Wireless Communications and Mobile Computing

Blockchain and Edge Intelligence in the Internet of Things

Lead Guest Editor: Celimuge Wu
Guest Editors: Qingqi Pei and Maode Ma

Blockchain and Edge Intelligence in the Internet of Things

Wireless Communications and Mobile Computing

Blockchain and Edge Intelligence in the Internet of Things

Lead Guest Editor: Celimuge Wu
Guest Editors: Qingqi Pei and Maode Ma

Chief Editor

Associate Editors

Academic Editors

Contents

Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 6852822, 10 pages
https://doi.org/10.1155/2022/6852822

Research Article

The PSL MAC Protocol for Accumulated Data Processing in the Energy-Harvesting Wireless Sensor Network

Ruihong Wang, Wuyungerile Li, Fei Gao, and Taofeng Jiao

Inner Mongolia University, China

Correspondence should be addressed to Wuyungerile Li; gerile@imu.edu.cn

Received 11 March 2022; Revised 19 May 2022; Accepted 17 June 2022; Published 6 July 2022

Academic Editor: Celimuge Wu

In the energy-harvesting wireless sensor network (EH-WSN), the actual energy-harvesting rate of each node varies due to the difference in node deployment or a sudden change in the environment. Therefore, when a node with a low energy-harvesting rate is used as a relay node, its energy is depleted at an accelerated rate. This reduces connectivity and network life, thus causing data accumulation on the node and the neighboring nodes. This study proposes a medium access control (MAC) protocol to avoid data accumulation in nodes; the proposed algorithm comprises the following: (1) a data transmission mechanism based on probability control for high channel utilization and (2) an accumulation data processing mechanism (power control, same layer transmission, and "layer-down" processing, PSL) to circumvent data accumulation. The simulation results show that PSL improves the network throughput and reduces the packet loss rate, while effectively solving the data accumulation problem.

1. Introduction

Wireless sensor networks (WSNs) have been developed rapidly and used widely owing to their characteristics such as no infrastructure requirement, low cost, small size, and ease of deployment. However, the limited battery power and single mode of power supply restrict their further development. Furthermore, replacing batteries is not an economic, safe, or environment-friendly approach, especially in harsh environments [1, 2]. For the aforementioned reasons, energy-harvesting WSNs (EH-WSNs) have gained popularity as a viable alternative. EH-WSNs are no longer limited to dry battery energy; they can accrue clean energy from the environment to power the nodes. EH-WSNs circumvent the shortcomings of traditional WSNs with regard to energy storage and supply. With proper control of the collected energy, EH-WSNs can work indefinitely [3, 4]. However, due to environmental and climatic variations, the rate of harvested energy at the nodes of EH-WSNs, which switch between the working and sleeping modes, is inconsistent [5].

Conventional studies on WSNs have focused on reducing power consumption, extending network lifetime and accomplishing a larger number of tasks with minimum energy consumption [6, 7]. However, energy-saving and network lifetime are no longer the primary focal points of studies pertaining to EH-WSNs.

In EH-WSNs, nodes are always deployed in outdoor environments, which are often harsh and uninhabited areas. In addition, solar irradiation varies widely with time, weather, and season and is greatly influenced by the placement angle of the solar collector. Therefore, in practical applications, solar energy collection is largely subjected to environment factors such as time, season, wind, sand, rain, snow, light blocking, and damage resulting from animals and plants. The aforementioned factors can potentially lead to insufficient node energy supply or intermittent energy supplements, thus resulting in unstable node energy. Consequently, the stability and reliability of network connectivity are affected, which results in the formation of isolated nodes that can accumulate a large amount of historical data. To address the aforementioned roadblocks, we propose a stacked data processing algorithm in a harsh environment. To avoid data conflict, a data transmission mechanism based on probability control is proposed. In addition, after the data

accumulation occurs in the node, the method of increasing power, same layer forwarding, and "layer-down" processing is considered to deal with the problem of data accumulation.

The rest of this paper is organized as follows: In Section 2, the existing medium access control (MAC) protocols for EH-WSNs are introduced, along with some power control algorithms and the characteristics of environmental energy. Section 3 proposes the probabilistic transmission mechanism to avoid signal conflict and proposes an accumulation data processing mechanism to solve the accumulation data problem, while the simulation results are presented in Section 4. A summary of the study and its achievements is presented in Section 5.

2. Related Work

Extensive studies and the widespread utilization of new clean energy sources have led to the advancement of the environmental energy collection technology, with steady improvements in energy collection efficiency. Kansal [8], Park Chou [9], Zhu [10], and other scholars have established mathematical models to demonstrate the collection of environmental energy for WSNs. In 2005, Raghunathan proposed the concept of EH-WSNs [11], with the idea of collecting energy from the environment for sensor nodes by using the traditional MicaZ wireless sensor node. Kansal [11] set up a model for energy collection, which optimized the work cycle and task scheduling of the nodes [12], as well as the energy utilization and service life of the whole system [8]. AH-MAC (adaptive hierarchical MAC protocol) [13] is an adaptive MAC protocol initiated by the sink node. It optimizes the duty cycle to prolong the sleep mode of the network, thereby reducing the energy consumption and improving the throughput.

On-demand MAC protocol (ODMAC) [14] can be customized according to the requirements of specific applications. The protocol supports nodes with different duty cycles, which enables each node to maintain its energy consumption at the same level as the energy collected, and each node can operate as close to energy neutral operation as possible. The ID polling MAC protocol [15] is initiated by the sink node and polled by the node number (ID). In EH-WSNs, energy collection and conversion of a node are often affected by time, space, climate, and other factors. Moreover, static transmission power, transmission range, link quality, and topology control have no significant effect in optimizing the energy collection network in harsh environments. Therefore, real-time transmission power control is proposed in EH-WSNs. Xu et al. proposed a dynamically adjusted duty cycle-optimized congestion scheme based on real-time queue length (ADCOC) [16], which avoids network congestion, minimizes system latency, and improves energy efficiency. Tang conducted research to analyze blind spot localization in road traffic WSNs [17] and proposed solutions and measures to reduce monitoring results. In QAEE-MAC [18], the receiver node selects the sender node according to the sender's data priority, and the receiver node also adjusts its wake-up period based on the energy state. Therefore, the energy consumption of the receiving node can be minimized.

3. Accumulated Data Processing Algorithm in the Energy-Harvesting Wireless Sensor Network

In EH-WSNs, the actual energy-harvesting rate of each node in the network is different due to differences in the deployment environments of the nodes or the sudden change in the environment around the nodes. Therefore, when a node with a low energy-harvesting rate is used as a relay node, its energy will be depleted at an accelerated rate. This eventually decreases the network's connectivity and network life. Simultaneously, data accumulation occurs on the node and the neighboring nodes. Failure to handle the accumulated data accurately will cause a high packet loss rate in the network, thereby reducing network throughput. This paper proposes a MAC protocol to deal with the problem of data accumulation; the protocol comprises two proposals: (1) a probabilistic transmission mechanism to solve the problem of signal conflict that may occur between the sender and the receiver and (2) an accumulation data processing mechanism of "power control," "same layer forwarding," and "layer-down processing" (PSL) to address the data accumulation problem caused by differences in the energy-harvesting rate of nodes.

3.1. Network Model. This paper adopts a hierarchical convergent data harvesting model, where a flat structure is centered on a sink node, and sensor nodes in the network carry solar panels to power themselves. The sink node collects and processes the perception data of all the nodes in the network. Except for the outermost node, all other nodes can be used as relay nodes. The outermost node only senses data and does not perform data-forwarding tasks. The sink node is assumed to have a continuous power supply, as the sensor nodes use solar modules to harvest environmental energy to perform sensing and relay tasks. The hierarchical convergence EH-WSN topology is shown in Figure 1.

As shown in Figure 1, the outer node sends the perceived data to the inner node, whereas the inner node merges the data received from the outer node with the data it senses and sends them to the next-hop neighboring node. In this manner, data are transmitted to the sink node. In the existing EH-WSN, due to the characteristics of the node itself and the characteristics of environmental energy, the sensor node cannot work continually. A sensor node has the following two states:

(1) Sleeping state: In this state, the wireless transceiver and the microprocessor stop working, thus reducing energy consumption to a minimum

(2) Working state: The node is in a working state after being fully charged. The working state can be further divided into the random back-off state, carrier monitoring state, data sending state, and data receiving state

There are two types of sending states: (1) sending a data request packet and (2) sending the data that the node perceives or receives from an outer node.

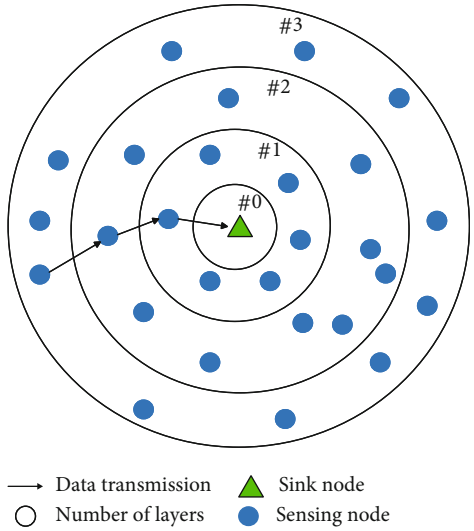


Figure 1: Network topology.

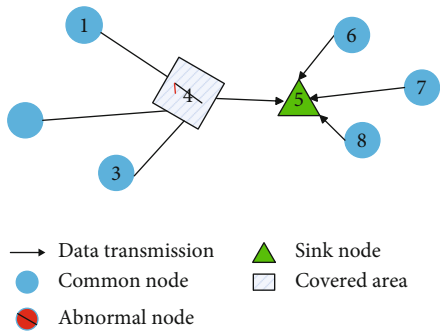


Figure 2: Data accumulation due to insufficient energy in the relay node.

Similarly, the receiving state is divided into two types: (1) waiting for the request response packet and (2) waiting for the data packet sent by the outer node.

3.2. Cause and Solution to the Regional Data Accumulation Problem. Solar panels are the most widely used energy-harvesting equipment; however, the energy-harvesting rate of solar panels is greatly affected by factors such as time, weather conditions, and node locations. When the solar panel is inclined or covered, it has a significant impact on the rate of energy harvesting. The energy-harvesting rate of a single sensor node may affect the network performance of the entire EH-WSN to a certain extent, especially when these are relay nodes. This is because it will affect the connectivity of the network and cause data accumulation problems on some nodes. Figure 2 illustrates the situation when the relay node has insufficient energy.

The figure shows five nodes in a given network. Nodes 1, 2, and 3 are common sensing nodes; node 4 has both sensing and forwarding functions; and node 5 is the sink node. In general, the energy-harvesting efficiency of all nodes in similar locations should be at a similar level. However, node 4 cannot continue to harvest energy due to being covered, thus causing node 4 to become an abnormal node. Consequently, nodes 1, 2, and 3 that forward data through node 4 will become isolated nodes in the network and cannot transmit data, leading to data accumulation.

3.3. Probability Control Data Transmission Mechanism. Regardless of the type of harvesting equipment employed for EH-WSNs, the energy-harvesting rate of each node is not the same, which renders it impossible for EH-WSNs to set all scheduling information during initialization. Therefore, this study proposes a data transmission mechanism based on the probability control including collision avoidance and data probability transmission.

(1) Collision avoidance: In wireless communication, a sending node sends data packets to a receiving node. The data transmission process will not commence if the transmission channel is busy. Hence, a channel allocation mechanism suited to our defined network model is needed. It was designed as follows: When a node in the network is activated from the sleep mode, the node will randomly select a back-off value in the random back-off window as the random back-off time. Thereafter, the node detects the channel within the random back-off time and assumes a receiving state after securing the channel. When the node receives data from an inner node, it immediately shifts from the receiving state to the sending state and transmits the received data. After sending the data, the node switches to the carrier monitoring state until the channel is idle and then initiates a new receiving state, where the aforementioned process is repeated

(2) Data probability transmission: This study proposes a MAC protocol initiated by a receiving node. Data conflicts occur at the receiving node when a receiving node receives data simultaneously from two or more sending nodes. This study attempts to solve this issue through sending data probability. The process is outlined as follows: After the sending node responds to the data request packet from the receiving node, it waits for the response packet from the receiving node. The response packet of the receiving node includes a value n, which is received by the receiving node within the waiting time after sending the data request packet. The response packet of the receiving node includes a value n, where n is the total number of request response packets received by the receiving node within the waiting time after sending the data request packet. After receiving the value, the sending node takes the reciprocal n, generates a random number p between 0 and 1, and compares p with 1/n. When p is >1/n, the energy state is assessed. If the energy is sufficient, the random back-off time is reselected to continue the predefined transmission process. Otherwise, it returns to the sleep state for charging operation. If the node

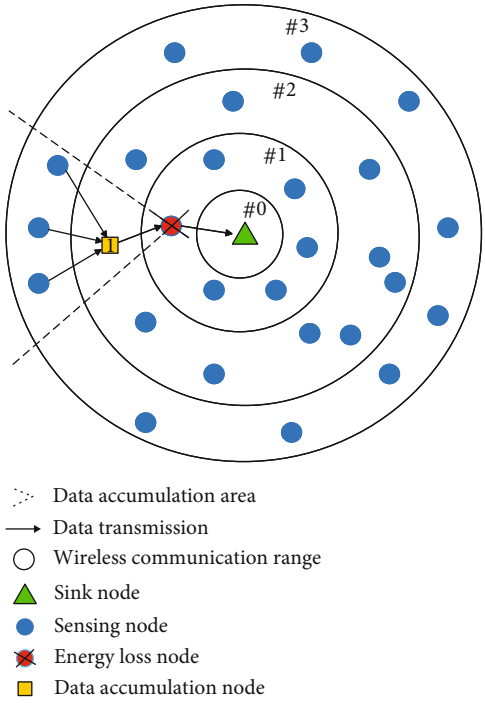


Figure 3: The case of data accumulation.

receives a data packet within the random back-off time, it stops receiving the data packet and switches to the carrier monitoring state until the end of the packet. When the channel is idle again and is still in the random back-off time, the node returns to the receiving state and repeats the appeal process until the end of the random back-off time or until a data request packet is received from the inner node

3.4. PSL-Stacked Data Processing Mechanism. The sensor nodes are deployed in the simulation area by using methods such as spreading. In EH-WSN, when the solar panel of a node is tilted, the energy-harvesting rate of the node cannot reach the average level of the network energy-harvesting rate. At this time, all nodes that use only this node as the relay node cannot transmit data to the sink node, resulting in regional data accumulation. The phenomenon of regional data accumulation is presented in Figure 3.

As shown in Figure 3, when a node has an energy loss phenomenon, the data of all nodes in the data accumulation area cannot be transmitted to the sink node, which may lead to data loss in some areas. The data in the entire accumulating area accumulate at accumulating node 1. When the buffer area of node 1 is full, it can neither sense new data nor receive data from other nodes; consequently, nodes in the data accumulating area inevitably become invalid nodes. Thus, data accumulation caused by the energy-harvesting problem of a single node decreases the performance of the entire network. This phenomenon has the least impact when it occurs at edge nodes, and it has the most considerable impact at relay nodes, especially at nodes around the sink node.

After a node wakes up, it sends the data buffered by the node. If a node does not receive the data request packet in threshold T_t, it is judged that data accumulation has occurred. This paper comprehensively considers that the threshold value used by all nodes is set as three times the average number of awakenings of nodes N_w, that is, $T_t = 3 \times 2N_w$ is considered the threshold for data accumulation judgment; the value of N_w verified by experiments as 6. Therefore, the processing mechanism for the data accumulation problem is as follows.

3.4.1. Increase Power

(1) When a node exceeds the preset number of wake-up times and fails to find a next-hop node, it increases the transmission power to send the emergency data beacon frame, and the data of the node are merged

(2) The node that receives the urgent data beacon sends a suppression packet. When other nodes in the network receive the suppression packet, they stop data transmission and enter a random back-off state to wait for the next data transmission or enter a sleep state to harvest energy

(3) After receiving the suppression packet, the node where data accumulation occurs increases the power of the fused data and transmits them. After the data transmission is complete, the node reduces the transmission power and restores to the original power value

(4) Then, the accumulation of the number of times is restarted, that is, the aforementioned process is repeated until the threshold number of times is exceeded again

(5) When the channel is idle, other suppressed nodes restart the normal data transmission process

When the transmission power of a certain node in a network is too large, multiple nodes receive the emergency data beacon frame simultaneously; these nodes also receive the emergency data packet at the same time. Although this process ensures the success of emergency data transmission, the priority protocols of emergency data packets may inhibit the sending of valid data packets to some extent, leading to wastage of resources. In this study, the following methods are used to control transmission power.

Let us assume that the initial communication radius of node S is R_0, the network area is $C_r \times C_r$, and the number of nodes is N. When the connectivity of node S is 0, the following mechanism is used to ensure that node S finds and communicates with only the next-hop node as far as possible:

(1) Accumulation of the average number of wake-up times N_w for the node to find the next-hop node

(2) Calculation of the size Sn of the average area occupied by the node using the following:

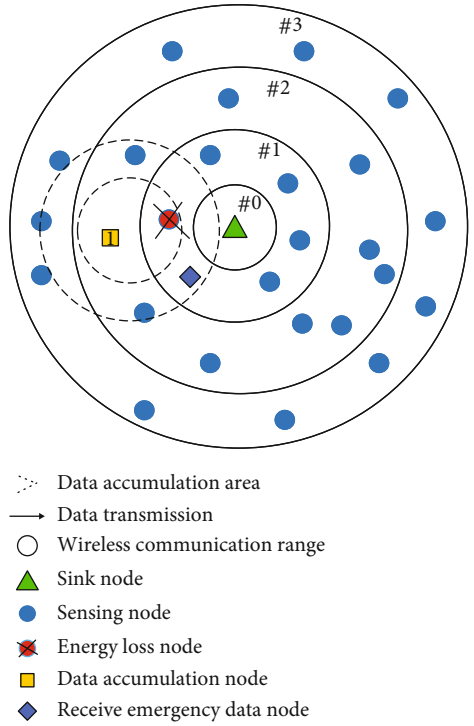


Figure 4: The case of increase power.

$$S_n = \frac{C_r \times C_r}{N}. \quad (1)$$

(3) Calculation of the radius R_i of the area where the power area needs to be increased. In this study, the number of times a node increases its power is equal to the average number of wake-up times Nw of the node, and this number is equally distributed to all radius $<1.5R_0$. The calculation is performed:

$$R_i = \sqrt{\frac{\pi \times R_0^2 + 2i \times S_n}{N}}, \quad (2)$$

(4) When the wake-up times of the node exceeds $2N_w$, the radius R_i of each increased power area is calculated using Formula (2)

(5) When $R_i > 1.5R_0$, the data are transmitted to the same layer node instead of increasing power, and from there, the data are sent to the sink node by a node in the same layer

Figure 4 shows a schematic of the power increase process. The data accumulation node increases the power, that is, it increases the wireless communication range; consequently, all nodes within the wireless communication range of the node receive the accumulated emergency data.

3.4.2. Same Layer Forwarding. Although EH-WSN relaxes the restriction on node transmission power to some extent, the following effects may occur when the power is excessively large:

(1) Effect on the normal transmission process of nodes within a certain range

(2) Excess consumption of the energy of the node, which increases the time required by the node to wake-up next time

(3) Broadcast storm: When the transmission power of the node is increased according to method 1 and the next-hop node is still not found, the wireless communication range of the node does not increase. Rather, the data are sent to another node in the same layer, and then, the same layer node sends the emergency data packet to the sink node. Herein, a hierarchical aggregation model is used to send emergency data to nodes at the same layer, ensuring success of the emergency data reaching the aggregation node. The same layer forwarding diagram is presented in Figure 5

As shown in the figure, node 1 becomes a data accumulation node because it cannot find the next-hop node. When the node cannot find its next-hop node after power is increased, it sends the data packet to nodes 1 and 2 in the same layer, bypassing the dead node, and then, again, the same layer node is used to send the accumulated data to the sink node.

3.4.3. "Layer-Down" Processing. Herein, a hierarchical aggregation model is used, in which a node sends its sensing data and the data received from an outer node to its inner node. When a node has accumulated data, it will "layer-down" and no longer request data from outer nodes. Instead, it participates in the data request processes of its peer nodes and outer nodes; the priority of data request packets of peer nodes is higher than that of outer nodes. "Layer-down" is advantageous as it can avoid further accumulation of data. The node maintains the "layer-down" state until it receives a data request packet from the inner node again; then, it judges whether to exit the "layer-down" state and then restarts the normal data transmission process. The node continues to accumulate the number of wake-ups that have not received a data request packet in the "layer-down" state and uses increased power and the same layer forwarding method for data transmission. The "layer-down" processing algorithm is presented in Figure 6.

