

Here, the bs is the publisher of $C_{mndb}-SC$; the sn is the aggregation node that the bs authorizes to vote; the δ is the malicious sensor evaluation indicator which has DT, FR and RT; η is the reputation of the node; and $C_{mndb} - BDS$ is the data structure of the C_{mndb}; QM is the WSNs positioning method.

As shown in Fig. 3, the specific steps for using smart contract to detect malicious nodes are as follows:

Step 1. The bs releases smart contract $C_{mndb}-SC$ to the entire network.

Step 2. The bs authorizes the sn to become a voter of the $C_{mndb}-SC$.

Step 3. The sn locates all $sensor$ according to (6) in QM, so that the position and ID of $sensor$ are in one-to-one correspondence, and then a complete NLL is obtained.

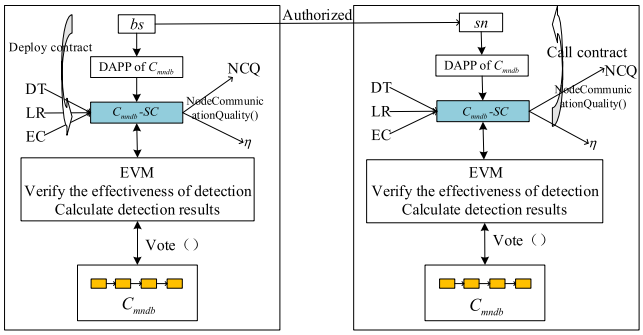

FIGURE 3. An architecture of C_{mndb}.

Step 4. The sn determines the $state$ of the sensor based on the actual situation.

Step 5. Under the working state of the $sensor$ node, the collected sensor information is calculated by (1-3) to obtain the corresponding DT, FR and RT.

Step 6. Send the values of DT, FR and RT to the NodeCommunicationQuality() function, and calculate the NCQ value by (4).

Step 7. Calculate the values of the corresponding N_S and N_F according to the value of NCQ.

Step 8. Calculate the value of η by (5).

Step 9. Finally, based on the obtained η, the sn uses the Vote() to vote on the ID of malicious $sensor$. (The function Vote() may be divided into three parts: Firstly, set an appropriate voting threshold ε according to the actual scene; Then, sn determines the range of the η of all the sensors in the coverage area. If $\eta > \varepsilon$, then the $sensor$ is malicious; finally, the ID of the malicious $sensor$ is cast.)

Step 10. According to the ID of the cast, look at the NLL and find the corresponding sensor location.

2) MALICIOUS NODE DETECTION METHOD IN SMART CONTRACT

Because the evaluation of malicious nodes is subjective and uncertain, in this paper it takes the state of node, processing delay, forwarding rate and response time as the evaluation indicators of malicious nodes to improve the credit degree of nodes. In order to better describe the state of the sensor node in the network environment, it is divided into working state and non-working state. If the sensor nodes are not working, we directly remove from the network; otherwise, collect the following three factors:

a: DELAYED TRANSMISSION (DT) FACTORS

Analyzing the time when nodes forward packets will meet the needs of identifying malicious nodes in WSNs, and improve the effectiveness of network data collection. During a certain time interval, we count the time from the receipt of the complete packet to the end of the packet forwarding, and calculate the proportion of the time in the interval. The calculation

formula is given by

$$DT = \begin{cases} \dfrac{Tsensor\_id}{T_1} * 100\% & 0 < \dfrac{Tsensor\_id}{T_1} < 1 \\ 1 & \dfrac{Tsensor\_id}{T_1} \geq 1 \end{cases} \quad (1)$$

Here, $Tsensor\_id$ is the time from the receipt of the complete packet to the end of the packet forwarding of $sensor\_id$; T_1 is a certain time interval.

b: FORWARDING RATE (FR) FACTOR

To prevent malicious nodes from tampering with forwarding packets, it is necessary to evaluate the integrity of forwarded packets. When the source node sends the data packet, the next hop node is monitored in a certain time whether the data forwarding is performed correctly. The calculation formula is given by

$$FR = \frac{sd}{td} * 100\% \quad (2)$$

Here, sd is the amount of data sent by the node; td is the total amount of data received by the node.

c: RESPONSE TIME (RT) FACTOR

Response time is a very important evaluation factor in some special situations (e.g. disaster, fire detection), which is mainly reflected in the communication delay process. Therefore, in order to identify a malicious node, the response time speed of the node needs to be evaluated. During a certain time interval, we count the time from the start of the request until the receipt of valid data, and calculate the proportion of the time in the interval. The calculation formula is given by

$$RT = \frac{\frac{dbn}{bw} + \frac{pd}{ps} + pt}{T_2} \quad (3)$$

Here, dbn is the number of data bits; bw is the network bandwidth; pd is the propagation distance; ps is the propagation speed; pt is the processing time; and T_2 is a certain time interval.