As shown in Figure 6, when the energy-deficient node dies, node 1 judges itself to become a data accumulation node. It begins to participate in the data request phase of nodes 1 and 2 and the outer node 3 of the same layer. It may send the data of this node to the same layer node, or the outer layer may use the same layer and outer layer nodes to detour data to the sink node. The flow chart of PSL accumulation data processing mechanism is shown in Figure 7. When the node exceeds the threshold and cannot find the

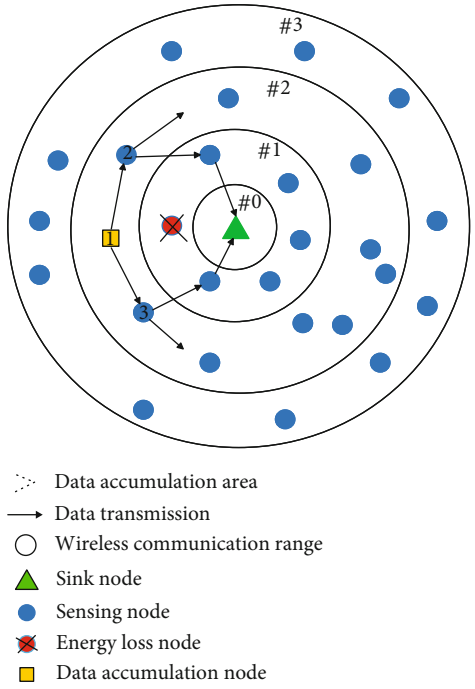


Figure 5: The case of peer forwarding.

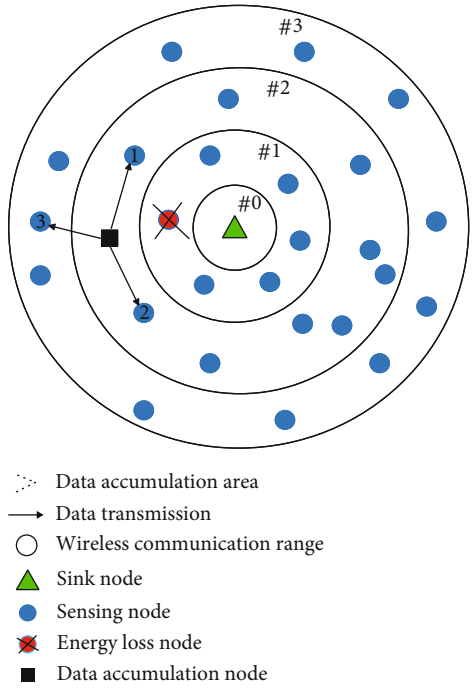


Figure 6: The case of "layer-down" processing.

next-hop node, the transmission power to send data is increased. If the node increases the power and fails to find the next-hop node, it will send the accumulation data packet to the peer node of the data accumulation node. If the data cannot be forwarded by the same layer forwarding, the data of the node is sent to the sink node by means of "layer-down" processing.

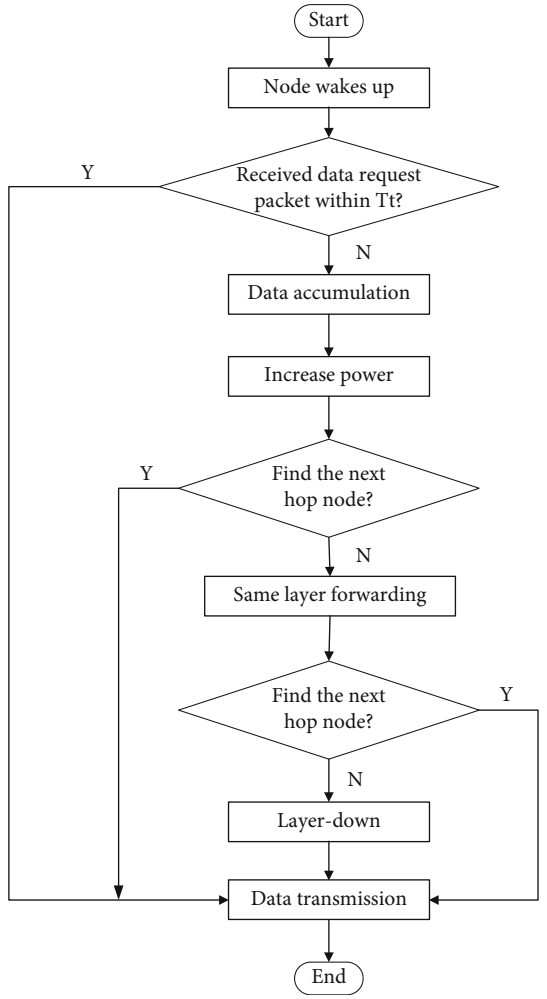


Figure 7: PSL accumulation data processing flow chart.

4. Simulation Results

4.1. Simulation Parameter Setting. In this study, MATLAB is used for simulations. The environment settings for simulation are as follows:

(1) In total, 10–100 energy-harvesting sensor nodes are randomly deployed in a simulation area of 300 × 300 m^2

(2) Size of the data packet (S_d) is 100 bytes, whereas that for each for the data request beacon frame, request response packet, data transmission beacon frame, and emergency data beacon frame is 15 bytes

(3) The average transmission range of sensor nodes is 70 m

(4) The transmission rate of each sensor node is 250 Kbps

(5) The average energy-harvesting rate is 1–10 mW

(6) Receiving power and transmitting power of the node are 72.6 mW and 83.7 mW, respectively

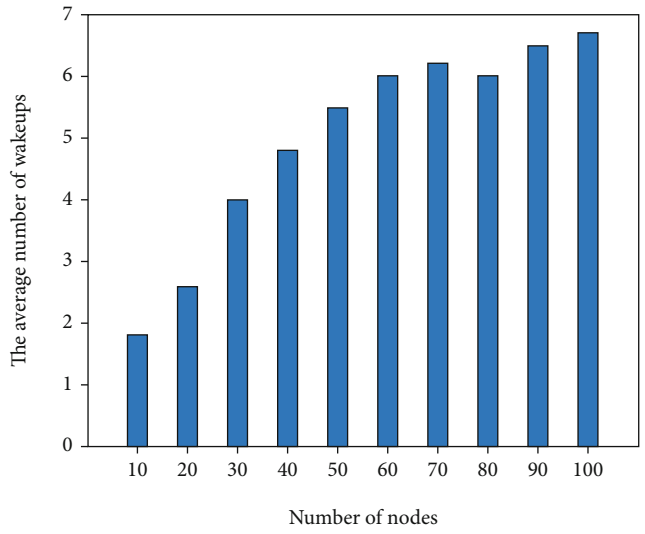


Figure 8: The relationship between the number of nodes and the average number of wake-ups.

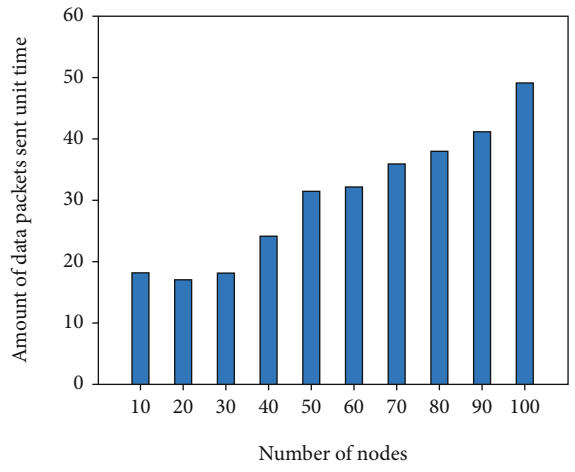


Figure 9: The number of accumulated data packets sent per unit time.

4.2. Analysis of Simulation Results. Figure 8 shows the relationship between the number of nodes and the average wake-up times of the identified next-hop node. When the number of nodes is only 10 and 20, the number of wake-ups for identifying the next-hop node for data transmission is twice on average. This is because the number of nodes in the lower network is small, and the probability that the channel remains idle is high. With an increase in the number of nodes, the number of times the node finds the next-hop node increases. This is because as the number of nodes increases, the number of signal collisions increases, and the probability that the channel remains idle after the random back-off time of the node decreases. As a result, the node will turn to a sleep state or reselect a random back-off time, because of which the average number of wake-up times for identifying the next-hop node increases with the increasing number of nodes.

The blue histogram in Figure 9 shows the relationship between the number of nodes and the number of data accumulation packets transmitted per unit time. With an increase in the number of nodes in the network, the number of data accumulation packets transmitted per unit time increases. When the number of nodes is small, there are more isolated nodes in the network. At this time, most of the data accumulation packets are generated and sent by the isolated nodes. Because the number of nodes is small, the isolated nodes can determine the next step through the PSL accumulation data processing mechanism. The probability of node hopping is also small. As the number of nodes increases, the number and frequency of isolated nodes to be accessed for identifying the next-hop node increase, and the number of data accumulation packets increases.

The network throughput when the number of nodes changes is shown in Figure 10. When the number of nodes in ID polling is <50, the network throughput increases with an increase in node density. When the number of nodes in the network reaches 50, the network throughput remains nearly unchanged. The throughput of ID polling is the lowest compared with the probabilistic polling and the MAC protocol proposed in this paper. The sink node selects only one node to communicate with at one time. The condition for successful data transmission is that the receiving node is in the wake-up state and receives only the beacon frame of this node ID number. Energy-harvesting nodes periodically harvest energy from the environment; however, the energy-harvesting process is longer than the communication process. Therefore, the probability that the node is in the receiving state with sufficient energy and receives only the beacon frame with its own ID is very small. Probabilistic polling has a higher throughput than ID polling because it uses competitive probability values instead of node ID numbers to select the communication node, avoiding having to select a fixed node as the only communication node each time, which increases the success rate of data transmission and further increases the network throughput. The proposed MAC protocol, thus, has higher throughput, as herein a node acts as the receiving node to transmit a data request packet after waking up, and it informs each transmitting node of the number of received response packets. The actual value replaces the probabilistic value, decreasing the time of collision and facilitating convergence of the probability values.

The packet loss rate with changes in the number of nodes is shown in Figure 11. ID polling only selects a certain node to communicate with each time, and no signal collision or collision occurs. The packet loss rate is independent of the number of nodes and remains zero. The packet loss rate of both probabilistic polling and the MAC protocol proposed in this study increases with an increase in the number of nodes. Probabilistic polling has a low packet loss rate before 50 nodes. QAEE-MAC always sends high-priority packets first, causing low-priority packets to fail. For a higher number of nodes, the proposed MAC protocol has better performance because in this case, actual values are used to increase the success rate of data transmission.

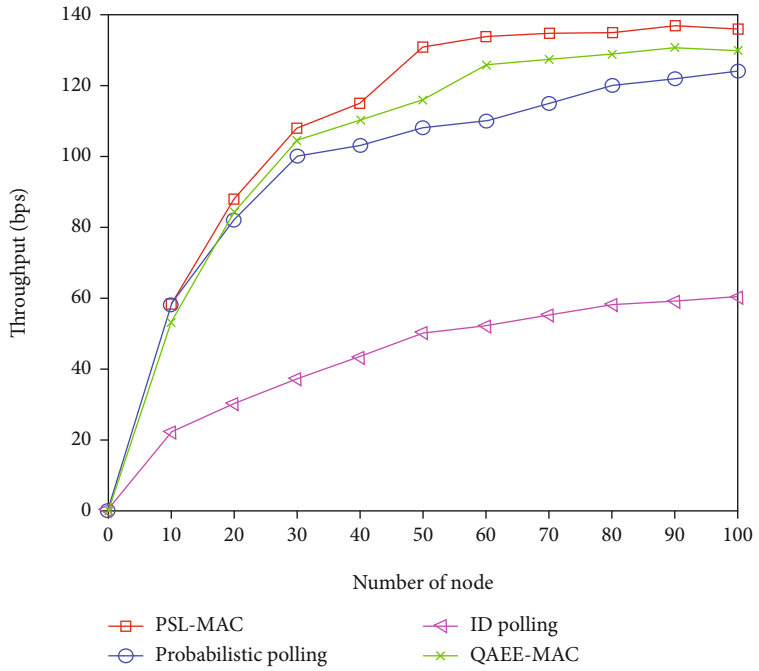


Figure 10: Network throughput when the number of nodes changes.

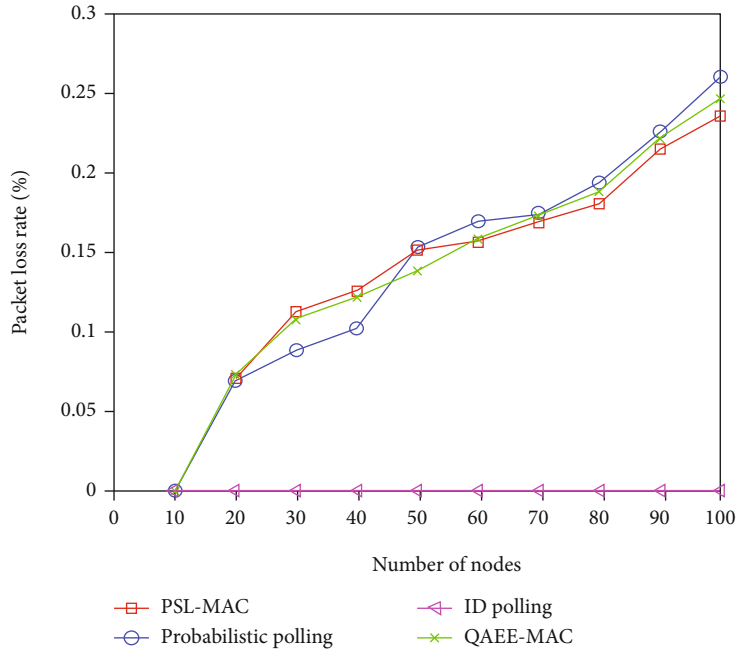


Figure 11: Packet loss rate when the number of nodes changes.

The throughput when the node energy-harvesting rate changes is shown in Figure 12. The network throughput of all the aforementioned three protocols increases with an increase in the energy-harvesting rate. The proposed MAC protocol has higher throughput. For ID polling, when the energy-harvesting rate increases, the node charging process is shortened, the probability of the node receiving the polling packet with the node ID is increased, and the throughput is also increased. With an increase in the energy-harvesting rate, the number of nodes in the wake-up state at the same time increases, and the probability value of probabilistic polling remains low, avoiding collisions and ensuring success of data transmission. QAEE-MAC uses priority to adjust the contention window size of the receiver, which reduces the throughput. For the proposed MAC protocol, when the energy-harvesting rate increases, the number of communicable nodes increases, and the network throughput increases accordingly. When the energy-harvesting rate

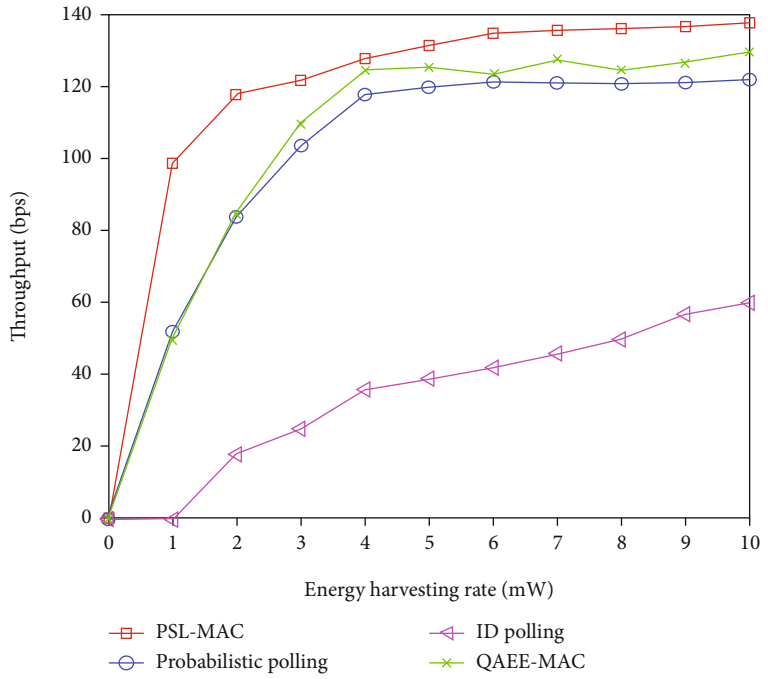


Figure 12: Throughput when the energy-harvesting rate changes.

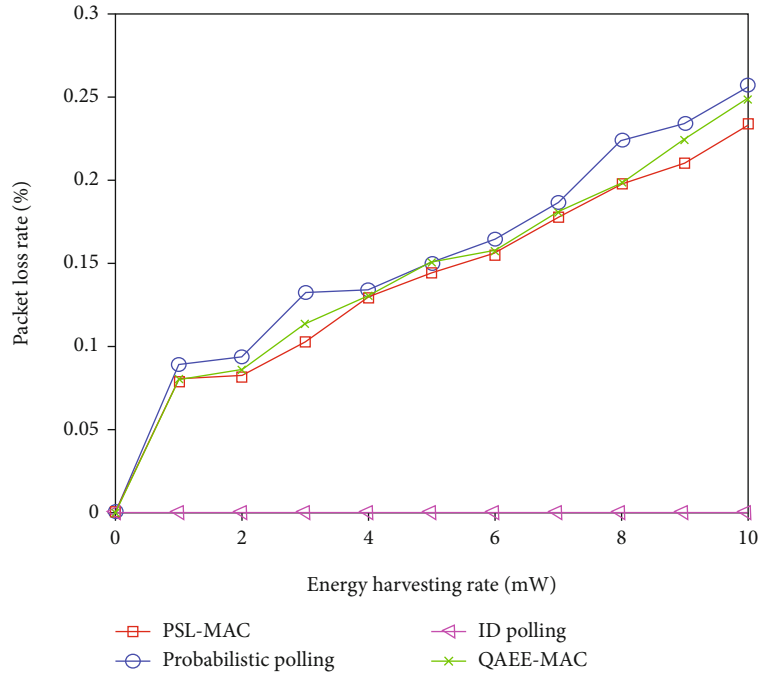


Figure 13: Packet loss rate when the energy-harvesting rate changes.

reaches a certain level, the throughput is not greatly affected by the energy-harvesting rate. This is because in this case, the sink node only establishes a connection with a certain node each time it wakes up, and more nodes return to the charging state to avoid collisions or reselect random back-off values to participate in the next competition.

The packet loss rate with changes in the node energy-harvesting rate is shown in Figure 13. ID polling only selects a certain node to communicate with it each time, thereby avoiding data collision, and thus, the packet loss rate remains zero. The probabilistic polling, QAEE-MAC, and the proposed MAC protocol packet loss rate in this study increase because of an increase in the energy-harvesting rate, which in turn increases the number of nodes in the wake-up state, the probability of channel competition and collision, and the packet loss rate. However, the proposed MAC protocol has a

lower packet loss rate. This is because probabilistic polling uses the AIMD method to calculate the probability of competition for conflict avoidance, QAEE-MAC always sends high-priority packets first, causing low-priority packets to be in a dormant waiting state, and this paper uses the actual value of neighboring nodes when the node wakes up as the probability of competition to avoid conflict, resulting in a low packet loss rate.

5. Conclusions

This study proposes a MAC protocol called PSL for limited energy-harvesting conditions to solve the data accumulation problem in EH-WSNs. A data transmission mechanism based on probability control is also proposed to reduce data conflicts during data transmission. Simulation results show that the proposed PSL scheme can solve the problem of data accumulation and improve the network transmission rate.

Data Availability

All sensor nodes can collect data from the environment and forward the data they have collected.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This paper was supported by the National Natural Science Foundation of China Grants Nos. 61761035, 61461037, and 61661041 and the Natural Science Foundation of Inner Mongolia Autonomous Region 2019MS06030.

References

[1] X. Chen, C. Wu, Z. Liu, N. Zhang, and Y. Ji, "Computation offloading in beyond 5G networks: a distributed learning framework and applications," *IEEE Wireless Communications*, vol. 28, no. 2, pp. 56–62, 2021.

[2] X. Liu, C. Sun, M. Zhou, C. Wu, P. Bao, and P. Li, "Reinforcement learning-based multislot double-threshold spectrum sensing with Bayesian fusion for industrial big spectrum data," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 4, pp. 3391–3400, 2021.

[3] A. E. Assaf, S. Zaidi, S. Affes, and N. Kandil, "Effificient node localization in energy-harvesting wireless sensor networks," in *IEEE international conference on ubiquitous wireless broadband*IEEE.

[4] X. Liu, Q. Sun, W. Lu, C. Wu, and H. Ding, "Big-data-based intelligent spectrum sensing for heterogeneous Spectrum communications in 5G," *IEEE Wireless Communications*, vol. 27, no. 5, pp. 67–73, 2020.

[5] O. Bouachir, A. Ben Mnaouer, F. Touati, and D. Crescini, "EAMP-AIDC-energy aware mac protocol with adaptive individual duty cycle for EHWSN," in *Wireless Communications & Mobile Computing Conference*, pp. 2021–2028, IEEE, 2017.

[6] S. Lin and J. Yifeng, "Study on the influence of data fusion on clustering energy in wireless sensor networks. Service Operations, Logistics and Informatics, 2009. SOLI'09," in *IEEE/INFORMS International Conference on*IEEE.

[7] J. A. Ansere, G. Han, H. Wang, C. Choi, and C. Wu, "A reliable energy efficient dynamic spectrum sensing for cognitive radio IoT networks," *IEEE Internet of Things Journal*, vol. 6, no. 4, pp. 6748–6759, 2019.

[8] A. Kansal, D. Potter, and M. B. Srivastava, "Performance aware tasking for environmentally powered sensor networks," in *ACM SIGMETRICS performance evaluation review*, vol. 32no. 1, pp. 223–234, ACM, 2004.

[9] C. Park and P. H. Chou, "Ambimax: autonomous energy harvesting platform for multi-supply wireless sensor nodes," in *2006 3rd annual IEEE communications society on sensor and ad hoc communications and networks*, vol. 1, pp. 168–177, IEEE, 2006.

[10] T. Zhu, Z. Zhong, Y. Gu, T. He, and Z. L. Zhang, "Leakage-aware energy synchronization for wireless sensor networks," in *proceedings of the 7th international conference on Mobile systems, applications, and services*, pp. 319–332, ACM, 2009.

[11] V. Raghunathan, A. Kansal, J. Hsu, J. Friedman, and M. Srivastava, "Design considerations for solar energy harvesting wireless embedded systems," in *Proceedings of the 4th international symposium on information processing in sensor networks*, IEEE Press, p. 64, 2005.

[12] A. Kansal, J. Hsu, M. B. Srivastava, and V. Raghunathan, "Harvesting aware power management for sensor networks," in *Proc. of the ACM/IEEE DAC*, pp. 651–656, 2006.

[13] A. S. A. Ismail, B. Subir, and A. S. Bayez, "AH-MAC: adaptive hierarchical MAC protocol for low-rate wireless sensor network applications," *Journal of Sensors*, vol. 2017, Article ID 8105954, 15 pages, 2017.

[14] N. Dragoni and X. Fafoutis, "ODMAC: on-demand MAC protocol for energy harvesting-wireless sensor networks," in *Acm symposium on performance evaluation of wireless ad hoc*ACM.

[15] Z. A. Eu, W. K. G. Seah, and H. Tan, "A study of MAC schemes for wireless sensor networks powered by ambient energy harvesting," in *International Conference on Wireless Internet. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering)*, 2008.

[16] T. Xu, M. Zhao, X. Yao, and K. He, "An adjust duty cycle method for optimized congestion avoidance and reducing delay for WSN," *Computers, Materials & Continua*, vol. 65, no. 2, pp. 1605–1624, 2020.

[17] C. Tang, "Research and analysis of WSN node location in highway traffic based on priority," *Journal of Quantum Computing*, vol. 2, no. 1, pp. 1–9, 2020.

[18] S. C. Kim, J. H. Jeon, and H. J. Park, "QoS aware energy-efficient (QAEE) MAC protocol for energy harvesting wireless sensor networks," *Convergence and Hybrid Information Technology*, 2012.

Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 7099494, 13 pages
https://doi.org/10.1155/2022/7099494

Research Article

Marine Drifting Trajectory Prediction Based on LSTM-DNN Algorithm

Xianbin Li,[1] Kai Wang,[1] Min Tang,[1] Jiangyi Qin,[1] Peng Wu,[1] Tingting Yang,[2] and Haichao Zhang[1]

[1]*National Innovation Institute of Defense Technology, Academy of Military Science, No. 53, Dongdajie Road, Fengtai District, Beijing 100071, China*
[2]*Navigation College, Dalian Maritime University, Dalian 116026, China*

Correspondence should be addressed to Kai Wang; cxywangkai@163.com and Min Tang; mtangcn@163.com

Received 15 February 2022; Accepted 26 May 2022; Published 2 July 2022

Academic Editor: A.H. Alamoodi

In this paper, the long short-term memory with dense neural network (LSTM-DNN) is first introduced to calculate marine drifting trajectory. Based on the Internet of Things technology and the LSTM-DNN algorithm, the marine drifting trajectory model is established. In this model, the information such as wind field, temperature field, ocean current motion field, and target attributes are included, and the influences of the above information on the trajectory model are studied in detail. In order to verify the proposed model, the marine experiments are carried out in the end. The results show that the predicted trajectory data matches well with the experimental trajectory data. By introducing DNN into the algorithm, computational accuracy of drifting trajectory can be significantly improved compared with the conventional LSTM-based prediction model. A detailed comparison of the two algorithms has also been given in the paper. The proposed remote sensing of marine drifting trajectory model can provide a high accurate trajectory prediction and will lead an important guidance in the marine search and rescue work.

1. Introduction

In recent years, the marine accidents have greatly affected the shipping and marine production. The methods to improve the efficiency of marine search and rescue have gained popularity among researchers [1–3]. Due to the particularity of sea conditions, the drifting characteristic of the objects is a key factor for rapid search and rescue [4, 5].

Nowadays, massive efforts have been down in predicting the trajectory, including human motion [6, 7], intelligent vehicles [8–11], service robots [12], surveillance systems [13, 14], wind power generation [15, 16], magnetic field intensity [17, 18], and ship trajectory [19, 20]. The long short-term memory (LSTM) is a neural network that is responsible for calculating the dependence between observations in a time series. Therefore, it is often used for forecasting purposes. By using the temporal dependence-based LSTM networks, Liu et al. predicted the ocean-temperature changes successfully [21]. Cruz and Bernardino used the LSTM associated with a pretrained convolutional neural network to improve the detection performance in videos captured by small aircraft [22]. Choi et al. used the LSTM to predict the occurrences of abnormally high water temperature phenomena [23]. Tang et al. proposed an improved LSTM model with a random deactivation layer to deal with the time series problem [24]. Zhao and Shi presented a method that consists of a density-based spatial clustering of applications with noise algorithm and the LSTM to cluster and predict the ship trajectory [25]. Park et al. proposed a marine intelligent collision avoidance algorithm, and the ship trajectory prediction model was developed by using bidirectional LSTM [26]. Gao et al. proposed a multistep prediction method for ship trajectory, by using a novel MP-LSTM method. The proposed method can be used to solve the problems of complex mapping relationships and large data requirements [27]. Although the LSTM can solve

the gradient disappearance and explosion problems of traditional recurrent neural networks, its accuracy still needs to be further improved for more accurate trajectory prediction schemes.

The dense neural network (DNN) is one of the most classic neural networks. It has the merits of easy to understand and convenient to apply [28]. Moreover, it shows excellent fitting ability for most nonlinear functions. Once the depth of the fully connected neural network is increased, the function can be accurately fitted. The algorithm that combines the LSTM and the DNN can further improve the accuracy and flexibility of predicting trajectories. However, the application of LSTM algorithm in the marine mainly focuses on ship trajectory prediction. Researches on the marine drifting trajectory prediction model focus on smaller and unmotivated target are rarely reported so far, especially those based on the LSTM-DNN algorithm.

In this paper, we are interested in the prediction of the marine drifting trajectory, which can be used in the marine rapid search and rescue. The remote sensing of marine drifting trajectory prediction model is build based on the LSTM-DNN algorithm. Meanwhile, the information such as wind field, temperature field, ocean current motion field, and target attributes are included, and the influences of the above information on the trajectory model are studied in detail. The marine experiments are carried out to verify the accuracy of the proposed model. The conventional LSTM-based prediction model is used for comparison. By introducing DNN into the LSTM algorithm, the accuracy of drifting trajectory can be significantly improved. A detailed comparison of the two algorithms has also been given in this paper. The proposed remote sensing of marine drifting trajectory model can provide a high accurate trajectory prediction and will lead an important guidance in the marine search and rescue work. This work is aimed at investigating a high accurate and rapid prediction model for marine drifting trajectory. The contribution of the paper is as follows:

(1) The DNN is introduced into the conventional LSTM-based prediction model. A detailed comparison between the LSTM and LSTM-DNN-based model has been studied

(2) Compared with the conventional LSTM-based marine drifting trajectory model, the proposed model shows the merits of high accuracy

(3) The marine drifting trajectory database under the condition of ocean current and drifting, environmental field, target physical properties, and prediction duration is established by trials

The rest chapters of this paper are arranged as follows. Some related works are discussed in Section 2. The principle of the LSTM-DNN-based model is presented. System model and problem formulation are presented in Section 3. In Section 4, the test scheme is presented, and the real data and the simulation results are given to verify the correctness and the accuracy of the proposed model. And the conclusion part is shown in Section 5.

2. Related Work

In literature, several research works are related to the marine trajectory prediction, including the ship trajectory prediction mentioned in the introduction part. In addition to predicting the trajectory of marine ships, some researches have also been down on the trajectory prediction for marine search and rescue. For example, Zhu et al. proposed an ensemble trajectory prediction model for maritime search and rescue [29]. The authors proposed a regional subgrid velocity model based on drifting buoy data, and the proposed model is used to simulate the unsolved velocity that composed of turbulence and advection simulation errors. In Ref. [30], a drifting trajectory prediction model based on the object shape and stochastic motion features was proposed. Compared to the traditional factors of wind and currents, the proposed method also involved the effects of the object shape and stochastic motion features. Meanwhile, the computer simulation-based method was used to estimate the uncertainty parameters of the stochastic factors of the drifting objects. By using the model for the trajectories of objects drifting at the ocean surface, Blanken et al. proposed a fuzzy number-based framework for quantifying and propagating uncertainties [31]. Based on historical HF coastal radar data sets, Jitkajornwanich et al. proposed a predictive model for future current data [32]. By utilizing association rule mining combined with an object dispersion concept, the full potential of HF radar systems was exploited. Shchekinova et al. used the high-resolution ocean forecast and atmospheric data to solve the effects caused by the wind-induced drift on Lagrangian trajectories of surface sea objects [33].

In this paper, we focus on designing a marine drifting trajectory prediction model that can be used in the field of marine search and rescue. Based on the LSTM-DNN algorithm, a highly accurate prediction model of marine drifting trajectory is proposed.

3. System Model

An independent satellite maritime rescue system is established based on LSTM-DNN algorithm, satellite communication, and marine environment information The LSTM-DNN algorithm is used to deduce the real-time dynamic information of the overboard target needing to be searched and rescued.

3.1. The LSTM-DNN Predicting Model. Figure 1 shows the schematic diagram of the remote sensing of the marine drifting trajectory model. The proposed model is mainly composed of LSTM, DNN, and embedded encoder. The environment field information is processed by embedded encoder. The marine environment field variables are encoded into drifting data features by embedded encoder. In this paper, the marine environmental information includes current trajectory motion field, wind field, and temperature field. Each embedded encoder encodes an environmental information, in which the input of current trajectory motion field is the longitude and latitude coordinates and velocity component of the time and position corresponding

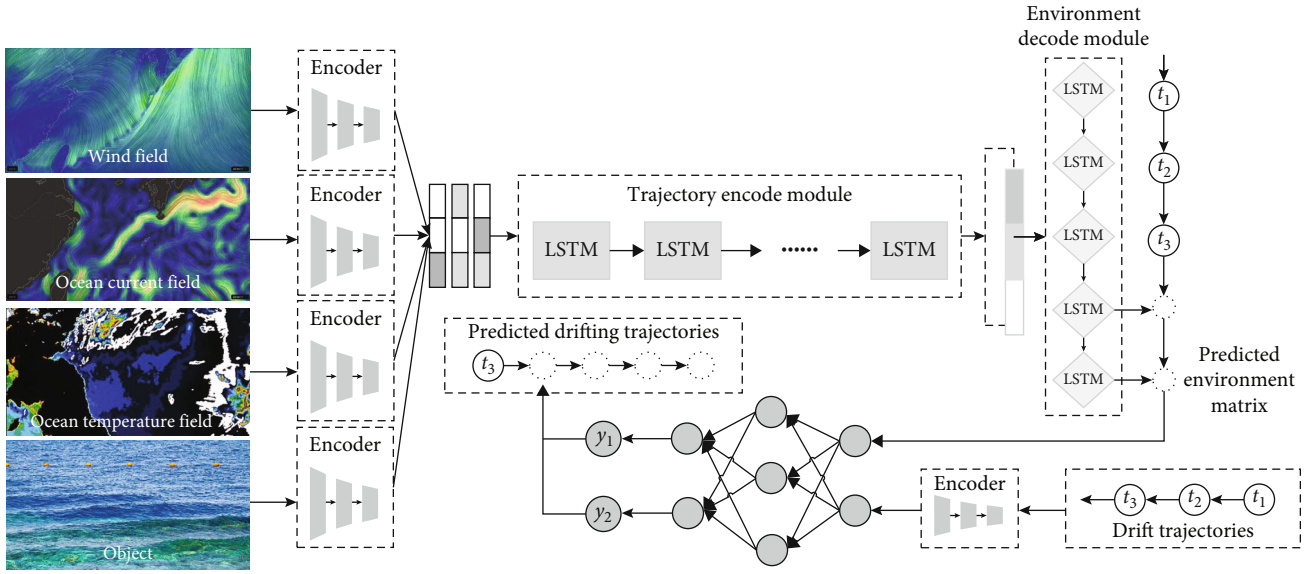


Figure 1: The schematic diagram of the remote sensing of the marine drifting trajectory model.

to the drift, and the velocity component is the east component and the north component, respectively. The input of wind field is the velocity component at the coordinate corresponding to the drift. The input of the temperature field is the ocean surface temperature of the drifting path. The encoding process is to calculate the characteristic matrix of the corresponding drifting track in the same time period according to the initial state of each marine environment variable and the longitude and latitude coordinates of each component and take the matrix and the characteristic matrix of drifting track data as the input of the drifting track prediction submodule. The environmental characteristic matrix of the future time is predicted in the trajectory encode module, and the obtained matrix is fused with the drift trajectories into the DNN module to predict the drifting track coordinates at the future time. Trajectory encode module is mainly responsible for fusing environmental data matrix. Environment decode module is mainly responsible for predicting future environmental information.

N_1, N_2, N_3, and N_4 represent the ocean current tracks, wind field data, temperature field, and N_4 drifting track, respectively. There is $N = N_1 + N_2 + N_3 + N_4$. The ocean current track data can be represented by $O_1, O_2, \cdots, O_{N_1}$, the wind field corresponding to each ocean current can be represented by $W_1, W_2, \cdots, W_{N_2}$, the drifting track data is represented by $P_1, P_2, \cdots, P_{N_4}$, and the temperature is represented by $K_1, K_2, \cdots, K_{N_3}$. Normalize the data of the same time node, and the data coordinates of the i th track at the time node be expressed as $S_i^t = (x_i^t, y_i^t)$, where $o_1, o_2, \cdots, o_{N_1}$, $p_1, p_2, \cdots, p_{N_1} \in i$, input the coordinate point of the above trajectory at $t = 1, 2, \cdots, t_o$ time. Based on the environmental prediction submodule, output the trajectory characteristic data x_i of marine environmental variables at $t = t_{0+1}$, $t_{0+2}, \cdots, t_{pred}$ time; then, the matrix x_i and drifting trajectory data p_i are used as the inputs of the drifting trajectory prediction submodule to further learn the time correlation between environmental variables at different locations and drifting trajectory variables through the DNN layer. Through the module, the drifting track coordinates at $t = t_{0+1}, t_{0+2}, \cdots, t_{pred}$ can be calculated. The environment prediction submodule is composed of a stacked LSTM network.

3.2. Equations. All figures and tables should be cited in the main text as Figure 1, Table 1, etc. In the proposed marine drifting trajectory model, the ocean current trajectory data, wind field, and temperature field are encoded into the input characteristic matrix of the environmental prediction submodule by the embedded encoder, represented by $x_i = \{\overrightarrow{x_1}, \overrightarrow{x_2}, \cdots, \overrightarrow{x_N}\}$. The environment prediction submodule is mainly composed of input module, redundancy module, and output module; x_i^t is the characteristic matrix of marine environmental variables at time t; h_i^{t-1} is the short-term memory matrix output by LSTM at time $t-1$; s_i^{t-1} is the long-term memory matrixoutput by LSTM at time $t-1$; then, the input module can be expressed as

$$a_\xi^t = \sum_{i=1}^{I} W_{i\xi} x_i^t + \sum_{h=1}^{H} W_{h\xi} b_h^{t-1} + \sum_{c=1}^{C} W_{C\xi} S_c^{t-1}, \quad (1)$$

where $W_{i\xi}$ represents the weight coefficient matrix between the incoming environment data matrix x_i and the input module, $W_{h\xi}$ represents the weight coefficient matrix between the short-term memory unit and the input module at time $t-1$, $W_{c\xi}$ represents the weight coefficient matrix between the long-term memory unit and the input module at time $t-1$, b_h^{t-1} is the short-term memory unit matrix, S_C^{t-1} is the long-term memory unit matrix, $b_\xi^t = f(a_\xi^t)$ is obtained through the activation function, and the activation function is tanh, which makes a linear transformation and

Table 1: Evaluate results.

	Start time: 2021-12-15 11:27 end time: 2021-12-15 17:37					
	Positioning time	Latitude	Longitude	Speed	Direction	Positioning mode
1	2021/12/15 11:27	38.85251	121.6708	0	157	GPS
2	2021/12/15 11:37	38.85313	121.6709	7.41	331	GPS
3	2021/12/15 11:48	38.85375	121.6706	0	331	GPS
4	2021/12/15 11:57	38.85439	121.6707	3.7	119	GPS
5	2021/12/15 12:07	38.85513	121.671	14.82	10	GPS
6	2021/12/15 12:17	38.8562	121.6718	3.7	19	GPS
7	2021/12/15 12:27	38.85708	121.673	18.52	309	GPS
8	2021/12/15 12:47	38.85844	121.6758	1.85	345	GPS
9	2021/12/15 13:37	38.862	121.6876	7.41	161	GPS
10	2021/12/15 13:57	38.86253	121.6946	0	161	GPS

nonlinear mapping on the data to improve the convergence speed of the model.

The redundant module can be expressed as

$$a_{\varphi}^{t} = \sum_{i=1}^{I} W_{i\varphi} x_i^t + \sum_{h=1}^{H} W_{h\varphi} b_h^{t-1} + \sum_{c=1}^{C} W_{C\varphi} S_c^{t-1}, \tag{2}$$

where W_{φ} represents the weight coefficient matrix between the incoming environment data matrix x_i and the input module, the weight coefficient matrix between the short-term memory unit and the input module at time $t-1$, and the weight coefficient matrix between the long-term memory unit and the input module at time $t-1$. The $b_{\varphi}^{t} = f(a_{\varphi}^{t})$ is obtained through the activation function.

The final data matrix is calculated by the output module to obtain the updated environment data matrix, which can be expressed as

$$a_{\omega}^{t} = \sum_{i=1}^{I} W_{i\varphi} x_i^t + \sum_{h=1}^{H} W_{h\omega} b_h^{t-1} + \sum_{c=1}^{C} W_{c\omega} S_c^{t-1}. \tag{3}$$

Activate get $b_{\omega}^{t} = f(a_{\omega}^{t})$.

The state matrix of neurons can be expressed as

$$a_c^t = \sum_{i=1}^{I} W_{ic} x_i^t + \sum_{h=1}^{H} W_{hc} b_h^{t-1}, \tag{4}$$

$$S_c^t = b_{\varphi}^t S_c^{t-1} + b_{\xi}^t g\left(a_c^t\right). \tag{5}$$

The output matrix of the final environmental prediction submodule can be expressed as

$$q_i = b_{\omega}^t h\left(S_c^t\right). \tag{6}$$

The output q_{i} and drifting track data matrix p_{i} are transmitted to the drifting track path prediction submodule, and the matrix $q_{\mathrm{i}}, p_{\mathrm{i}}$ is output through DNN network, drifting track coordinates at $t = t_1, t_2, \cdots, t_{pred}$ in the future are output through linear network. The whole process is shown in Figure 2.

The characteristic state matrix of DNN intermediate hidden layer is calculated by the following formula.

$$z = w^{(1)} * \left(\vec{\alpha}\right)^T + \left(b^{(1)}\right)^T. \tag{7}$$

In the formula, $\vec{\alpha} = [q_{\mathrm{i}}, p_{\mathrm{i}}]$, the weight coefficient matrix is expressed as

$$w^{(1)} = \begin{bmatrix} w(q_i, 1), w(p_i, 1) \\ w(q_i, 2), w(p_i, 2) \\ w(q_i, 3), w(p_i, 3) \end{bmatrix}, b^{(1)} = [b_1, b_2, b_3]. \tag{8}$$

Finally, the predicted value is

$$\hat{y} = w^{(2)} z^T + \left(b^{(2)}\right)^T. \tag{9}$$

In the formula, the weight coefficient matrix is expressed as

$$w^{(2)} = \begin{bmatrix} w(1,4), w(2,4), w(3,4) \\ w(1,5), w(2,5), w(3,5) \end{bmatrix}, b^{(2)} = [b_4, b_5]. \tag{10}$$

4. Test Scheme

In this paper, the real data from the northern sea area of the Yellow Sea in Dalian is used to train the proposed network framework. Among them, the ocean current trajectory data, wind field, and temperature field are obtained by field measurement, and the drifting trajectory data is obtained by simulating the real drifting test of counterweight buoy in the sea. Through the visual analysis of the data, the displacement generated by the current track and drifting track changes little in a short time, which can be regarded as uniform linear motion.