Therefore, the final node communication quality (NCQ) is given by

$$NCQ = \begin{cases} 0 & state = 0 \\ \gamma * DT + \lambda * (1 - FR) + \sigma * RT & state = 1 \end{cases} \quad (4)$$

Here, γ, λ and σ are the weights of delayed transmission, forwarding rate, and response time respectively. It can be adjusted according to the specific scene to meet the requirements of $\gamma + \lambda + \sigma = 1$.

In this paper, we set a threshold κ that can be adjusted according to the specific scenario. When $NCQ \leq \kappa$, the number of successful communications N_S of the node is increase. Conversely, it increases the number of failed communications N_F of the node.

Finally, calculate the credit of the node based on the distribution [34], N_S and N_F.

$$\eta = \frac{N_S + 1}{N_S + N_F + 2} \quad (5)$$

Similarly, when $\eta \leq \varepsilon$, the sensor is determined to be a malicious node.

3) WSN POSITIONING METHOD IN SMART CONTRACT

Sensor location data is very important for WSNs in certain scenarios. When a disaster occurs (e.g. an earthquake, a forest fire, a natural gas leak), Wireless sensor network-based monitoring does not mean much if it only knows that a serious incident has occurred but does not know its specific location. Therefore, all sensor location information contained in the entire WSNs is stored in the sensor node.

However, because the number of sensors in WSNs is too numerous, it is difficult for sensor network systems to obtain location information of all nodes at an early stage. In this paper, the terminal node in a network is grouped into two, namely anchor node-set (AN) and unknown locations node-set (ULN). Among them, AN represents a special class that knows its own location, and ULN represents a class of nodes that do not know their locations. Every node has a same node location list (NLL) which has node ID and location. The initial list only contains the location information of the anchor node. In order to obtain the location information of all nodes, this paper proposes a quadrilateral measurement (QM) method in [35], which can acquire nodes of unknown locations, and then improve the unified NLL. To meet the needs of quickly monitoring the state and behavior of the sensor in real time, so as to quickly obtaining their location. The above method can improve NLL written to the smart contract block chain, the detailed flowchart shown in Fig. 4:

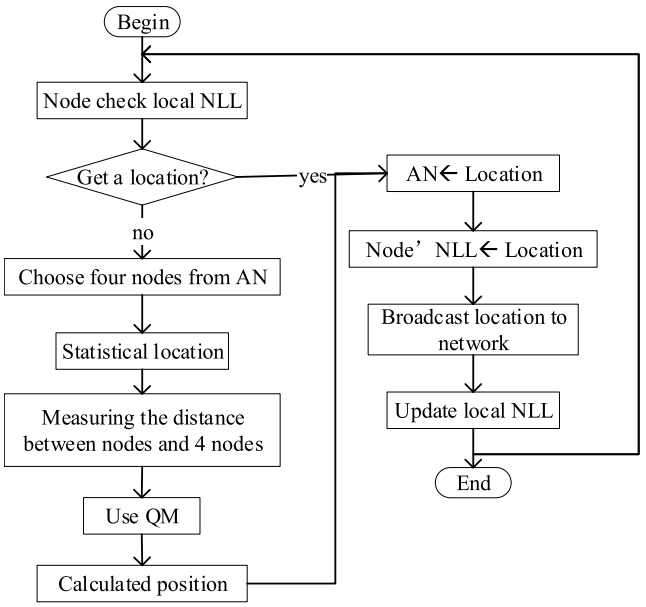

FIGURE 4. A flow chart of WSNs positioning in smart contract.