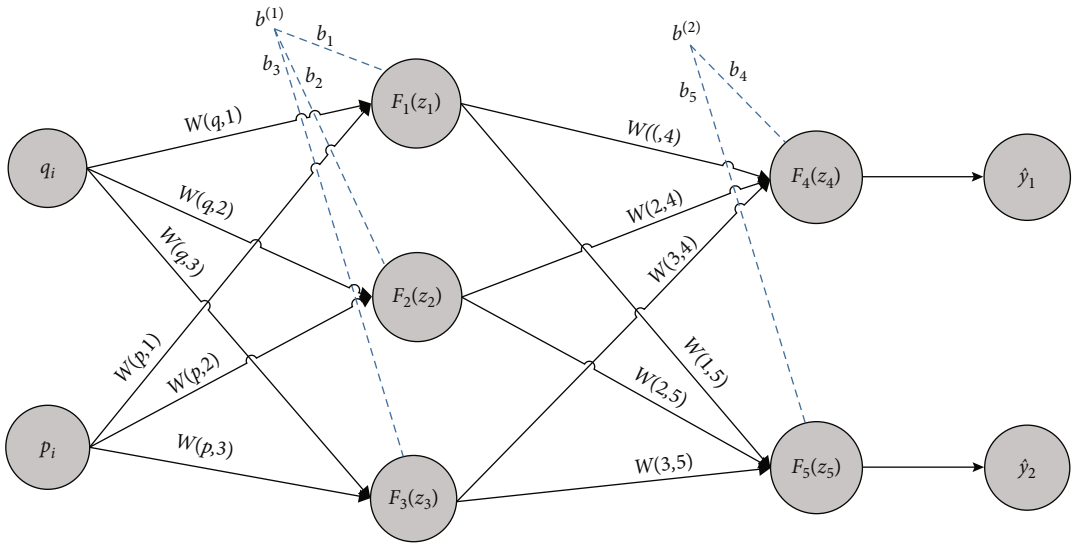


Figure 2: The output architecture of the DNN linear network.

Figure 3: Unequal volume circular scale.

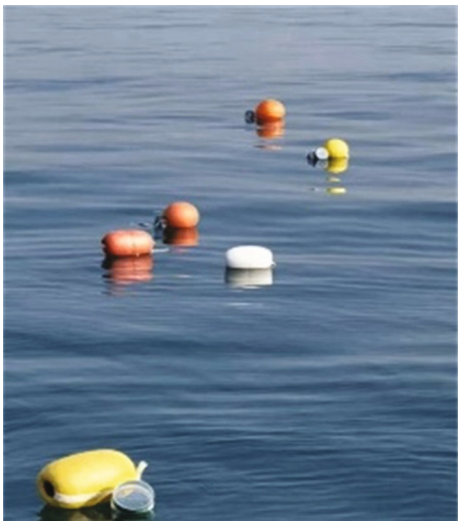


Figure 4: Drifting buoy.

4.1. Data Acquisition Scheme. In the same sea area located in the north of the Yellow Sea in Dalian, the water temperature from the ocean surface to 30 m below the sea is tested based on sensor conductivity temperature depth (CTD) equipment.

As shown in Figure 3, buoys with GPS positioning information are configured, including 40 buoys with different volumes with a diameter of about 6-30 cm. The forty buoys (different volumes) are thrown at a distance of 2 km offshore Dalian. After the current exchange period, the longitude and latitude information and direction data of buoys in different periods are recorded, respectively, and each buoy is divided into a group of drifting data.

Anemometer is used to record the wind speed; direction and the atmospheric temperature data at the place where the buoy is released. The ocean current data can be obtained by releasing drifting bottles every 500 m within 2 kilometers, so that there will be five drifting bottles used to record ocean current data. And then record the longitude, latitude, and direction data three hours after the current exchange period as the ocean current reference data.

Table 2: Wind field data.

	Wind speed (m/s)	Direction (°)
Point 1	0	358
Point 2	2.4	354
Point 3	2.2	355
Point 4	0	358
Point 5	2.9	356

4.2. Test Process. Training data collection:

Prepare the experimental ship and carry the GPS sealed bottle, CTD, buoy, anemometer, and other equipment.

Drive the ship to the northern sea area of the Yellow Sea, and release a sealed bottle and a group of buoys every 0.5 km

Table 3: Temperature field data.

Data	Vbatt [V]	Press [dBar]	Temp [°C]	Sound [m/s]	Sigma [kg/m^3]	Time	Date
1	6.27	0.16	9.43	1445.02	-0.25	11 : 20 : 55	2021/12/15
2	6.26	0.17	9.44	1445.03	-0.25	11 : 20 : 55	2021/12/15
3	6.26	0.16	9.44	1445.07	-0.25	11 : 20 : 55	2021/12/15
4	6.26	0.16	9.45	1445.1	-0.25	11 : 20 : 55	2021/12/15
5	6.26	0.14	9.46	1445.12	-0.25	11 : 20 : 55	2021/12/15
6	6.26	0.16	9.46	1445.15	-0.25	11 : 20 : 56	2021/12/15
7	6.27	0.16	9.47	1445.17	-0.25	11 : 20 : 56	2021/12/15
8	6.26	0.17	9.48	1445.19	-0.25	11 : 20 : 56	2021/12/15
9	6.27	0.14	9.48	1445.21	-0.25	11 : 20 : 56	2021/12/15
10	6.26	0.16	9.48	1445.22	-0.25	11 : 20 : 56	2021/12/15

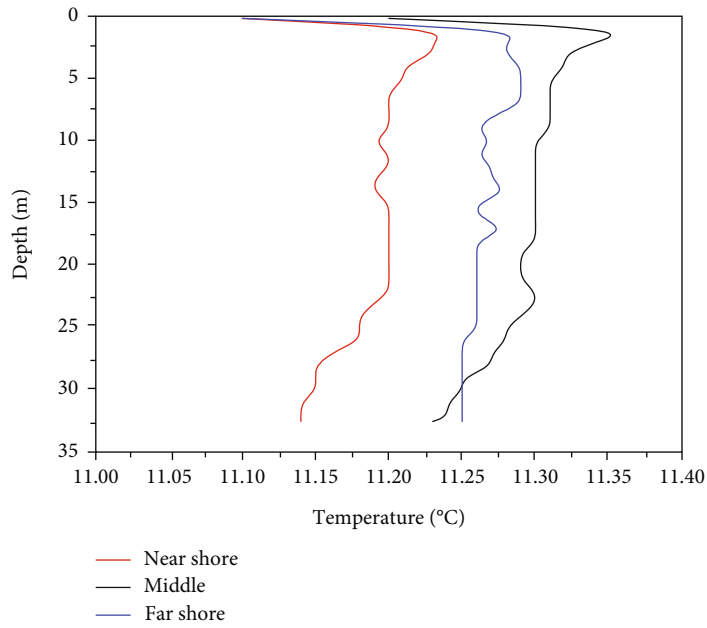


Figure 5: Temperature change curve of near shore release point, middle release point, and far shore release point during descent.

(eight buoys in a group, different specifications, and pre-prepared groups), a total of five groups as shown in Figure 4.

After the release, the information of sea surface wind field and temperature field is measured and the data is recorded.

Record the drifting position information of the separate sealed bottle for three hours, which can be used as ocean current data. Record the three-hour buoy position information, which is the drifting track. During this time, the data can be preprocessed according to the model requirements.

After waiting for three hours, count the overall data, import the data into the model for training after data preprocessing, and save the model.

Prediction stage:

Drive the ship to 2 km off the coast of Dalian.

Release buoy and life jacket with GPS positioning signal.

Record the position information of coordinate points every 10 minutes when the buoy floats.

After data preprocessing, the first four floating positions and the positions sent by personnel are transferred to the pretrained model to obtain the predicted position information of the next time node, record the corresponding data, and drive the ship to search for drift near the coordinate point.

4.3. Data Processing. Based on the above test process, the following ocean current trajectory data are collected, some of which are shown in Table 1.

The longitude and latitude coordinates, motion speed, and motion direction of the current track are collected in Table 1 every ten minutes. The lack of time data means that the original coordinates of the object at this time node have not changed. During data processing, the longitude and latitude of adjacent nodes are used for filling. Similarly, the

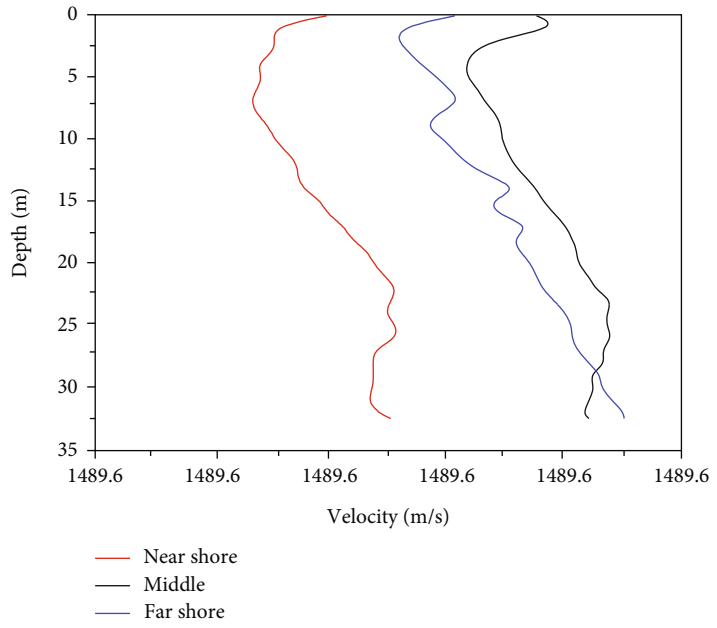


Figure 6: Sound velocity variation curve at near shore release point, middle release point, and far shore release point during descent.

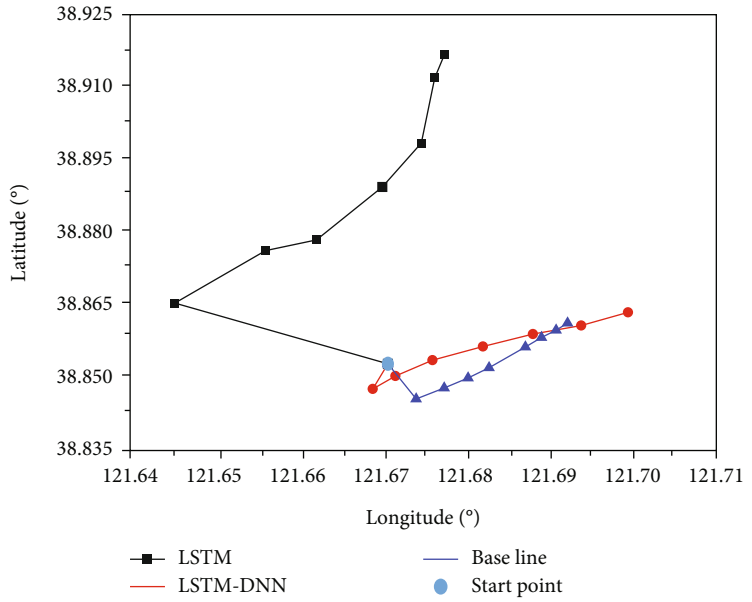


Figure 7: Comparison diagram of observed trajectory and predicted future trajectory based on LSTM and LSTM-DNN.

drifting track motion data of the same dimension are collected and preprocessed accordingly.

The wind field information and temperature field information of each release point are collected according to the drifting track, and the environmental field data are recorded every ten minutes. Some data are shown in Tables 2 and 3.

In Table 3, the main parameters Vbatt represent voltage, press represents depth, temp represents temperature, sound represents sound velocity, and sigma represents salinity. During the temperature field acquisition, the equipment detected the depth data from the sea surface to 32 meters below the sea water and detected the environmental information of the five release points. Because the temperature field in the near sea area changes very little, it mainly analyzes the data information of the three positions, namely, the near coast release point, the middle release point, and the far coast release point. It mainly analyzes the change of temperature with depth and the change of sound velocity with depth during the decline process, as shown in Figures 5 and 6, respectively.

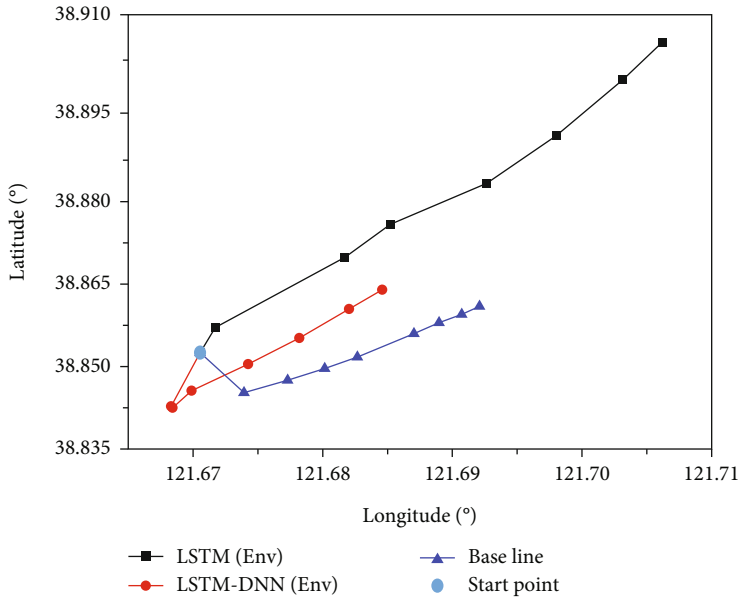


Figure 8: Comparison diagram of observed trajectory and predicted future trajectory based on LSTM and LSTM-DNN under the same environmental field.

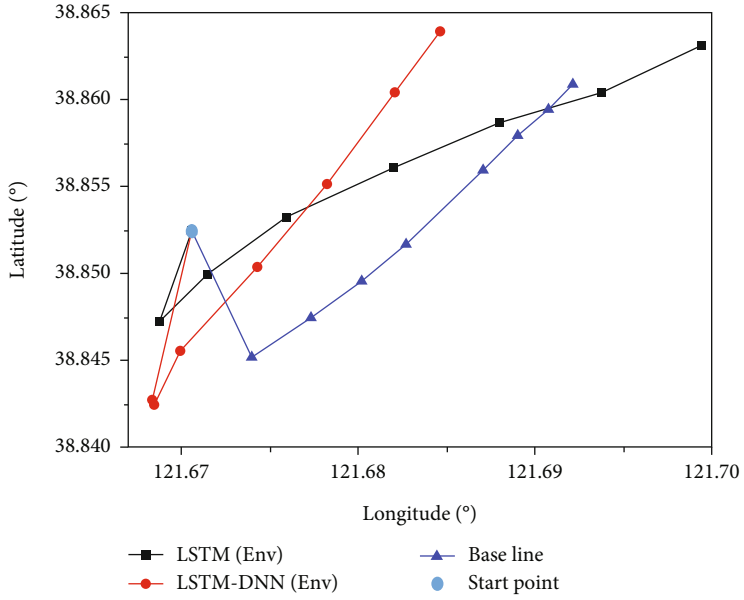


Figure 9: Comparison of LSTM-DNN observed orbits and predicted future orbits under the same environmental field.

According to the data processing and analysis, it can be determined that the temperature is approximately isothermal layer with depth, and the sound velocity is approximately weak positive gradient with depth. Therefore, in the preprocessing of temperature data, the temperature field can be approximately the mean value in the depth direction of the point, and the change of sound velocity can be ignored.

Collect the test data, mainly including the time stamp, latitude, longitude, speed, direction, target attribute, temperature field, wind field, and other parameters of each track, and save them in the LSTM-DNN model.

4.4. Performance of LSTM-DNN Predicting Model

(1) Influence of ocean current motion field on drifting trajectory

In order to explore the influence of ocean current trajectory on drifting trajectory, only ocean current data is added

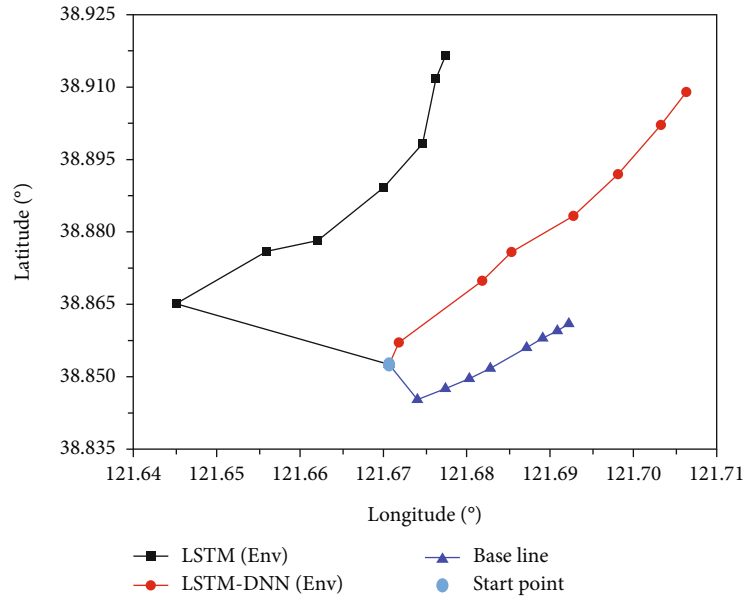


Figure 10: Comparison between LSTM observation orbit and predicted future orbit under the same environmental field.

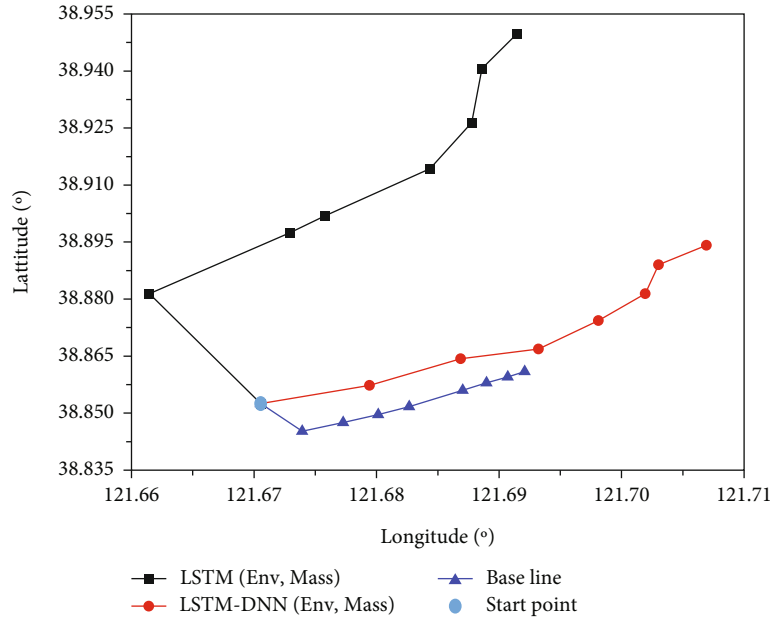


Figure 11: Comparison diagram of observed trajectory and predicted future trajectory based on LSTM and LSTM-DNN under the same target attributes.

during training analysis. The traditional LSTM model and LSTM-DNN model mentioned in this paper are trained, respectively, to model and predict the test data. The test results are shown in Figure 7.

The dot in Figure 7 is the initial position of the drifting track, the red track represents the predicted track based on the LSTM-DNN model, the black path represents the predicted track of the traditional LSTM model, and the blue track is the real acquisition path. It can be seen that under the same environment, the prediction result of LSTM-DNN is significantly better than that of LSTM model, in which the root mean squared error (RMSE) calculation result of LSTM-DNN model is 0.0332, and the RMSE result of LSTM is 0.0357. Therefore, LSTM-DNN has higher prediction accuracy.

(2) Analysis on the influence of wind field and temperature field on drifting trajectory

In order to explore the influence of wind field and temperature field on drifting trajectory, the preprocessed ocean current trajectory data in (1) is applied, the environmental

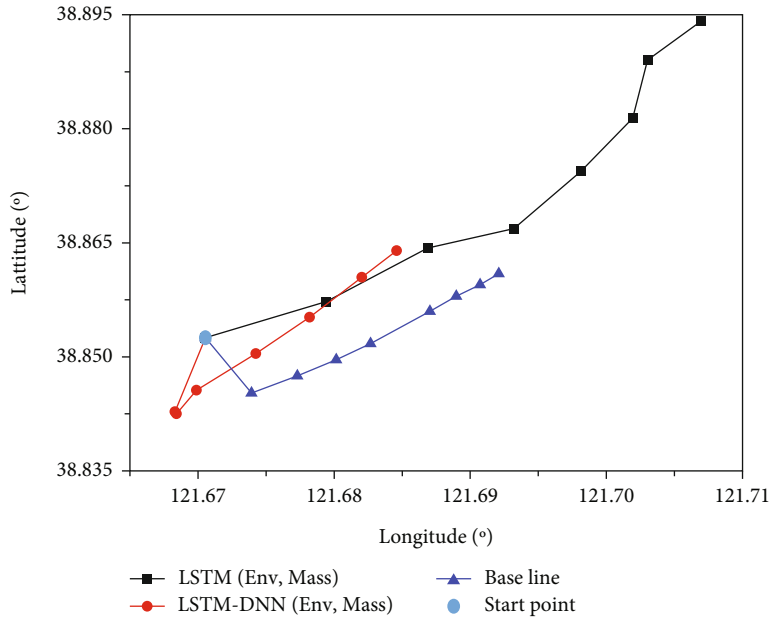


Figure 12: Comparison between LSTM-DNN observation orbit and predicted future orbit under the same target attributes.

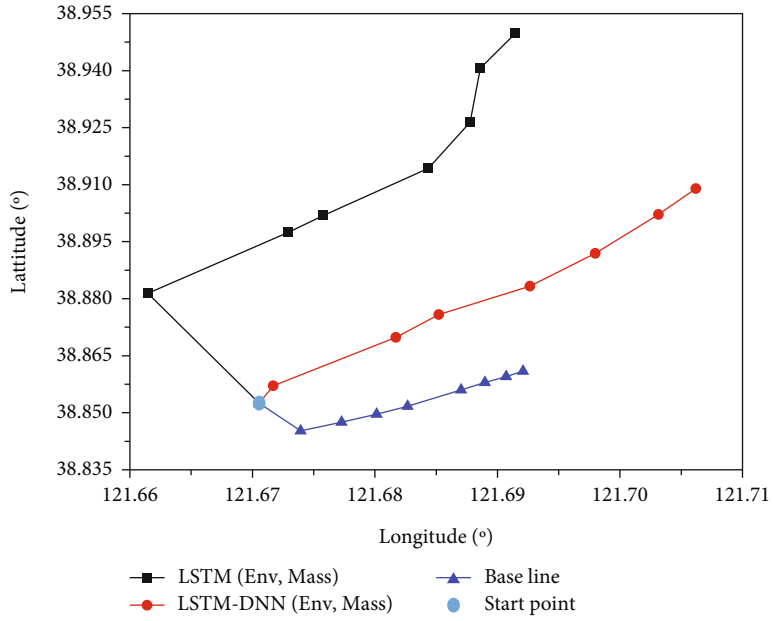


Figure 13: Comparison between LSTM observation orbit and predicted future orbit under the same target attributes.

field data is newly loaded into the training data, the LSTM model and LSTM-DNN model are further trained to predict the same set of test data, and the prediction results of the two models and the results of each model and (1) are analyzed and compared, respectively, as shown in Figure 8.

The dot in Figure 8 is the initial position of the drifting track, the red track represents the predicted track after adding the environmental field to the training data based on the LSTM-DNN model, the black path represents the predicted track after adding the environmental field to the training data of the traditional LSTM model, and the blue track is the real acquisition path. It can be seen that under the same environment, the prediction result of LSTM-DNN model is significantly better than that of the LSTM model. The RMSE calculation result of LSTM-DNN model is 0.0256, and the RMSE result of LSTM model is 0.0339. LSTM-DNN model has higher prediction accuracy.

Figures 9 and 10 show the analysis and comparison between the LSTM-DNN model, the LSTM model, and the original data after adding the wind field and temperature field environment, respectively. It can be seen that the prediction accuracy of both models is improved. Therefore,

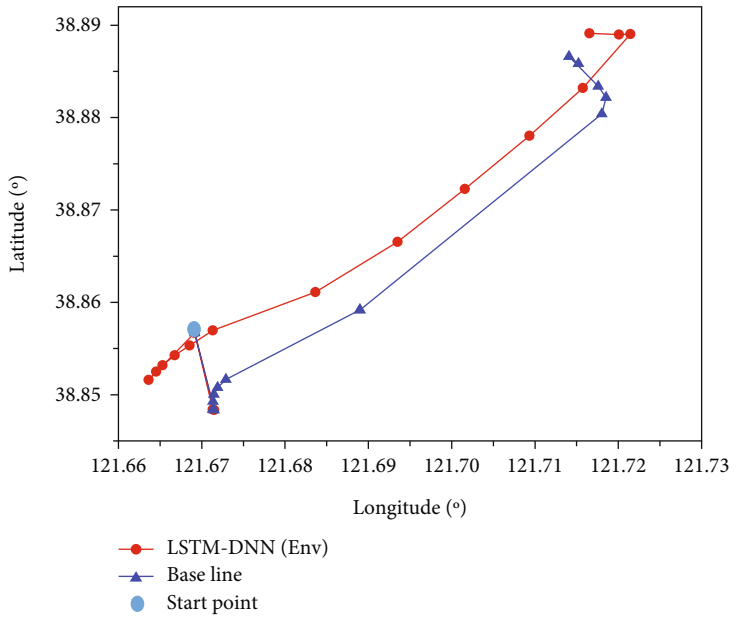


Figure 14: Comparison between observed orbit and predicted future orbit under 6 hours of LSTM-DNN model.

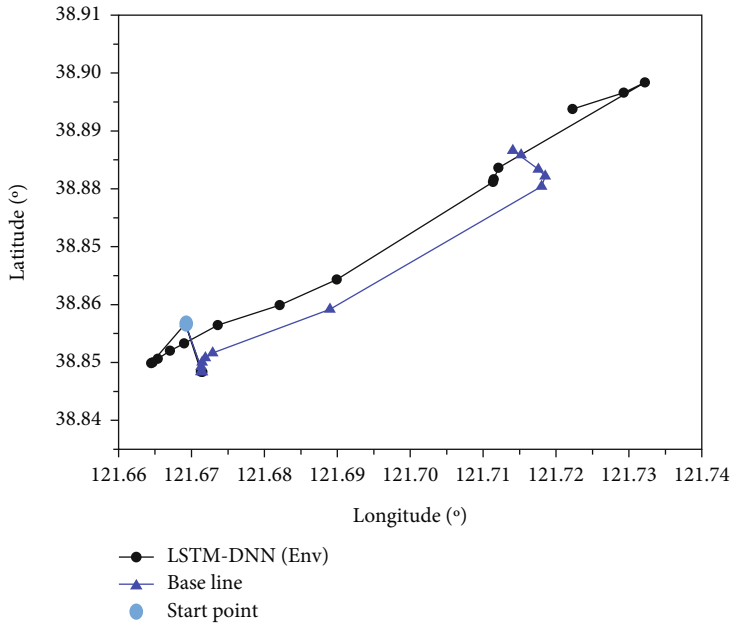


Figure 15: Comparison between observed orbit and predicted future orbit under 6 hours of LSTM model.

determining the prediction results of wind field and temperature field on drifting trajectory can improve its accuracy.

(3) Influence of target attributes on drifting trajectory

Under the same data, explore the impact of target object attributes on drifting trajectory, and set target attributes for each group of drifting objects, mainly shape, volume, and weight. The shape is converted into data format by one hot coding, and LSTM model and LSTM-DNN model are trained. The analysis and comparison prediction results are shown in Figure 11.

The dot marked as the start point in Figure 11 is the initial position of the drifting track. The red track represents the predicted track after adding the environmental field and target attribute in the training data based on the LSTM-DNN model. The black path represents the predicted track after adding the environmental field and target attribute in the training data of the traditional LSTM model, and the blue track is the real acquisition path. Under the same data, the RMSE calculation result of LSTM-DNN model is 0.0325. The RMSE result of LSTM model is 0.0343, and the prediction accuracy of the two models is lower than that of the environmental field. Therefore, the

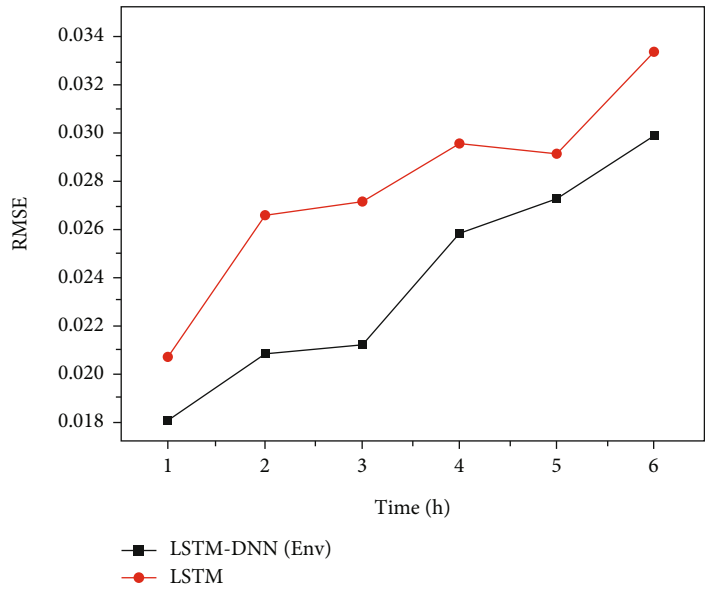


Figure 16: Comparison between LSTM-DNN and LSTM 6-hour prediction RMSE.

accuracy of the target's own attributes for drifting trajectory prediction will show a weak downward trend.

Figures 12 and 13 show the prediction results of the two models after adding the target attribute. By comparing the prediction accuracy, the accuracy of the prediction trajectory shows a weak downward trend after adding the target attribute on the basis of the environmental field. This is because the floating objects on the sea surface are mainly affected by environmental factors such as wind field and ocean current trajectory. Meanwhile, adding too many self attributes will increase the redundant characteristics of the model.

(4) Influence of prediction duration on prediction results of drifting trajectory

In this section, comparison between observed orbit and predicted future orbit under 6 hours base of the LSTM-DNN model and LSTM model is given. The tested data is in the same sea area, the first six groups of data are intercepted as the observation path trajectory, and the two models are used to predict the subsequent drift path. The results are shown in Figure 14 (LSTM-DNN model) and Figure 15 (LSTM model), respectively.

Figure 16 shows the comparison of LSTM-DNN model and LSTM model with the real path in 6 hours. The RMSE corresponding to 1st hour to 6th hour is calculated according to the predicted path, as shown in Figure 16. It can be observed that the prediction results of LSTM-DNN model in 6 hours are more accurate than LSTM model.

5. Conclusions

The marine drifting trajectory prediction method proposed in this paper modeled the environmental field based on LSTM-DNN. It can be used to predict the environmental information matrix at a point in the future and further transfer the environmental information matrix and the path information of the drifting trajectory into the DNN network; then, the drifting trajectory at a certain time in the future can be predicted. In this paper, the influence of ocean current motion field, environmental fields, and target attributes on the drifting trajectory is discussed in detail. Numerical results show that ocean current motion field and environmental fields have a relatively more obvious impact on the predicted results. The proposed model can provide a very accurate trajectory prediction in up to 6 hours. The proposed algorithm has a significant effect on short-term maritime trajectory prediction and will lead an important guide in the marine search and rescue work.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare no conflict of interest.

Authors' Contributions

Conceptualization was done by Xianbin Li and Kai Wang; software was done by Xianbin Li and Kai Wang; writing—original draft preparation was done by Xianbin Li, Kai Wang, Min Tang, Jiangyi Qin, and Peng Wu; writing—review and editing was done by Min Tang, Kai Wang, Tingting Yang, and Haichao Zhang. All authors have read and agreed to the published version of the manuscript.

References

[1] E. Akyuz, "A marine accident analysing model to evaluate potential operational causes in cargo ships," *Safety Science*, vol. 92, pp. 17–25, 2017.

[2] J. Kwesi-Buor, D. A. Menachof, and R. Talas, "Scenario analysis and disaster preparedness for port and maritime logistics risk management," *Accident; Analysis and Prevention*, vol. 123, pp. 433–447, 2019.

[3] A. Ee, B. Ap, and A. Mv, "Statistical analysis of ship accidents and review of safety level," *Safety Science*, vol. 85, pp. 282–292, 2016.

[4] A. M. Zhang and H. G. Sui, "An intelligent marine search and rescue directing system," *Proceedings of SPIE - The International Society for Optical Engineering*, vol. 6421, 2006.

[5] Y. Yang, Y. Mao, R. Xie, Y. Hu, and Y. Nan, "A novel optimal route planning algorithm for searching on the sea," *Aeronautical Journal*, vol. 125, no. 1288, pp. 1064–1082, 2021.

[6] A. Rudenko, L. Palmieri, M. Herman, K. M. Kitani, D. M. Gavrila, and K. O. Arras, "Human motion trajectory prediction: a survey," *International Journal of Robotics Research*, vol. 39, no. 8, pp. 895–935, 2020.

[7] P. Lv, H. Wei, T. Gu et al., "Trajectory distributions: a new description of movement for trajectory prediction," *Computational visual media*, vol. 8, no. 2, pp. 213–224, 2022.

[8] J. Yan, Z. Peng, H. Yin et al., "Trajectory prediction for intelligent vehicles using spatial-attention mechanism," *IET Intelligent Transport Systems*, vol. 14, no. 13, pp. 1855–1863, 2020.

[9] F. Large, D. Vasquez, T. Fraichard, and C. Laugier, "Avoiding cars and pedestrians using velocity obstacles and motion prediction," *IEEE Intelligent Vehicles Symposium, IEEE*, pp. 375–379, 2004.

[10] L. Liu, M. Zhao, M. Yu, M. A. Jan, D. Lan, and A. Taherkordi, "Mobility-aware multi-hop task offloading for autonomous driving in vehicular edge computing and networks," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–14, 2022.

[11] J. Feng, L. Liu, Q. Pei, and K. Li, "Min-max cost optimization for efficient hierarchical federated learning in wireless edge networks," *IEEE Transactions on Parallel and Distributed Systems*, p. 1, 2022.

[12] S. Thompson, T. Horiuchi, and S. Kagami, "A probabilistic model of human motion and navigation intent for mobile robot path planning," in *2009 4th International Conference on Autonomous Robots and Agents*, pp. 663–668, IEEE, 2009.

[13] M. Luber, J. A. Stork, G. D. Tipaldi, and K. O. Arras, "People tracking with human motion predictions from social forces," in *2010 IEEE International Conference on Robotics and Automation*, pp. 464–469, IEEE, 2010.

[14] B. Musleh, F. García, J. Otamendi, J. M. Armingol, and A. De la Escalera, "Identifying and tracking pedestrians based on sensor fusion and motion stability predictions," *Sensors*, vol. 10, no. 9, pp. 8028–8053, 2010.

[15] E. Mohammed, S. Wang, and J. Yu, "Ultra-short-term wind power prediction using a hybrid model," *Science*, vol. 63, p. 012005, 2017.

[16] Q. Wu, F. Guan, C. Lv, and Y. Huang, "Ultra-short-term multi-step wind power forecasting based on CNN-LSTM," *IET Renewable Power Generation*, vol. 15, no. 5, pp. 1019–1029, 2021.

[17] J. Wu, X. Yang, H. Wang, J. Chen, Q. Zhao, and B. Xiao, *Study on Prediction Model of Magnetic Field Intensity of Submarine Power Cable Based on LSTM*, Pervasive Health: Pervasive Computing Technologies for Healthcare, 2020.

[18] S. N. Khotimah and S. Viridi, "Influence of initial velocity on trajectories of a charged particle in uniform crossed electric and magnetic fields," *European Journal of Physics*, vol. 38, no. 2, p. 025204, 2017.

[19] W. Luo and G. Zhang, "Ship motion trajectory and prediction based on vector analysis," *Journal of Coastal Research*, vol. 95, no. sp1, pp. 1183–1188, 2020.

[20] C. Tang, M. Chen, J. Zhao et al., "A novel ship trajectory clustering method for finding overall and local features of ship trajectories," *Ocean Engineering*, vol. 241, p. 110108, 2021.

[21] J. Liu, T. Zhang, G. Han, and Y. Gou, "TD-LSTM: temporal dependence-based LSTM networks for marine temperature prediction," *Sensors*, vol. 18, no. 11, p. 3797, 2018.

[22] G. Cruz and A. Bernardino, "Learning temporal features for detection on maritime airborne video sequences using convolutional LSTM," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 9, pp. 6565–6576, 2019.

[23] H. M. Choi, M. K. Kim, and H. Yang, "Abnormally high water temperature prediction using LSTM deep learning model," *Journal of Intelligent Fuzzy Systems*, vol. 40, no. 4, pp. 8013–8020, 2021.

[24] G. Tang, J. Lei, C. Shao, X. Hu, W. Cao, and S. Men, "Short-term prediction in vessel heave motion based on improved LSTM model," *IEEE Access*, vol. 9, pp. 58067–58078, 2021.

[25] L. Zhao and G. Shi, "Maritime anomaly detection using density-based clustering and recurrent neural network," *Journal of Navigation*, vol. 72, no. 4, pp. 894–916, 2019.

[26] J. Park, J. S. Jeong, and Y. S. Park, "Ship trajectory prediction based on bi-LSTM using spectral-clustered AIS data," *Journal of marine science and engineering*, vol. 9, no. 9, p. 1037, 2021.

[27] D. W. Gao, Y. S. Zhu, J. F. Zhang, Y. K. He, K. Yan, and B. R. Yan, "A novel MP-LSTM method for ship trajectory prediction based on AIS data," *Ocean Engineering*, vol. 228, p. 108956, 2021.

[28] Y. Heng, Y. Jian-Ping, and Xing, "Study on dam deformation prediction based on deep fully connected neural network [J]," *Geodesy and Geodynamics*, vol. 41, no. 2, pp. 162–166, 2021.

[29] K. Zhu, L. Mu, and X. Xia, "An ensemble trajectory prediction model for maritime search and rescue and oil spill based on sub-grid velocity model," *Ocean Engineering*, vol. 236, p. 109513, 2021.

[30] S. Z. Wang, H. B. Nie, and C. J. Shi, "A drifting trajectory prediction model based on object shape and stochastic motion features," *Journal of Hydrodynamics*, vol. 26, no. 6, pp. 951–959, 2014.

[31] H. Blanken, C. Valeo, C. G. Hannah, U. T. Khan, and T. Juhász, "A fuzzy-based framework for assessing uncertainty in drift prediction using observed currents and winds," *FMARS*, vol. 8, 2021.

[32] K. Jitkajornwanich, P. Vateekul, U. Gupta et al., "Ocean surface current prediction based on HF radar observations using trajectory-oriented association rule mining," in *2017 IEEE international conrerence on big data (big data)*, pp. 4293–4300, Boston, MA, 2017.

[33] E. Y. Shchekinova, Y. Kumkar, and G. Coppini, "Numerical reconstruction of trajectory of small-size surface drifter in the Mediterranean Sea," *Ocean Dynamics*, vol. 66, no. 2, pp. 153–161, 2016.

Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 3469221, 19 pages
https://doi.org/10.1155/2022/3469221

Research Article

Data Collection Method of Energy Adaptive Distributed Wireless Sensor Networks Based on UAV

Bo Yang,[1,2] Xiangyu Bai,[1,2] and Changxing Zhang[1,2]

[1]*College of Computer Science, Inner Mongolia University, Hohhot 010021, China*
[2]*Inner Mongolia Key Laboratory of Wireless Networking and Mobile Computing, Inner Mongolia University, Hohhot 010021, China*

Correspondence should be addressed to Xiangyu Bai; bxy@imu.edu.cn

Received 16 December 2021; Revised 7 April 2022; Accepted 26 April 2022; Published 28 June 2022

Academic Editor: Celimuge Wu

In sensor networks, UAVs are often introduced to assist data collection tasks. UAVs can operate as data ferry nodes, connecting distributed areas that are separated from each other. This paper proposes a data collection method for distributed wireless sensor networks based on UAV and introduces the idea of edge computing in it. In the single-hop transmission scenario, the K-means++ clustering method is used for sensor node clustering and cluster head election in the initial state. In the next rounds of data collection, UAV is used to assist in the election of new cluster heads and data collection tasks, taking into account the relative distance and the relative remaining energy relationship of the sensor nodes in their clusters. In addition, reasonable priorities are set for some nodes that have never been elected in the previous rounds and for the dead nodes. In the multihop transmission scenarios, for nodes that cannot deliver directly, the optimal relay node is selected for routing by comprehensively considering factors such as transmission angle, transmission distance, and remaining energy of the node in each cluster. The method proposed in this paper coordinates the overall energy consumption of sensor nodes in the environmental monitoring area, delays the death time of key sensor nodes, and extends the network lifetime. At the same time, an improved ACO is used to reasonably plan the data collection path of the UAV. Compared with the comparison scheme, the improved ACO can obtain a better shortest path length and has the fastest convergence speed when reaching the shortest path.