The QM method mentioned above can be described by the following simple example: Hypothesis $A(x_1, y_1, z_1)$, $B(x_2, y_2, z_2)$, $C(x_3, y_3, z_3)$,$D(x_4, y_4, z_4) \in$ AN, $U(x, y, z) \in$ ULN, d_1, d_2, d_3, d_4 is the distance from U to A, B, C, D,

the position data calculation formula of U is given by

$$\begin{cases} (x-x_1)^2+(y-y_1)^2+(z-z_1)^2=d_1^2 \\ (x-x_2)^2+(y-y_2)^2+(z-z_2)^2=d_2^2 \\ (x-x_3)^2+(y-y_3)^2+(z-z_3)^2=d_3^2 \\ (x-x_4)^2+(y-y_4)^2+(z-z_4)^2=d_4^2 \end{cases} \tag{6}$$

III. SIMULATION

To validate the performance of our proposed model, we conduct the simulation experiment on the BTM model. Firstly, we present the network environment required to build BTM in Sec. III-A. Then, the simulation results are analyzed in Sec. III-B. Finally, Sec. III-C analyze the performance of the model.

A. MODEL BUDLING

In order to verify the effectiveness of the method, this paper uses Truffle as a smart contract development framework and build tool. A contract abstract interface that can directly operate contract functions in JavaScript through web3.js. Ganache-cli is a private-chain local simulation tool that mines faster than Geth. Npm is built as a project in development, including management and staging servers, etc. Solidity is the language of smart contract development. We consider running Truffle and Ganache-cli on the sensor node. In addition, in order to better simulate the wireless sensor network environment, we conduct simulations by using OPNET network simulation software to collect and analyze the key parameters.

The contents of the simulation are as follows: the simulation scenario is set to $1000m \times 1000m \times 500m$ cube disaster mountainous area where 50 nodes are randomly deployed (5 *sn* nodes, 45 *sensor* nodes). Assuming that the *sn* node is the legitimate node, there are malicious nodes in the *sensor* node. In order to locate the sensor node accurately, the NLL is obtained by using QM to locate the *sensor* node in three-dimensional space. Smart contract that receives the key parameters of the *sensor* node, such as *state*, DT, FR, RT, N_S and N_F, calculates the NCQ and η. All *sn* node can only vote for a malicious node based on η for the *sensor* node within the scope of their own communication. In addition, all *sn* nodes comply with the content specified in the smart contract and records it on the blockchain.

B. SENSOR STATISTICS EXPERIMENTAL RESULTS AND ANALYSIS

In this section, we verify the feasibility of the BTM model from two aspects.

1) SENSOR 3D SPATIAL LOCATION AND CLUSTERING

To understand the state of the *sensor* node in real time and avoid the failure to react in time when a disaster occurs, it is necessary to position the *sensor*. As shown in Fig. 5, it is an effect diagram of positioning processing of 45 *sensor* nodes according to QM. The black spot is the anchor node, ie. the *sn* node. The green circle is the unknown node, ie. the *sensor* node. And the red star is the node after positioning

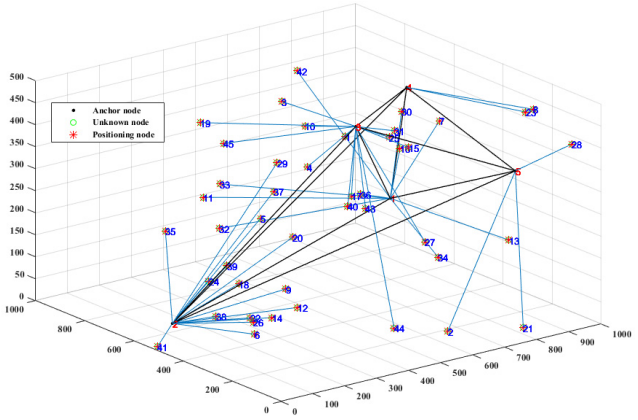

FIGURE 5. A schematic diagram of the location of node.