1. Introduction

In recent years, with the rapid development of wireless networks and the technical advantages of wireless sensor networks, such as self-organization, rapid deployment, high error tolerance, and low cost, sensor networks are often used to monitor and collect ecological environment data, collect information in the process of geological monitoring, and sense some areas that are not suitable for human beings to stay and live for a long time. However, due to the influence of the volume of sensor nodes and the difficulty of replacing batteries of a large number of sensor nodes, how to reduce energy consumption when collecting data is an important problem. After a large number of sensor nodes are deployed, how to effectively collect sensor network data and how to effectively prolong the service life of sensor nodes in the process of sensor network data collection have become hot issues in academic and industrial circles in recent years.

For sensor data monitoring and collection tasks in special environments, multiple sensor data collection areas may be separated from each other due to geographical environmental factors such as rivers, mountains, and swamps. Deploying relay nodes in these areas will consume a lot of manpower and material resources and is very inconvenient to implement. The use of UAVs can effectively solve the impact of ground environmental factors on data collection tasks. With the increasing maturity of UAV technology, UAVs have been widely used in vehicle networking, agriculture, military reconnaissance, and other fields, and civil UAVs are also gradually popularized.

By introducing UAVs into wireless sensor networks, it can form a delay-tolerant network with traditional sensor networks, and effectively assist sensor networks in data

collection tasks by using the "store-carry-forward" method. The UAV serves as a relay to collect data from sensor networks in remote areas and transmit it to the data center for processing, thus avoiding manual data collection and effectively addressing the impact of environmental factors on data collection on the ground.

UAVs assist wireless sensor networks where the UAV can move over the network, retrieve and collect data from sensor nodes. Using effective routing protocols can reduce energy consumption, avoid long-distance transmission and redundant transmission, and prolong the service life of sensors.

We have previously studied data collection by UAV-assisted wireless sensor networks. In reference [1], we mainly proposed a data collection strategy based on drone technology in wireless sensor networks, using K-means++ method to conduct clustering and cluster head election in the initial state. Then, on the basis of comprehensive consideration of relative distance and relative residual energy of each sensor node, UAV is used to assist cluster head election and data collection. In addition, for some unelected nodes, a reasonable priority is set to make the energy consumption of sensor nodes more balanced. Experiments show that this strategy reduces the energy consumption and improves the performance of sensor networks. On the basis of the previous work, this paper further expands and utilizes UAV-related technology to assist data collection from multiple distributed sensor network areas separated from each other due to geographical environmental factors, so as to make the energy consumption of sensor nodes in the monitoring area more balanced. The main contributions of this paper are as follows.

Introducing the idea of edge computing into the data collection process of distributed sensor network can greatly improve the efficiency of data collection. We discuss the case that the sensor node transmits the environmental data to the cluster head node through single-hop or multihop transmission mode. This method is not only flexible but also has low cost, which solves the problem that the data collection task of sensor network cannot be carried out uniformly in the environment monitoring area under complex geographical conditions.

Due to the limited battery capacity and high energy consumption of UAV, an efficient and energy-saving routing protocol is needed in both military and commercial applications. In addition, the deployment and trajectory planning of UAVs have a significant impact on the performance of routing protocols. Therefore, an improved ACO is used in this paper to plan the UAV's path, and the "store-carry-forward" method is adopted to collect sensor data from each cluster head node. In this way, the data is transmitted to the data center with a small path cost and time cost.

The remaining chapters of this paper are organized as follows. The second chapter mainly introduces the related work, including the data collection method of the sensor network and the typical UAV path planning method. The third chapter mainly introduces the system model. The fourth chapter mainly introduces the data collection of UAV-based sensor network. The fifth chapter mainly introduces the UAV path planning based on improved ACO. The sixth chapter mainly introduces the simulation experiment and result analysis. The seventh chapter mainly introduces the conclusion and outlook.

2. Related Work

2.1. Data Collection Method of Sensor Network. For the data collection scheme of traditional sensor network, the data collection performance of the sensor network can be greatly improved by introducing mobile nodes as assistance, which has been paid attention to by most researchers [2–4]. In recent years, due to the reduction of UAV cost and the rapid development of UAV technology, using UAV as an auxiliary node to assist sensor network in data collection has become a research hotspot [5–8]. Traditional mobile nodes are susceptible to ground path restrictions and often cannot fully utilize the performance of sensor networks. However, the UAV has broken through the node's movement path restriction and has better flexibility in data collection tasks.

Chen et al. [9] proposed a universal NOMA-enabled UAV-assisted data collection protocol to maximize the total rate of wireless sensor networks during the data collection process. Xu et al. [10] introduced blockchain into the UAV-assisted IoT scenario and proposed a data collection system that takes into account both safety and energy efficiency, which can effectively improve the safety and efficiency of data collection. In order to ensure the timeliness of the collected data, Zhu et al. [11] optimized the trajectory and wake-up time allocation of the UAV as well as the transmission power of the sensor nodes to minimize the task completion time. Ma et al. [12] modelled the convergence node, UAV deployment, and resource allocation as a mixed-integer nonconvex optimization problem. They used heuristic methods to effectively solve the problem, thereby prolonging the life of the network. Ebrahimi et al. [13] used UAVs in dense wireless sensor networks to use projection-based compressed data collection (CDG) as a novel solution to collect data. Du et al. [14] used UAVs to vehicle tolerance delay network (VDTN) for message storage and forwarding and proposed a VDTN routing protocol based on UAV, which considered both the probability of each encounter and the duration of connection between mobile nodes. This method not only reduced network overhead and end-to-end delay but also improved the reliability of message forwarding.

In summary, the existing data collection methods of sensor network rarely consider the problem of distributed wireless sensor networks composed of multiple isolated monitoring areas. In this network environment, how to collect data from the sensor network to reduce the overall energy consumption of the network and how to plan the UAV to optimize the data collection path of the separated monitoring area are problems worthy of study. These are also the focus of this paper.

2.2. Typical UAV Path Planning Method. Typical UAV path planning methods are mainly divided into two categories: classic UAV path planning methods and UAV path planning methods based on intelligent algorithms. The classic

UAV path planning methods mainly include A-star algorithm, cell decomposition method, and artificial potential field method. UAV path planning methods based on intelligent algorithms mainly include genetic algorithm, particle swam optimization, and wolf colony algorithm.

The A-star algorithm is a direct search algorithm for planning the shortest flight path of UAVs in static state. At the same time, the A-star algorithm is one of the heuristic algorithms [15]. Its basic algorithm idea is to use the heuristic function to evaluate some candidate nodes and select the node with the best condition as the next node on the path. This is a purposeful search method that effectively avoids blind searches. The A-star algorithm can achieve faster calculation speed when the path matrix is small and efficiently obtain the UAV path information that needs path planning. However, when the number of paths increases sharply, its running time also increases accordingly, which is not suitable for path planning problems in dynamic states.

The artificial potential field path planning is a method that uses virtual forces in the environment to assist path planning. The basic idea of this algorithm is to abstract the movement process of the UAV in the environment as a movement process of the UAV under the virtual artificial potential field. The target point to be reached by the UAV is a gravitational field for it, and the obstacles in the path are a repulsive field for it. Under the combined action of the gravitational field and the repulsive field, the UAV starts from the starting point, avoids obstacles, and finally reaches the destination node. Generally speaking, the path generated by using the artificial potential field to plan the UAV's path is smooth and safe, but this method has the problem of generating local optimal solutions, and there are certain human factors in the design of the repulsion field and the gravitational field. When there are obstacles near the target node, the UAV may not be able to reach the target node, which also limits the development of artificial potential field.

Particle swarm optimization is a kind of evolutionary algorithm. It simulates a predation behaviour of a flock of birds randomly searching for food, without requiring any leader. In Particle swarm optimization, the potential solution of each UAV optimization problem can be abstracted as a "particle" in the search space, that is, a bird. "Particles" follow the current optimal "particles" to search in the solution space. These "particles" are initially some random solutions, and the optimal solution is found after many iterations of optimization. Particle swarm optimization is widely used in the field of UAV path planning. It has good convergence and path optimization capabilities and is suitable for the optimization of continuous problems. However, the algorithm may get trapped in local optimal solutions and cannot handle optimization problems in discrete cases well.

Wolf colony algorithm is an algorithm based on the swarm intelligence of wolves. The algorithm simulates the predation behaviour of wolves and how wolves distribute their prey. The main body of the wolf colony algorithm is composed of three intelligent behaviours: wandering, calling, and besieging. The algorithm's method of generating the head wolf uses the "winner is king" rule, while the algorithm's method of updating the wolf colony uses the "stronger survival" mechanism. In solving the problem of UAV path planning, the wolf colony algorithm improves the probability of obtaining the optimal solution of UAV path planning in a limited time to a certain extent and reduces the understanding space. Although it can deal with simple UAV path planning problems and realize UAV path planning in a continuous environment, it cannot realize path planning in a discrete environment, and the iteration convergence speed is slow.

Reinforcement learning is a new algorithm based on learning. In recent years, reinforcement learning has been widely used. Chen et al. [16] applied reinforcement learning to the Internet of Vehicles and proposed an online deep reinforcement learning scheme. Each mobile user only made use of local information to make decisions such as channel auction, computational task unloading, and input packet scheduling, so as to optimize task unloading of the air-ground integrated multiaccess edge computing (MEC) system. Reinforcement learning has also been applied in path planning. In order to reduce communication delay between vehicles, Wu et al. [17] proposed a multichannel vehicle edge computing routing scheme based on cooperative learning to solve the communication path selection problem in multichannel vehicle environment. Tong et al. [18] modelled the UAV-assisted data collection problem as a limited range Markov decision process with limited state and action space and developed a deep reinforcement learning algorithm to find the asymptotically optimal strategy. The introduction of reinforcement learning breaks the previous idea of using intelligent algorithm to optimize the path and provides a new method for path planning. Therefore, we can consider future work and design reinforcement learning algorithm for our own application scenarios to optimize the flight path of UAV.

3. System Model

Edge computing refers to an open platform that integrates network, computing, storage, and application core capabilities to provide the nearest end service on one side of the object or data source. Edge computing provides faster services and is used to enforce business, security, and privacy. Introducing the idea of edge computing in the data collection process of distributed sensor networks can greatly improve the efficiency of data collection.

In a distributed sensor network environmental data monitoring and collection task which has multiple partitioned monitoring areas, multiple sensor nodes in each monitoring area are responsible for environmental data sensing. The sensor node transmits the environmental data to the cluster head node through single-hop transmission or multihop transmission. After that, the cluster head node aggregates and fuses the data of each sensor node. Starting from the data center, the UAV uses a "store-carry-forward" method according to a certain path planning method to collect data from the cluster head nodes in each partitioned area. The UAV returns the collected data to the data center for transmission and processing. The application scenario of sensor network data collection system is shown in Figure 1. The entities in the system and their functions are described as follows:

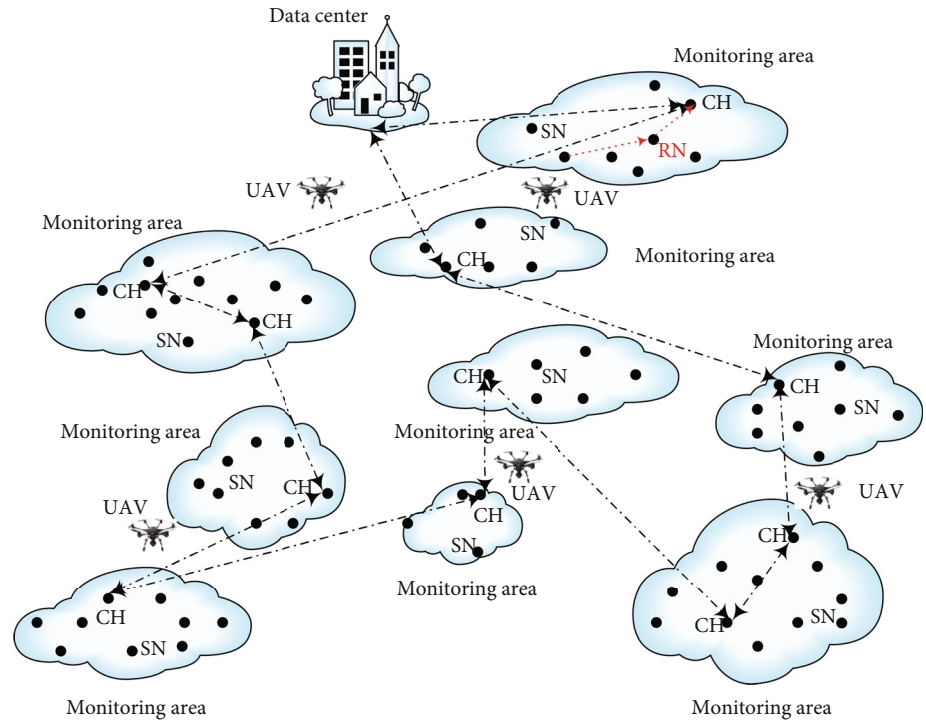


Figure 1: System application scenario.

Sensor Node (SN). The sensor node is used to collect the sensing data in the environmental monitoring area, such as temperature, humidity, light intensity, and the acid, alkali, and salt concentration in the soil, etc. The sensor node periodically sends the sensed environmental data to the cluster head node and at the same time transmits its own remaining energy information to the cluster head node.

Relay Node (RN). Due to the limited communication distance of the sensor node, when the Euclidean distance between the sensor node and the cluster head node is greater than the communication distance, the relay node is responsible for relaying the transmitted data. The relay node can replace an unreachable link with multiple links of better quality to obtain better network coverage.

Cluster Head Node (Cluster Head, CH). The cluster head node converges and fuses the data information collected by the sensors in the cluster. At the same time, the cluster head node is responsible for interacting with the UAV, transmitting the collected sensing data to the UAV, and receiving control information from the UAV.

Data Center. The data center gathers the environmental monitoring data collected by UAV from each partitioned monitoring area and performs data processing.

By delegating computing and processing tasks to sensor nodes and UAV nodes, many controls will be achieved through local equipment without having to hand over to the data center in the cloud. The processing process is completed at the local edge layer, which greatly improves processing efficiency. This way, because it is closer to the terminal, the demand can be solved at the edge.

The whole system model can be divided into two parts: data collection of sensor network based on UAV and UAV path planning based on improved ACO. Between them, the data collection of UAV-based sensor network is divided into two cases: single-hop transmission and multihop transmission, according to the relationship between communication distance and transmission distance.

The system model is based on the following premise assumptions. (1) The position coordinates of each sensor node are fixed and the initial energy is known. (2) The UAV stores the mapping relationship information between the identity document (ID) of the sensor node and its location coordinates. (3) The sensor nodes can adjust the signal transmission power adaptively according to the distance of the nodes. (4) The communication channel is a symmetric propagation channel. (5) Issues such as communication security and encryption will not be considered for the time being.

4. Data Collection of UAV-Based Sensor Network

4.1. Sensor Clustering and Cluster Head Election in the Initial State. Abstract each area in the partitioned environmental monitoring area. Assume that each monitoring area is a rectangle with length A and width B. N sensor nodes are randomly distributed, as shown in Figure 2. The communication distance of each sensor node is R. The sensor node transmits the sensed data to the cluster head node of its own cluster. The cluster head node performs data fusion on the collected

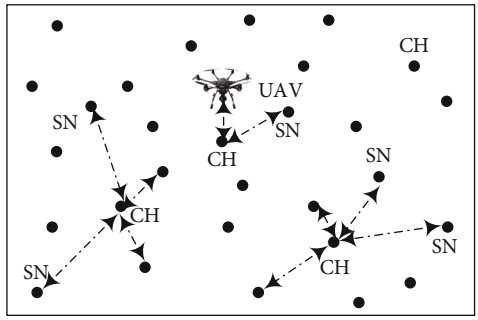


Figure 2: Abstract of a single monitoring area.

sensed data, waits for the arrival of the UAV, and interacts with the UAV. Then, the data is transferred to the data center.

For each separated environmental monitoring area, when the coverage of the monitoring area exceeds the communication range of the sensor node, two ways can be used to realize the communication between sensor nodes. (1) Divide a large area so that the range of each divided area will not exceed the communication range of sensor nodes, thereby ensuring single-hop transmission between sensor nodes. (2) Sensor nodes use multihop transmission to transmit data to other nodes through relay nodes.

First discuss the first case, that is, clustering a larger network, so that each sensor node can communicate with all other nodes in the cluster. The cluster number Num of sensor nodes is calculated according to Equations (1) and (2). Num takes an integer value in the interval $[Num1, Num2]$.

$$Num1 = \frac{Area}{\pi \times (R/2)^2}, \tag{1}$$

$$Num2 = \frac{2 \times Area}{R^2}, \tag{2}$$

where $Num1$ is the smallest value of the number of sensor node clusters in the environmental monitoring area, $Num2$ is the largest value of the number of sensor node clusters in the environmental monitoring area, $Area$ is the area of the environmental monitoring area, and R is the communication distance of the sensor nodes. The environmental monitoring area can be of any shape. In order to simplify the description, a regular shape is selected for illustration. The basic idea of calculating the number of clusters of sensor nodes in a regular shape area or an irregular shape area is the same.

Because the main factor affecting the energy consumption of sensor nodes is transmission distance, the distance is evaluated as a similarity. Compared to the K-means algorithm, the K-means++ algorithm significantly reduces the error of classification results. So, we use the K-means++ algorithm to cluster the nodes in the sensor network.

In the initial state, by using the K-means++ algorithm, the clustering of randomly distributed sensor nodes in the environmental monitoring area is completed, and the initial cluster head nodes are elected. The initial cluster head node broadcasts its own node information in the cluster and uses time division multiple access (TDMA) [19] to allocate transmission time slots for the data collection of other sensor nodes in the cluster. During each round of data collection, the normal sensor nodes in each cluster use allocated time slots to transmit the sensing environmental data to the cluster head node and declare their remaining energy information. The data packet contains the remaining energy information of the node and the environmental monitoring data collected by the node in this round. The data packet format is shown in Figure 3. After the cluster head node obtains the remaining energy information of all nodes in the cluster, it forms the remaining energy matrix for the current round of transmission.

4.2. UAV-Assisted Cluster Head Election and Data Collection in a Single-Hop Scenario. In the process of UAV-assisted sensor network data collection, each sensor node is fixed in the environmental monitoring area. Therefore, the relative distance value of each sensor node in the cluster is calculated using

$$Dis[i] = \frac{\sum_{\forall n_j \in C_k} d(n_i, n_j)}{|C_k| \times d_{max}}, \tag{3}$$

where $Dis[i]$ represents the intracluster distance relationship of the i-th node in each cluster, n_i and n_j represent the i-th node and the j-th node, C_k represents the k-th cluster, $d(n_i, n_j)$ represents the Euclidean distance between the current node and all nodes in the cluster, $|C_k|$ represents the total number of the nodes in the k-th cluster, and d_{max} represents the maximum Euclidean distance between the current node and other nodes in the cluster. $d(n_i, n_j)$ is calculated according to

$$d(n_i, n_j) = \sqrt{(x_a - x_b)^2 + (y_a - y_b)^2 + (z_a - z_b)^2}, \tag{4}$$

where the coordinate of node n_i is (x_a, y_a, z_a) and the coordinate of node n_j is (x_b, y_b, z_b).

In an environmental monitoring region, the position of each sensor node is fixed. Then after clustering in an initial state, the cluster of the node is fixed. Therefore, UAV only needs to calculate the relative distance of each sensor node once, which greatly reduces the calculation burden of the UAV.

During each round of data collection, the cluster head node collects and fuses the collection data of each node in the cluster and transmits the remaining energy information of each node in the cluster to the UAV. The UAV calculates the relative residual energy value of each node in each cluster according to

$$Eng[i] = \frac{E_{r,C_k}(n_i)}{E_{r,C_k}(max)}, \tag{5}$$

where $E_{r,C_k}(n_i)$ represents the remaining energy of the i-th node in the k-th cluster during the r-th round of data transmission, $E_{r,C_k}(max)$ represents the maximum remaining

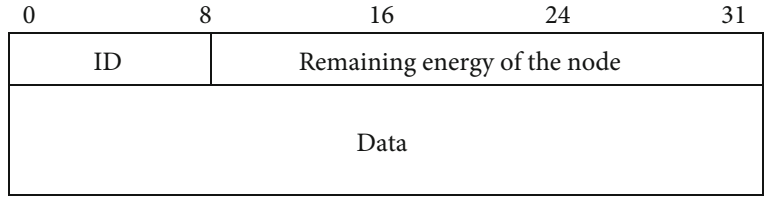


Figure 3: Packet format.

energy of the node in the k-th cluster during the r-th round of data transmission, i is an integer value in the interval $[1, q]$, and q is the total number of nodes in the current cluster.

Since the goal is to select the node with the smallest relative distance from other nodes in the cluster and the largest relative residual energy as possible as the new cluster head node in the next round, therefore, the UAV is used to calculate the priority of each node to be elected as the cluster head in each cluster according to the Equation (6) and save it in $Pri[i]$.

$$Pri[i] = Eng[i] - Dis[i], \tag{6}$$

where i is an integer value in the interval $[1, q]$ and q is the total number of nodes in the current cluster.