TABLE 1. The contract addresses for *sn* node.

sn_id	Contract addresses
sn_1	0x692...77b3a
sn_2	0xbbf...732db
sn_3	0x0dc...97caf
sn_4	0x5e7...26e9f
sn_5	0x089...659fb

TABLE 2. The *sensor_id* and numbers of *sn* node.

sn_id	*sensor_id*	*number*
sn_1	1, 7, 11, 13, 16, 27, 30, 32, 33, 42	10
sn_2	5, 6, 9, 12, 14, 18, 20, 22, 24, 26, 29, 35, 37, 38, 39, 41	16
sn_3	3, 4, 10, 15, 17, 19, 25, 31, 36, 40, 43, 45,	14
sn_4	8, 23	2
sn_5	2, 21, 28	3

with QM. From this figure we can get the accuracy of QM positioning. Moreover, to better manage the *sensor* node in the communication range of each *sn* node, we perform clustering processing on all *sensor* nodes. Table 1 shows the contract addresses for each *sn* node in a smart contract. Table 2 shows the *sensor_id* and numbers covered by each *sn* node.

2) SENSOR STATISTICS

Since we want to detect malicious *sensor* nodes better, we consider letting the *sn* node collect some parameters about the *sensor*, such as *state*, DT, FR, RT, N_S and N_F. However, in order to ensure the fairness of the detection, Table 3 only shows the information of the *sensor* node in the communication range of the *sn* node. Here state $= 0$ means the *sensor* is not working, state $= 1$ means the sensor is working, "$--$" means no information.

The specific detecting process as follows. First of all, *sn* node puts some sensor key parameters into $C_{mndb} - SC$, as shown in Table 4. Then, we calculate the key parameters according to (1-5) in $C_{mndb} - SC$, and get the corresponding

TABLE 3. ***Sensor*** **data parameters.**

sn_id	*sensor_id*	Key parameters						Corresponding parameters	
		State	DT/sec	RT/sec	TR	N_S	N_F	*NCQ*	η
sn_1	1	1	0.001443	0.039826	0.284187883	73	25	0.3701424585	0.74257
	7	1	0.000622	0.016097	0.343461535	82	61	0.3332227325	0.57534
	.....				.....				
	42	1	0.000335	0	0.145425075	86	63	0.4273544625	0.57895
sn_2	5	1	0.000834	0.042531	0.307827531	3	23	0.3590123345	0.17241
	6	1	0.000257	0	0.340211504	14	23	0.3299456480	0.40000
	.....				.....				
	41	0	--	--	--	--	--	0	0
.....	.....				.....				
sn_5	2	1	0.000647	0.029870	0.344576401	69	48	0.3368021995	0.59167
	21	0	--	--	--	--	--	0	0
	28	1	0.000540	0.041537	0.435745306	63	68	0.2946964470	0.47761

TABLE 4. **Input data of** ***sensor*** **in the smart contract.**

states	0x1 Transaction mined and execution succeed
transaction hash	0x2381719bdda531a06284e062d8e14626db072cd80d9fd2e647f40c2079341b80
from	0xca35b7d915458ef540ade6068dfe2f44e8fa733c
to	ClusterHeads1.add (uint 256, uint 256, uint 256, uint 256, uint 256, uint 256, uint 256, uint256) 0x692a70d2e424a56dac6c27aa97d1a86395877b3a
gas	3000000 gas
transaction cost	179758 gas
execution cost	156438 gas
hash	0x2381719bdda531a06284e062d8e14626db072cd80d9fd2e647f40c2079341b80
input	0x75f...00001
decoded input	"uint256_DT": "1443000", "uint256_RT": "39826000", "uint256_FR": "284187883", "uint256_NCQ": "0", "uint256_v": "0", "uint256_NS": "73", "uint256_NF": "25", "uint256_voted": "0", "uint256_state": "1"
decoded output	{}
logs	[]
value	0 wei

NCQ and η as shown in Table 5. Table 6 shows the *NCQ* and η in the communication range in the smart contract. As shown in Fig. 6, it shows all the η of *sensor* nodes. (Since the solidity language only supports integers, when the smart contract input data is called, the collected data is expanded accordingly. There is a slight error between the calculation result and the direct calculation data.)

C. MODEL PERFORMANCE ANALYSIS

1) MODEL SECURITY ANALYSIS

In this paper, it uses the distributed chained data structure so that the sensor node information recorded in each block on the chain can be traced back from the previous block, and affect the sensor node information of the next block record directly. If the node wants to tamper with past sensor data, the local data recorded by the sensor node must be changed by 51%. Such a large complexity process data can not be tampered. In addition, the model uses public key cryptosystem, merkle tree, hash and ECDSA signature algorithm to ensure data integrity and security.