If there is a node in the cluster that has never been elected as a cluster head in the previous $1/p$ (p is the proportion of cluster heads to all sensor nodes), its priority is added to the original basis by u ($u \geq 1$, and u is an integer). That is to say, in the next round, a node that has not been elected as a cluster head for a long time and has a relatively small relative distance from other nodes and a relatively large residual energy has a greater probability of becoming a new cluster head node. The equation is described in

$$Pri[i] = Eng[i] - Dis[i] + u. \tag{7}$$

When there is a node with a remaining energy value of 0 in the cluster, the priority value of the node is permanently set to $-u$ ($u \geq 1$, and u is an integer), as shown in

$$Pri[i] = -u. \tag{8}$$

Through the above method, the UAV calculates the priority of each node in the sensor network to be elected as the cluster head in the next round and forms a priority matrix. The node corresponding to the item with the largest element value in the priority matrix becomes the cluster head node elected in the next round. The priority matrix $Total\_Pri$ formed by all clusters in the UAV is shown in

$$Total\_Pri = \begin{bmatrix} Pri_{11} & Pri_{21} & Pri_{31} & \cdots & Pri_{k1} \\ Pri_{12} & Pri_{22} & Pri_{32} & \cdots & Pri_{k2} \\ Pri_{13} & Pri_{23} & Pri_{33} & \cdots & Pri_{k3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ Pri_{1i} & Pri_{2i} & Pri_{3i} & \cdots & Pri_{ki} \end{bmatrix}. \tag{9}$$

Among them, each column in the matrix represents the priority of each node in each cluster to be elected as the cluster head in the next round. There are k columns in total, representing k clusters. For clusters with fewer nodes, add several zeros at the end of the column.

UAV will use the identification number of the cluster head node in the next round as the control information to transmit to the current cluster head node. The current cluster head node broadcasts the information of the cluster head node in the next round in the cluster. The new cluster head node uses TDMA to allocate data transmission time slots for each node in the cluster. During the next round of data collection, the normal node will transmit the remaining energy information and data information to the new cluster head by the allocated time slot.

In this way, the UAV flies to each cluster head node to perform the task of collecting environmental monitoring data information, and the data information collected from each cluster head node is transmitted to the data center through the "store-carry-forward" DTN data transmission mode. The data center analyses, processes, and predicts the collected environmental monitoring data to complete a round of sensor network data collection tasks.

4.3. UAV-Assisted Data Collection in a Multihop Scenario. When the clustering range of each environmental monitoring area is greater than the communication range of the sensor node, that is, for the second case, the nodes outside the communication range need to use multihop transmission, and the communication with the cluster head node is completed by selecting the relay node.

First, use the same method as the previous method to perform the clustering of sensor nodes and the election of cluster heads under the initial conditions. After that, the same UAV-assisted cluster head election method is used to elect the cluster head nodes of each environmental monitoring area in the next round. The UAV uses a "store-carry-forward" approach to collect data from each cluster head node and passes it to the data center for processing. Next, the selection method of the relay node and the corresponding routing process are mainly explained. The model of data transmission using relay nodes is shown in Figure 4.

In the illustrated transmission model, during a round of data collection, nodes are mainly divided into three categories: cluster head nodes, relay nodes, and ordinary sensor nodes. Among them, the cluster head node is responsible for collecting the data sensed by noncluster head nodes in the cluster, performing data fusion, and then interacts with the UAV node. The relay node is selected from the set of candidate relay nodes and is responsible for relaying the data

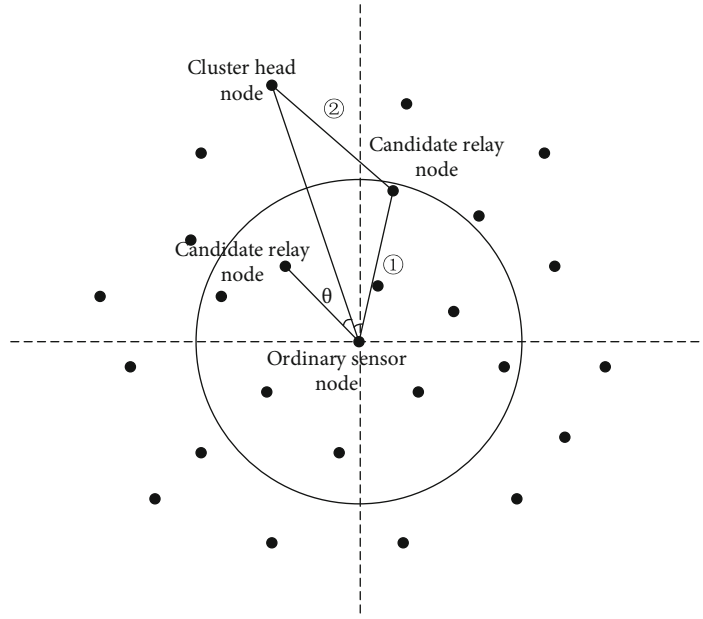


Figure 4: Data transmission model using relay node.

of sensor nodes that cannot directly reach the cluster head node. The relay node also completing its own data sensing task. The ordinary sensor node mainly completes the basic environmental data sensing task. In the above figure, θ is the angle in the middle of the connection between the candidate relay node and the ordinary sensor node and the connection between the ordinary sensor node and the cluster head node.

In the entire sensor network, the positions of all nodes are fixed. Therefore, it is assumed that after the initial state clustering is completed, each sensor node forms a mapping table of the distance between each node in the cluster and other nodes in the cluster.

Since ID and coordinates of each node in each cluster are known and they have a one-to-one correspondence, after the cluster head node broadcasts the next round of cluster head messages, the ordinary sensor node that needs to transmit data first obtains the distance between it and the cluster head node by querying the node distance mapping table. When the distance is less than the communication distance of an ordinary sensor node, the ordinary sensor node transfers the collected data to the cluster head node by direct delivery. When the distance is greater than the communication distance of an ordinary sensor node, the ordinary sensor node selects a relay node according to the method described below.

Assume that the position distribution relationship of cluster head node, candidate relay node, and ordinary sensor node is shown in Figure 5.

According to geometric knowledge, the calculation method of the included angle θ is shown in

$$\theta = cos^{-1} \frac{d_2^2 + d_3^2 - d_1^2}{2 \times d_2 \times d_3}, \tag{10}$$

where $d1$ is the distance between the cluster head node and the candidate relay node, $d2$ is the distance between the candidate relay node and the ordinary sensor node, and $d3$ is the distance between the ordinary sensor node and the cluster head node.

If the ordinary sensor node cannot directly transmit the data to the cluster head node, the nodes within the one-hop communication range of the ordinary sensor node are classified and stored in different sets. The classification method is based on the above-mentioned included angle θ, and nodes meeting different conditions will be divided into different sets.

Since the goal is to transfer data in the direction closer to the cluster head node, for nodes within the communication range of ordinary sensor nodes, nodes within the range of $\theta \in [\pi/2, \pi]$ are stored in the invalid set. The nodes in this set will never be selected. Next, divide the nodes in the range of $\theta \in [0, \pi/2)$ into three sets. The $Pri1$ set stores the nodes in the range of $\theta \in [0, \pi/6)$. The $Pri2$ set stores the nodes in the range of $\theta \in [\pi/6, \pi/3)$. The $Pri3$ set stores the nodes in the range of $\theta \in [\pi/3, \pi/2)$. The priority relationship of the nodes is $Pri1 > Pri2 > Pri3$. Then, calculate the priority of each candidate relay node in the $Pri1$, $Pri2$, and $Pri3$, as shown in

$$Pri_s(i, j) = \lambda \times \frac{(\pi/2) - \theta}{\pi/2} + (1 - \lambda) \times d_{ij}, \tag{11}$$

where $Pri_s(i, j)$ represents the priority function for the i-th ordinary sensor node in the cluster to select the j-th candidate relay node in the range of the set s for relaying. The set s is one of $Pri1$, $Pri2$, and $Pri3$. λ is the weight coefficient, and $\lambda \in (0, 1)$. θ is the angle in the middle of the connection between the candidate relay node and the ordinary sensor node, and the connection between the ordinary sensor node and the cluster head node. The larger the value of $(\pi/2) - \theta$,

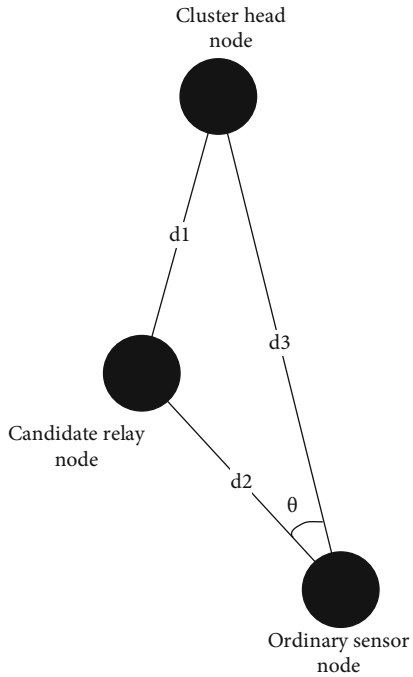


Figure 5: Position distribution relationship of three kinds of nodes.

the closer the position of the candidate relay node is to the direction of the cluster head node within the range of the candidate set. d_{ij} is the Euclidean distance between node i and node j. In order to reduce the number of hops from the ordinary sensor node to the cluster head node as much as possible, data should be transmitted to the candidate relay node closest to the cluster head node and farthest within the communication distance of the node. That is, a candidate relay node with a larger priority function value is more likely to become the final relay node.

Next, perform energy discrimination on the candidate relay node j with the largest priority function value $Pri_s(i, j)$. When the remaining energy of node j is greater than the average remaining energy of all nodes in the set $Pri1$, $Pri2$, and $Pri3$, the candidate relay node becomes the final relay node. Otherwise, the node with the second priority function value is judged. By analogy, the energy discrimination method of nodes is shown in

$$Flag[j] = \frac{E_{residual}[j]}{E_{avg}}, \tag{12}$$

where $Flag[j]$ is the flag bit for whether node j is elected. If its value is greater than 1, node j is elected as the final relay node. $E_{residual}[j]$ is the remaining energy value of the current candidate relay node j, and E_{avg} is the average remaining energy of all nodes in the set $Pri1$, $Pri2$, and $Pri3$. If there is no node in the set $Pri1$ that meets the requirements, the nodes in the set $Pri2$ are judged. If there is no node in the set $Pri2$ that meets the requirements, the nodes in the set $Pri3$ are judged until a suitable relay node is selected. If there is no suitable relay node for relaying, the ordinary sensor node is marked as unreachable.

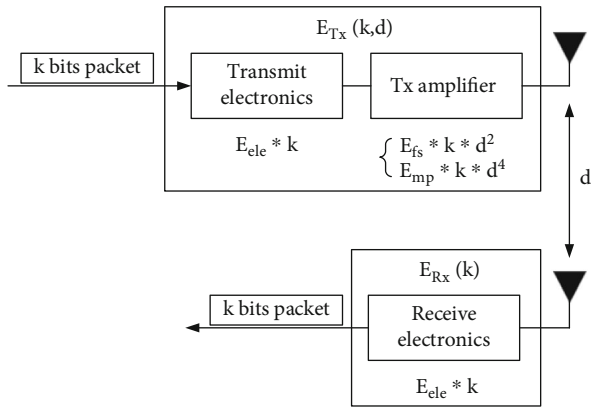


Figure 6: Wireless transmission model.

If the data transmission range of the optimal relay node still cannot transmit data to the cluster head node, continue to select the next hop relay node for relay transmission according to the above method. If within the specified number of hops, the data still cannot be transmitted to the cluster head node, the ordinary sensor node is marked as unreachable.

4.4. Energy Consumption Model of Wireless Sensor Networks. Since the energy consumption of nodes in a wireless sensor networks is mainly composed of transmission energy consumption and receiving energy consumption, in the process of data transmission and data reception of sensor nodes, for short-distance transmission, the free space model is adopted, and for long-distance transmission, the multipath fading model is adopted [20]. The wireless transmission model is shown in Figure 6.

For the symmetric propagation channel, the energy consumption when the sensor node transmits k bits data in the data packet to d meters away is shown in [21]

$$d_0 = \sqrt{\frac{E_{fs}}{E_{mp}}}, \tag{13}$$

$$E_{Tx}(k, d) = E_{Tx_{ele}}(k) + E_{Tx_{fs/mlp}}(k, d), \tag{14}$$

$$E_{Tx}(k, d) = \begin{cases} E_{ele} \times k + E_{fs} \times k \times d^2, d \le d_0, \\ E_{ele} \times k + E_{mp} \times k \times d^4, d > d_0. \end{cases} \tag{15}$$

The energy consumption of sensor nodes receiving k bits data is calculated according to

$$E_{Rx}(k) = E_{ele} \times k, \tag{16}$$

where $E_{Tx}(k, d)$ is the data sending energy consumption, $E_{Rx}(k)$ is the data reception energy consumption, E_{ele} is the energy consumption per bit of the transmitter or receiver, E_{fs} is the free space model parameter, and E_{mp} is the multipath fading model parameter.

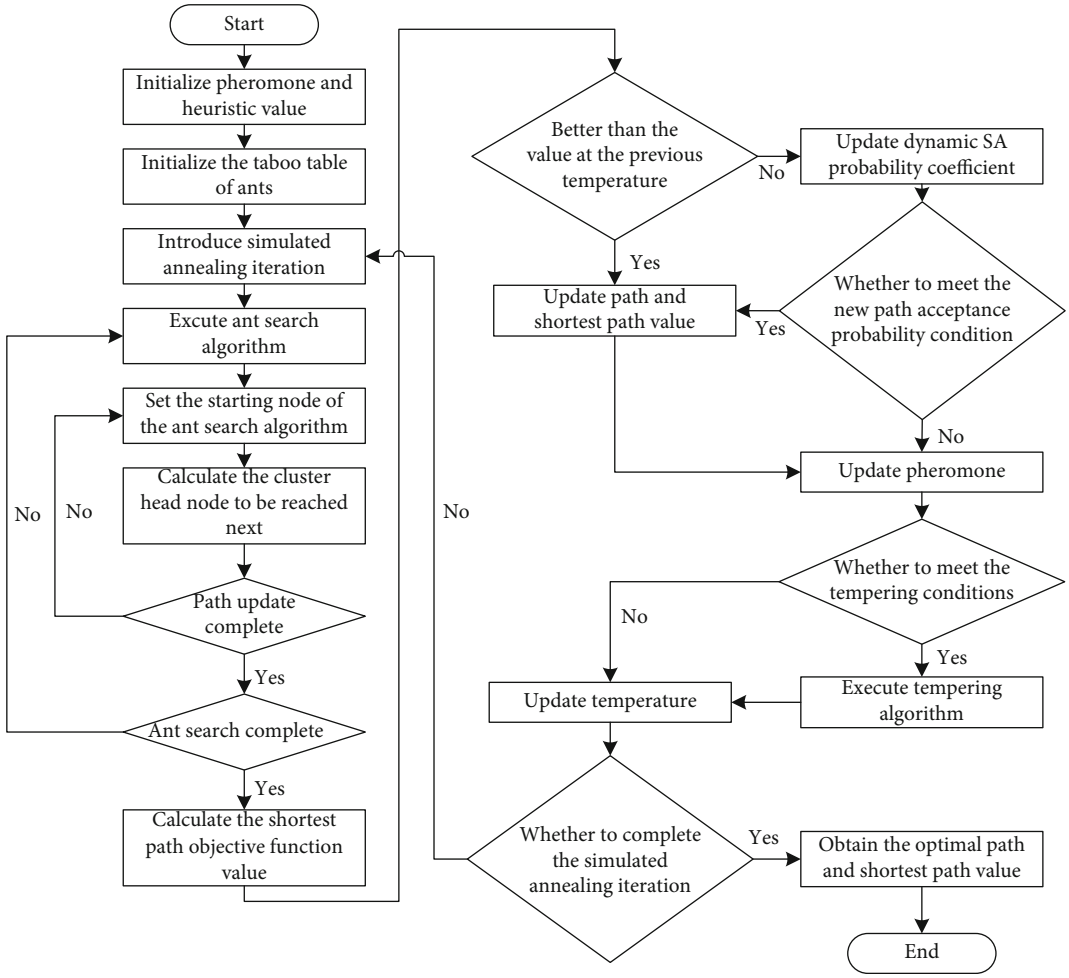


Figure 7: Execution flow chart of improved ACO.

The data fusion energy consumption of sensor nodes for k bits data is shown in

$$E_F(k) = k \times E_{da}, \tag{17}$$

where $E_F(k)$ is the energy consumption of data fusion and E_{da} is the energy consumption parameter of data fusion.

5. UAV Path Planning Based on Improved ACO

When there are multiple separated environmental monitoring areas, these different monitoring areas have different area and different number of sensor nodes, as shown in Figure 1 in Chapter 3. How to make the UAV start from the data center, use the new cluster head node information obtained in the previous round, use appropriate algorithms, follow the preplanned path to collect data from the cluster head nodes in each monitoring area, and transfer the collected environmental data to the data center is an important issue.

This problem can be simplified to a traveling salesman problem (TSP) with a fixed starting point and ending point [22]. The TSP problem is one of the most well-known problems in the field of mathematics. It can be described as follows: suppose a UAV starts from the data center and needs to fly to N sensor cluster head nodes for data collection. It must plan the flight path. The restriction is that each cluster head node can only be passed once and finally needs to return to the data center to submit sensing data. The goal of selecting the flight path is to require the resulting flight path to be the minimum of all paths. The TSP problem is an NP-hard problem with the computational complexity of NPC. Next, the research focus of this chapter will be put forward: UAV path planning method based on improved ant colony optimization.

For solving the task of UAV starting from a fixed data center, collecting sensing data from the cluster head nodes that in the separated environmental monitoring area in turn, and returning the collected data to the data center, this section uses an improved ACO to obtain a better approximate optimal solution. Introducing the simulated annealing algorithm with dynamic simulated annealing probability coefficient and tempering algorithm to further optimize the UAV's data collection path.

The algorithm process of the improved ACO is as follows.

```
1. Initialize pheromone and heuristic value;
2. Initialize the taboo table of ants;
3. Introduce Simulated Annealing iteration;
4. Execute ant search algorithm;
5. Set the starting node of the ant search algorithm;
6. Calculate the cluster head node to be reached next;
7. If Path update is not complete then
8.       Return to step 5 and continue to execute;
9. Else
10.      If Ant search is not complete then
11.            Return to step 4 and continue to execute;
12.      Else
13.            Calculate the shortest path objective function value;
14.            If Better than the value at the previous temperature then
15.                  Update path and shortest path value;
16.                  Update pheromone;
17.                  If the tempering conditions are met then
18.                        Execute tempering algorithm;
19.                        Update temperature;
20.                        If the Simulated Annealing iteration is completed then
21.                              Obtain the optimal path and shortest path value;
22.                        Else
23.                              Return to step 3 and continue to execute;
24.                        End if
25.                  Else
26.                        Go to step 19;
27.                  End if
28.            Else
29.                  Update dynamic SA probability coefficient;
30.                  If the new path acceptance probability condition is satisfied then
31.                        Go to step 15;
32.                  Else
33.                        Go to step 16;
34.                  End if
35.            End if
36. End if
```

Algorithm 1:The improved ACO.

Step 1. Initialize ant pheromone and heuristic value. Initialize the pheromone value of each edge in the UAV data collection path and the taboo table of each ant. The pheromone value on each edge is initialized to a smaller value r_0. The taboo table of each ant is used to record the sensor cluster head nodes that the ant has walked so far. Initialize the taboo table of each ant as the sensor cluster head node where the ant is currently located, and set the length of the taboo table to a. In the initial state, the pheromone value released by the ant on each edge is 0.