2) MODEL RELIABILITY ANALYSIS

This paper uses the decentralized storage feature of Blockchain technology to spread the workload to the network, record all sensor node data in a distributed way, and form a chain structure in a chronological manner. When one

TABLE 5. Output data of *sensor* in the smart contract.

transaction hash	0x5449525ed30c89ef576173e270793fd407d3c2852dfc0427a781193e80b987d5
from	0xca35b7d915458ef540ade6068dfe2f44e8fa733c
to	ClusterHeads1.Vote() 0x692a70d2e424a56dac6c27aa97d1a86395877b3a
transaction cost	82712 gas (Cost only applies when called by a contract)
execution cost	61440 gas (Cost only applies when called by a contract)
hash	0x5449525ed30c89ef576173e270793fd407d3c2852dfc0427a781193e80b987d5
input	0x6bf...52ffa
decoded input	{}
decoded output	"0": "uint256[]: 3701424585", "1": "uint256[]:74", "2": "uint256[]:0"
logs	[]

TABLE 6. *NCQ* and η in the SN_1 communication range in the smart contract.

transaction hash	0x5597dc0fcf3c90dc161d371eae7e45fba040e32ed15b311f71fe8dffee5292093
from	0xca35b7d915458ef540ade6068dfe2f44e8fa733c
to	ClusterHeads1.Vote() 0x692a70d2e424a56dac6c27aa97d1a86395877b3a
transaction cost	605804 gas (Cost only applies when called by a contract)
execution cost	584532 gas (Cost only applies when called by a contract)
hash	0x5597dc0fcf3c90dc161d371eae7e45fba040e32ed15b311f71fe8dffee5292093
input	0x6bf...52ffa
decoded input	{}
decoded output	"0": "uint256[]: 3701424585, 3332227325, 0, 3436375845, 2527449270, 3640864125, 2893822630, 3714172605, 2541621200, 4273544625", "1": "uint256[]:74, 57, 0, 84, 81, 19, 55, 61, 64, 57", "2": "uint256[]:0, 0, 1, 0, 0, 1, 0, 0, 0, 0"
logs	[]

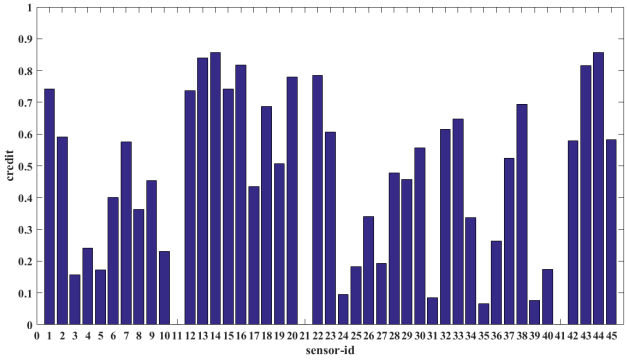

FIGURE 6. The credit of *sensor*.

node fails, other nodes will not be affected, avoiding single point failure and enhancing the stability and reliability of the system. At the same time, the smart contract is added to the blockchain by means of digitized code. When the contract trigger condition is met, the code of the smart contract will be automatically started. Once the malicious sensor node is authenticated, it will be recorded in the blockchain network in time so that it can prevent the loss of information asymmetry caused by time difference to the whole wireless sensor network.

3) TRACEABILITY ANALYSIS

In this model, it uses distributed data blocks linked in time-stamped order, and all sensor nodes are stored permanently. The information of the *sensor* node is bound to each data record in the link. The whole process is transparent and traceable, realizing dynamic update of data records in real time, and ensuring traceability of original data. It makes up for the problems of opacity, traceability and injustice in the traditional WSNs process of detecting malicious nodes. When malicious behavior occurs in WSNs, the identity of the attacking node can be identified by traceability of the model, and corresponding measures can be taken to prevent the further impact of such malicious nodes on the network.

IV. CONCLUSION

Nowadays, the malicious node detection in WSNs mostly adopts the way of one-time centralized decision-making. According to this method, the original data cannot be traced back, the detection process is difficult to reproduce and check, and the problems of error and false positives are difficult to avoid. In this paper, through 3D space it is realized by using block chain intelligent contract and WSNs quadrilateral measurement for localization of the detection of malicious

nodes in, and the consensus results of voting are recorded in the blockchain distributed. The simulation results show that the model can effectively detect malicious nodes in WSNs and ensure the traceability of the detection process.

At present, the model is still in the theoretical research stage, and there is still room for further research and improvement. In the next step, we will improve the model from the following two aspects:

1) Get more engineering and experimental data, and fit the NCQ and η in the formal model BTM to meet the practical needs.
2) The consensus method in the model is the traditional POW workload proof method, which requires relatively large computational power and high energy consumption, so it is not especially suitable for the running environment of wireless sensor networks. The next step will improve and experiment the combination of PoS and other methods.

ACKNOWLEDGMENT

The authors would like to thank the reviewers for their critical and constructive comments and suggestions.

STATEMENT

The author's confirmation that the mentioned received funding in the "Acknowledgment" section did not lead to any conflict of interests.

The simulation data used to support the findings of this study have not been made available because it is a series of data randomly generated based on sensor indicators.

REFERENCES

[1] T. Qiu, K. Zheng, M. Han, C. L. P. Chen, and M. Xu, "A data-emergency-aware scheduling scheme for Internet of Things in smart cities," *IEEE Trans. Ind. Informat.*, vol. 14, no. 5, pp. 2042–2051, May 2018. doi: 10.1109/TII.2017.2763971.

[2] J. N. Al-Karaki and A. E. Kamal, "Routing techniques in wireless sensor networks: A survey," *IEEE Wireless Commun.*, vol. 11, no. 6, pp. 6–28, Dec. 2004. doi: 10.1109/MWC.2004.1368893.

[3] S. N. Pakzad, G. L. Fenves, S. Kim, and D. E. Culler, "Design and implementation of scalable wireless sensor network for structural monitoring," *ASCE J. Infrastruct. Syst.*, vol. 14, no. 1, pp. 89–101, Mar. 2008. doi: 10.1061/(ASCE)1076-0342.

[4] H. Alemdar and C. Ersoy, "Wireless sensor networks for healthcare: A survey," *Comput. Netw.*, vol. 54, no. 15, pp. 2688–2710, Oct. 2010. doi: 10.1016/j.comnet.2010.05.003.

[5] K. Akkaya and M. Younis, "A survey on routing protocols for wireless sensor networks," *Ad Hoc Netw.*, vol. 3, no. 3, pp. 325–349, 2005. doi: 10.1016/j.adhoc.2003.09.010.

[6] W. Zhang and G. Cao, "Group rekeying for filtering false data in sensor networks: A predistribution and local collaboration-based approach," in *Proc. IEEE 24th Annu. Joint Conf. IEEE Comput. Commun. Soc.*, Mar. 2005, pp. 503–514. doi: 10.1109/INFCOM.2005.1497918.

[7] A. D. Wood and J. A. Stankovic, "Denial of service in sensor networks," *Computer.*, vol. 35, no. 10, pp. 54–62, Oct. 2002. doi: 10.1109/MC.2002.1039518.

[8] F. Li and J. Wu, "A probabilistic voting-based filtering scheme in wireless sensor networks," in *Proc. 6th Int. Conf. Wireless Commun. Mobile Comput. (IWCMC)*, New York, NY, USA, 2006, pp. 27–32. doi: 10.1145/1143549.1143557.

[9] T. Qiu, R. Qiao, and D. O. Wu, "EABS: An event-aware backpressure scheduling scheme for emergency Internet of Things," *IEEE Trans. Mobile Comput.*, vol. 17, no. 1, pp. 72–84, Jan. 2018. doi: 10.1109/TMC.2017.2702670.

[10] F. Zawaideh, M. Salamah, and H. Al-Bahadili, "A fair trust-based malicious node detection and isolation scheme for WSNs," in *Proc. 2nd IT-DREPS*, Amman, Jordan, Dec. 2017, pp. 1–6. doi: 10.1109/IT-DREPS.2017.8277813.

[11] L. G. Zeng, L. Y. Yuan, and H. Wang, "Detecting WSN node misbehavior based on the trust mechanism," *J. Zhejiang Normal Univ. (Nat. Sci.)*., vol. 41, no. 1, pp. 39–43, 2018. doi: 10.16218/j.issn.1001-5051.2018.01.007.

[12] Y. X. Su, X. F. Gao, and Y. Lu, "Credibility based WSN trust model," *Electron. Opt. Control*, vol. 25, no. 3, pp. 32–36, 2018. doi: 10.3969/j.issn.1671-637X.2018.03.008.

[13] V. R. Prabha and P. Latha, "Fuzzy trust protocol for malicious node detection in wireless sensor networks," in *Wireless Pers. Commun.*, vol. 94, no. 4, pp. 2549–2559, 2017. doi: 10.1007/s11277-016-3666-1.

[14] W. Zhang, S. Zhu, J. Tang, and N. Xiong, "A novel trust management scheme based on Dempster–Shafer evidence theory for malicious nodes detection in wireless sensor networks," *J. Supercomput.*, vol. 74, no. 4, pp. 1779–1801 2018. doi: 10.1007/s11227-017-2150-3.

[15] G. Yang, G. S. Yin, W. Yang, and D.-M. Zuo, "A reputation-based model for malicious node detection in WSNs," *J. Harbin Inst. Technol.*, vol. 41, no. 10, pp. 158–162, 2009. doi: 0367-6234(2009)10-0158-05.

[16] S. Das and A. Das, "An algorithm to detect malicious nodes in wireless sensor network using enhanced LEACH protocol," in *Proc. Int. Conf. Adv. Comput. Eng. Appl.*, Ghaziabad, India, Mar. 2015, pp. 875–881. doi: 10.1109/ICACEA.2015.7164828.

[17] Z. Chen, R. Zhang, L. Ju, and W. Wang, "Multivalued trust routing based on topology level for wireless sensor networks," in *Proc. 12th IEEE Int. Conf. Trust, Secur. Privacy Comput. Commun.*, Melbourne, VIC, Australia, Jul. 2013, pp. 1516–1521. doi: 10.1109/TrustCom.2013.185.

[18] W. R. Pires, T. H. de Paula Figueiredo, H. C. Wong, and A. A. F. Loureiro, "Malicious node detection in wireless sensor networks," in *Proc. 18th Int. Parallel Distrib. Process. Symp.*, Santa Fe, NM, USA, Apr. 2004, p. 24. doi: 10.1109/IPDPS.2004.1302934.

[19] E. M. de Lima Pinto, R. Lachowski, M. E. Pellenz, M. C. Penna, and R. D. Souza, "A machine learning approach for detecting spoofing attacks in wireless sensor networks," in *Proc. 32nd AINA*, Krakow, Poland, May 2018, pp. 752–758. doi: 10.1109/AINA.2018.00113.

[20] S. Althunibat, A. Antonopoulos, E. Kartsakli, F. Granelli, and C. Verikoukis, "Countering intelligent-dependent malicious nodes in target detection wireless sensor networks," *IEEE Sensors J.*, vol. 16, no. 23, pp. 8627–8639, Dec. 2016. doi: 10.1109/JSEN.2016.2606759.

[21] B. Uddin, A. Imran, and M. A. Rahman, "Detection and locating the point of fault in distribution side of power system using WSN technology," in *Proc. 4th ICAEE*, Dhaka, Bangladesh, Sep. 2017, pp. 570–574. doi: 10.1109/ICAEE.2017.8255421.

[22] S. Nakamoto. (2008). *Bitcoin: A Peer-to-Peer Electronic Cash System*. [Online]. Available: https://nakamotoinstitute.org/bitcoin

[23] M. Swan, *Blockchain: Blueprint for a New Economy*. Sebastopol, CA, USA: O'Reilly Media, 2015.

[24] K. Christidis and M. Devetsikiotis, "Blockchains and smart contracts for the Internet of Things," *IEEE Access*, vol. 4, pp. 2292–2303, 2016. doi: 10.1109/ACCESS.2016.2566339.

[25] D. Magazzeni, P. McBurney, and W. Nash, "Validation and verification of smart contracts: A research agenda," *Computer*, vol. 50, no. 9, pp. 50–57, 2017. doi: 10.1109/MC.2017.3571045.

[26] J. Ellul and G. J. Pace, "AlkylVM: A virtual machine for smart contract blockchain connected Internet of Things," in *Proc. 9th NTMS*, Paris, France, Feb. 2018, pp. 1–4. doi: 10.1109/NTMS.2018.8328732.

[27] M. N. Islam and S. Kundu, "Poster abstract: Preserving IoT privacy in sharing economy via smart contract," in *Proc. IoTDI*, Orlando, FL, USA, Apr. 2018, pp. 296–297. doi: 10.1109/IoTDI.2018.00047.

[28] E. S. Kang, S. J. Pee, J. G. Song, and J. W. Jang, "A blockchain-based energy trading platform for smart homes in a microgrid," in *Proc. 3rd ICCCS*, Nagoya, Japan, Apr. 2018, pp. 472–476. doi: 10.1109/CCOMS.2018.8463317.

[29] J. Pan, J. Wang, A. Hester, I. Alqerm, Y. Liu, and Y. Zhao, "EdgeChain: An edge-IoT framework and prototype based on blockchain and smart contracts," *IEEE Internet Things J.*, to be published. doi: 10.1109/JIOT.2018.2878154.

[30] T. Qiu, X. Liu, K. Li, Q. Hu, A. K. Sangaiah, and N. Chen, "Community-aware data propagation with small world feature for Internet of vehicles," *IEEE Commun. Mag.*, vol. 56, no. 1, pp. 86–91, Jan. 2018. doi: 10.1109/MCOM.2018.1700511.

[31] Y. Zhang, S. Kasahara, Y. Shen, X. Jiang, and J. Wan, "Smart contract-based access control for the Internet of Things," *IEEE Internet Things J.*, to be published. doi: 10.1109/JIOT.2018.2847705.
[32] L. Xu, R. Collier, and G. M. P. O'Hare, "A survey of clustering techniques in WSNs and consideration of the challenges of applying such to 5G IoT scenarios," *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1229–1249, Oct. 2017. doi: 10.1109/JIOT.2017.2726014.
[33] J. Kishigami, S. Fujimura, H. Watanabe, A. Nakadaira, and A. Akutsu, "The blockchain-based digital content distribution system," in *Proc. IEEE 5th Int. Conf. Big Data Cloud Comput.*, Dalian, China, Aug. 2015, pp. 187–190. doi: 10.1109/BDCloud.2015.60.
[34] M. Momani, M. Takruri, and R. Al-Hmouz, "Risk assessment algorithm in wireless sensor networks using beta distribution," *Int. J. Comput. Netw. Commun.*, vol. 6, no. 5, pp. 157–166, 2014. doi: 10.5121/ijcnc.2014.6511.
[35] M. B. Nirmala, A. S. Manjunath, and M. Rajani, "Secure and efficient voting based localization scheme for wireless sensor networks," *BVICA M's Int. J. Inf. Technol.*, vol. 6, no. 2, pp. 750–756, 2014.

WEI SHE received the B.S. degree in control engineering from the Air Defence Academy of PLA, in 2000, the M.S. degree in software engineering from Hunan University, in 2008, and the Ph.D. degree in computer software and theory from Zhengzhou University, China, in 2013, where he is currently an Associate Professor with the Software College. His research interests include information security, the energy Internet, and the Internet healthcare.

QI LIU received the B.S. degree from Zhengzhou University, China, in 2016, where she is currently a graduate student. Her research interests include information security and Petri net theory.

ZHAO TIAN received the B.S. degree in information and computing science from Huazhong Agricultural University, in 2008, the M.S. degree in computer software and theory from Zhengzhou University, in 2011, and the Ph.D. degree in safety technology and engineering from Beijing Jiaotong University, in 2016. He is currently a Lecturer with the Software College, Zhengzhou University. His research interests include information security, artificial intelligence, and intelligent transportation.

JIAN-SEN CHEN received the B.S. degree from Zhengzhou University, China, in 2017, where he is currently a graduate student. His research interests include information security and blockchain.

BO WANG (M'15) received the B.S. degree in electronic and information engineering, the M.S. degree, and the Ph.D. degree in signal and information processing from the Dalian University of Technology, China, in 2003, 2005, and 2010, respectively. He is currently a Visiting Scholar with The State University of New York at Buffalo. His current research interests include multimedia processing and security. Also, he is a member of the ACM and the Session Chair of Mlicom 2017.

WEI LIU received the B.S. and M.S. degrees in information engineering from Zhengzhou University, China, in 2003 and 2008, respectively, and the Ph.D. degree from Tohoku University, Japan, in 2013. He is currently an Associate Professor with the Software College, Zhengzhou University. His research interests include information security, wireless mesh networks, and the Internet healthcare.