Step 2. Introduce the simulated annealing to iterate to construct the flight path of UAV. At the temperature T of the simulated annealing algorithm, perform an ant search. Ant k determines the sensor cluster head node to be reached in the next step according to the probability $p_{ij}^k(t)$, until it finally forms a legal UAV data collection path. Among them, the cluster head nodes that have passed are recorded in the taboo table. These nodes recorded in the taboo table cannot be included in the cluster head nodes that the ant will reach in the future. The calculation method of probability $p_{ij}^k(t)$ is shown in

$$p_{ij}^k(t) = \begin{cases} \dfrac{\left[\tau_{ij}(t)\right]^{\alpha}\left[\eta_{ij}(t)\right]^{\beta}}{\sum_{s\in a_k}\left[\tau_{is}(t)\right]^{\alpha}\left[\eta_{is}(t)\right]^{\beta}}, & j \in a_k, \\ 0, & other, \end{cases} \tag{18}$$

where $p_{ij}^k(t)$ is the selection probability of ant k moving from cluster head node i to cluster head node j in the t-th iteration and α is the pheromone factor. The larger the value of α, the greater the influence factor of the pheromone in the selection probability of the cluster head node. β is the heuristic value factor. The larger the value of β, the greater the influence factor of the heuristic value in the selection probability of the cluster head node. a_k represents the cluster head node set that is not restricted by the taboo table. $\tau_{ij}(t)$ represents the pheromone on the edge (i, j) of the ant in the t-th iteration, and $\eta_{ij}(t)$ represents the heuristic value for the ant to

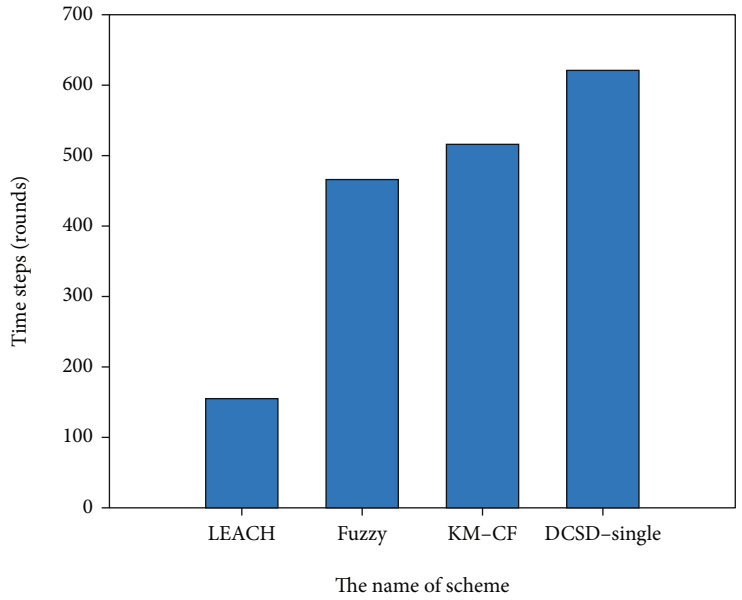


Figure 8: Number of rounds where 10% of the nodes die.

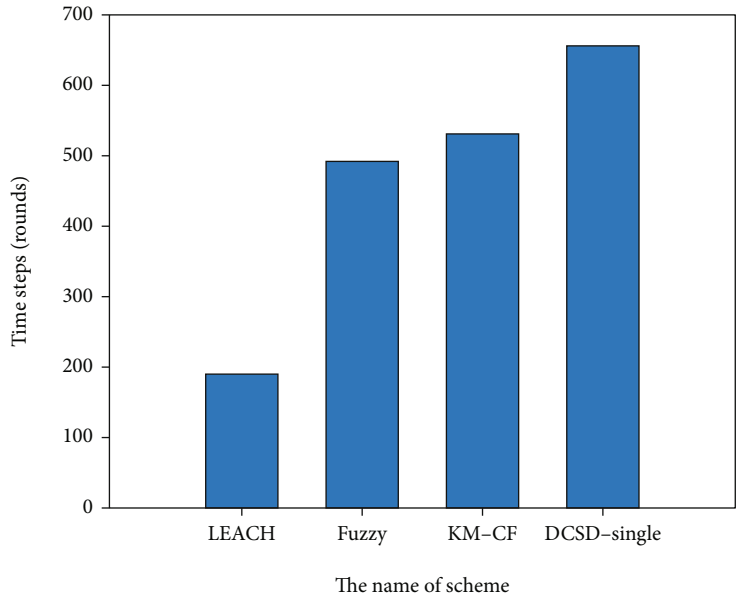


Figure 9: Number of rounds where 30% of the nodes die.

move from the cluster head node i to the cluster head node j in the t-th iteration. This value is usually the reciprocal of the distance d_{ij} between cluster head node i and cluster head node j. S are the cluster head nodes that are not restricted by the taboo table. With the constraint of Equation (18), the ant will not pass through the cluster head node again, thus ensuring that the UAV will not repeatedly pass the same cluster head node during the data collection process.

Step 3. Obtain the shortest flight path of the UAV in the t-th iteration. After the t-th iteration, each ant has completed a tour through each cluster head node. Calculate the length of the path travelled by each ant, and save the shortest path travelled, thereby obtaining the objective function value. Compare the newly obtained shortest path value with the original shortest path value (the shortest path value generated by the first iteration process is not compared, and the result of the next iteration is waited for). If the newly obtained shortest path value is better than the shortest path value at the last temperature, then directly update the UAV flight path and shortest path value. Then, follow Step 4 to update the pheromone on each edge. If the newly obtained UAV flight shortest path value is not better than the shortest

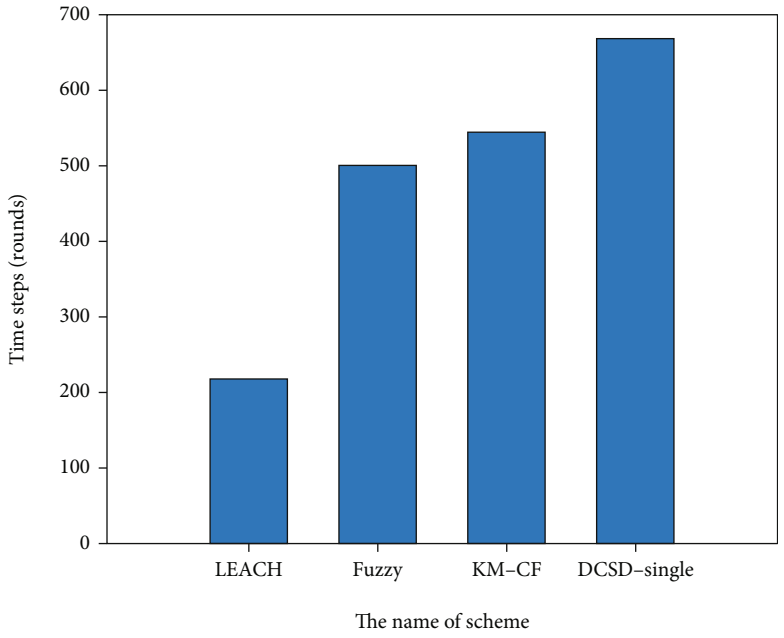


Figure 10: Number of rounds where half of the nodes die.

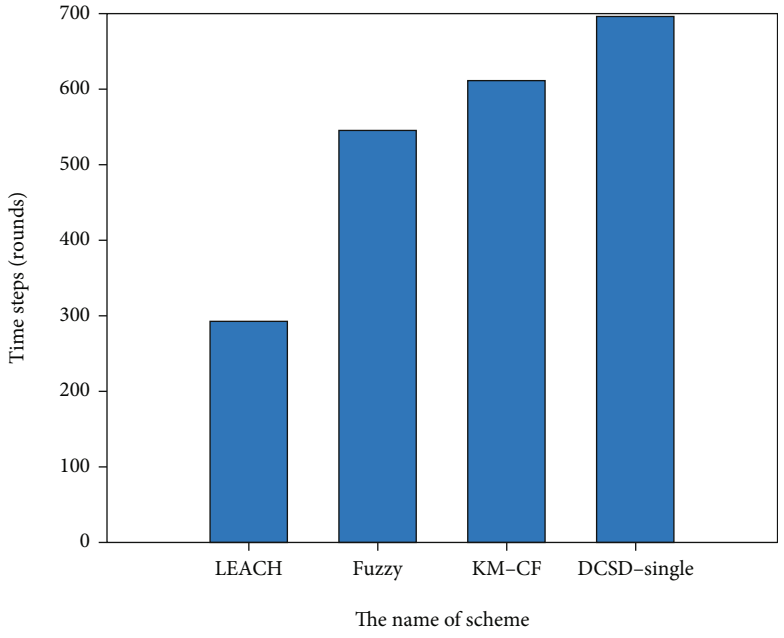


Figure 11: Number of rounds where all of the nodes die.

path value at the last temperature, update the dynamic simulated annealing probability coefficient according to

$$\omega_{current} = \omega_{max} - (\omega_{max} - \omega_{min}) \times \frac{t_{current}}{t_{max}}, \quad (19)$$

where $\omega_{current}$ is the dynamic simulated annealing probability coefficient of the current iteration number, ω_{max} is the maximum dynamic simulated annealing probability coefficient, ω_{min} is the minimum dynamic simulated annealing probability coefficient, $t_{current}$ is the current iteration number, and t_{max} is the maximum iteration number.

Then, the Metropolis acceptance criterion [23] is used to calculate the probability $P_{current}$ of introducing the dynamic simulated annealing probability coefficient at the current temperature $T_{current}$ to decide whether to accept the new path. The probability $P_{current}$ at temperature $T_{current}$ is calculated according to

$$P_{current} = \omega_{current} e^{dE/kT_{current}}, \quad (20)$$

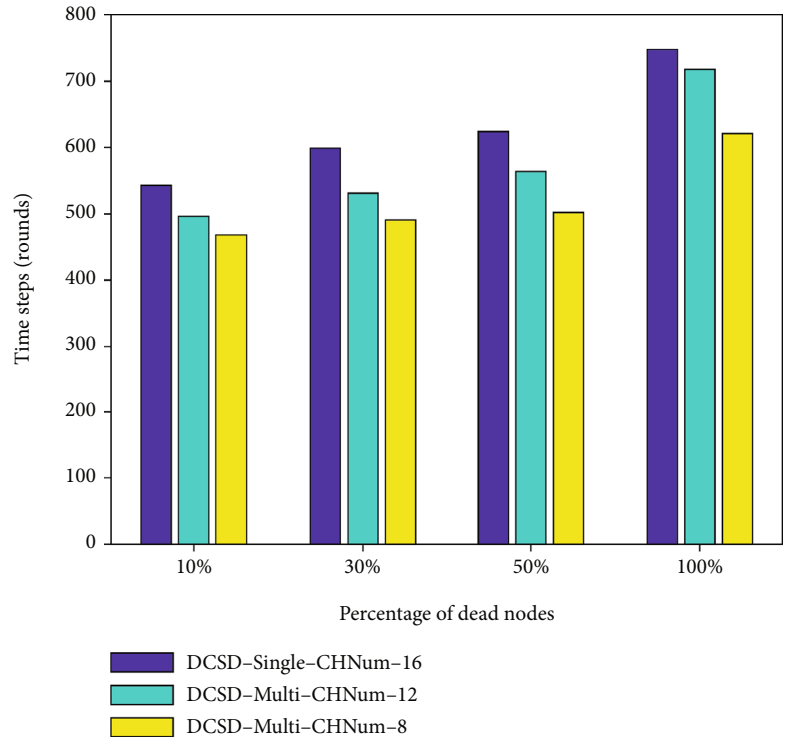


Figure 12: Comparison of the number of rounds where different number of nodes die.

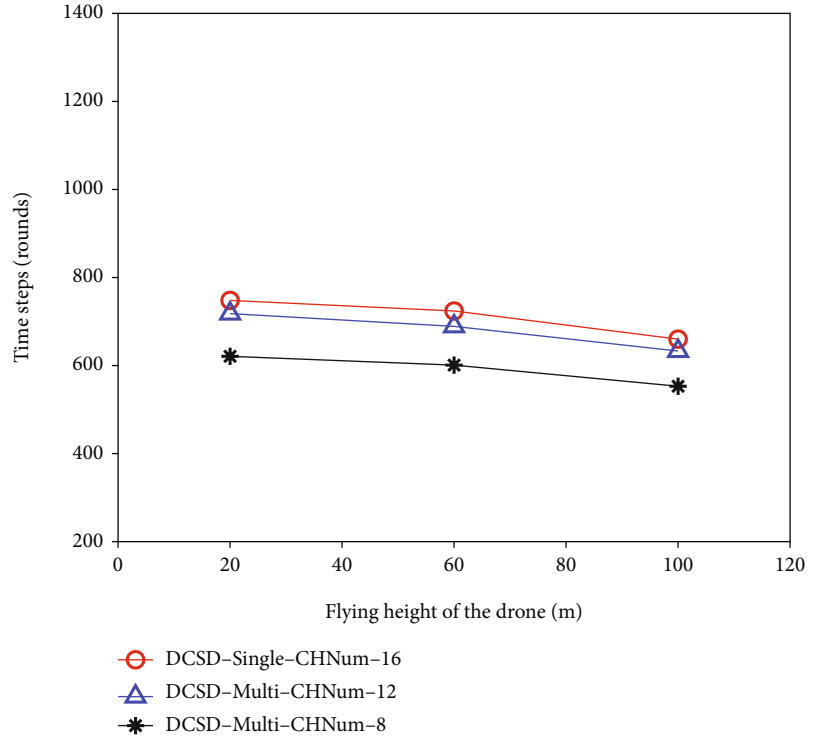


Figure 13: The influence of UAV height on the network life cycle.

where dE is the difference between the shortest path objective function value of this iteration and the shortest path objective function value of the previous iteration, k is the coefficient, and $T_{current}$ is the current temperature value.

Randomly generate a random number a_0 in the $(0, 1)$ interval. If $a_0 \leq P_{current}$, accept this solution, update the UAV flight path, and update the shortest path value. Otherwise, discard the solution generated in this iteration. From

Table 1: Simulation experiment parameter settings of improved ACO.

Parameters	Parameter choice
Number of ants	100
Pheromone factor	4
Heuristic factor	3
Pheromone volatile factor	0.1
Pheromone enhancer factor	1
Cooling factor	0.95
Initial temperature	2000 K
End temperature	0 K
Upper limit of tempering temperature	1000 K
Lower limit of tempering temperature	500 K
Maximum number of tempering	2
Number of iterations	300

the Equation (20), it can be concluded that as the iteration progresses, the temperature T is continuously reduced due to the effect of the cooling coefficient, and the value of $\omega_{current}$ continues to decrease with the increase of the number of iterations. Therefore, the value of the probability $P_{current}$ is continuously reduced. That is, the probability of accepting a poor solution is constantly decreasing, similar to the annealing crystallization process of crystals in nature.

Step 4. Update the pheromone value. The update of the pheromone on each edge of the path includes the volatilization of the pheromone due to the passage of time and the newly produced pheromone released by the ants when they passed by. Pheromone is updated according to the rules shown in

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \sum_{k=1}^{m} \Delta\tau_{ij}^{k}(t, t+1), \tag{21}$$

where $\tau_{ij}(t+1)$ is the pheromone on the edge (i, j) at the $(t+1)$-th iteration and $\tau_{ij}(t)$ is the pheromone on the edge (i, j) at the t-th iteration. ρ is the pheromone volatilization factor, and $\rho \in (0, 1]$. $1\text{-}\rho$ is the pheromone maintenance factor. k is the k-th ant. m is the total number of ants. $\Delta\tau_{ij}^{k}(t, t+1)$ is the pheromone released by the k-th ant on the edge (i, j) when it passes during the t-th iteration. $\Delta\tau_{ij}^{k}(t, t+1)$ is calculated according to the method shown in

$$\Delta\tau_{ij}^{k}(t, t+1) = \frac{Q}{l_k^{\mu}}, \tag{22}$$

where Q is the pheromone enhancement factor, l_k is the length of the path constructed by the k-th ant, and μ is the coefficient. The smaller the path length l_k, the Equation (22) shows that the higher the pheromone content on each edge of the path, and the greater the probability of being selected by other ants.

After all ants complete the pheromone update operation, save and record the shortest path of current UAV data collection. Then, initialize the taboo table and pheromone increment.

Step 5. Judgement of tempering conditions. After the pheromone released by the ants on each edge is updated, it is checked whether the tempering conditions are met, so as to decide whether to perform the tempering operation. Whenever the temperature T during the simulated annealing operation is less than or equal to the minimum value of the tempering temperature $T_{min\_tempering}$, and the current tempering times $H_{current}$ has not reached the set maximum upper limit of the tempering times $H_{\max}$, use Equation (23) for tempering operation.

$$T_{current} = T_{max\_tempering}, \tag{23}$$

where $T_{current}$ is the current temperature value and $T_{max\_tempering}$ is the highest temperature value of the tempering temperature. According to the above operation, at most H_{max} tempering operations are performed, the objective function value is more optimized.

Finally, use the cooling coefficient to update the temperature value, and the update method is shown in

$$T(t+1) = T(t) \times q_{cooling}, \tag{24}$$

where $T(t+1)$ is the temperature after the update, $T(t)$ is the temperature before the update, and $q_{cooling}$ is the cooling coefficient.

Repeat the execution from Step 2 and loop in turn until the termination condition of the improved ACO is met. The final optimal path and shortest path value are output. The execution process of the improved ACO is shown in Figure 7 and Algorithm 1.

The improved ACO is implemented in the UAV node to realize edge computing. Before the next round of data collection tasks arrive, the UAV plans the data collection path through the new cluster head node ID and coordinate value calculated by itself in the previous round. The UAV collects sensor network data according to the data collection path planned in advance. Among them, when the UAV performs the first data collection task, it uses the cluster head node ID and coordinate values obtained by the initial clustering to plan the path, and can ignore the influence of other factors.

6. Simulation Experiment and Result Analysis

This chapter uses MATLAB R2019a to perform simulation experiments to analyse the performance of the solutions proposed and discussed above.

Dead nodes refer to the nodes whose remaining energy is not enough to continue to complete the data collection task of the sensor network. Figures 8–11 show the first rounds of 10% of the nodes die, 30% of the nodes die, half of the nodes die, and all of the nodes die in the sensor network under LEACH [24], Fuzzy [25], KM-CF [26], and DCSD-single scheme proposed in this paper which is suitable for single-hop scenarios. In a sensor network, the number of rounds

Table 2: Coordinate information of data center and cluster heads.

The category of the node	Node number	Node abscissa	Node ordinate	Node number	Node abscissa	Node ordinate
Data center	1	500.0000	1000.0000			
Sensor network 1	2	221.5789	555.4625	3	42.1414	728.7543
	4	41.1697	638.8750	5	266.0404	768.2090
	6	108.4884	748.4868	7	35.6075	565.2155
	8	135.1900	646.9238	9	249.1216	642.9315
	10	119.5054	552.3907	11	199.0793	738.1214
Sensor network 2	12	1011.9638	770.9435	13	818.4280	657.0365
	14	842.5954	778.2446	15	894.6581	608.0746
	16	901.0757	706.7549	17	981.4248	656.3408
Sensor network 3	18	912.5777	831.7891			
	19	611.1036	430.7296	20	436.8288	384.1664
	21	504.2333	379.2322	22	557.5915	354.8317
	23	472.0340	473.5037	24	454.2806	296.9911
Sensor network 4	25	576.2068	289.9250			
	26	101.4379	27.3214	27	35.8978	71.9208
Sensor network 5	28	113.7558	118.9788			
	29	1242.7748	151.2065	30	1327.0224	163.8992
	31	1271.5899	74.8152			

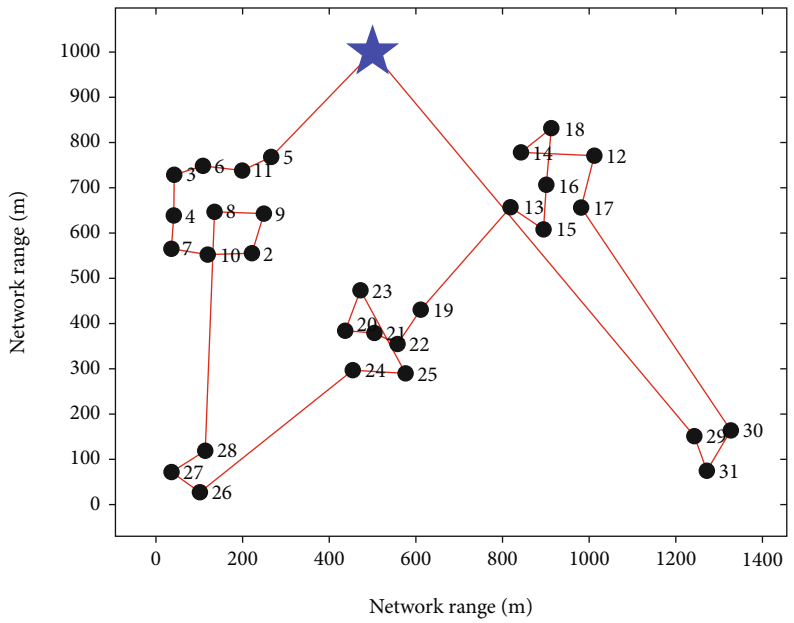


Figure 14: Optimal flight path of ACO.

in which 10% of the nodes die, 30% of the nodes die, half of the nodes die, and all of the nodes die for the first time is an important performance evaluation index for the life cycle of the sensor network. As can be seen from the figure, compared with the other three schemes, the scheme proposed in this paper delays the first occurrence of the death of 10% nodes, the death of 30% nodes, the death of half nodes, and the death of all nodes and has better performance.

As mentioned earlier, for scenarios with a large network range, the UAV-assisted sensor data collection method proposed in this paper for single-hop scenarios can be used to divide an appropriate number of clusters. This method makes the distance between the sensor node and the cluster head node less than or equal to the communication distance of the node to complete the single-hop transmission between the sensor node and the cluster head node, which is the DCSD-single scheme. Or use the UAV-assisted sensor network data collection method proposed in this paper for multihop scenarios. Select a suitable relay node as a transmission relay, and solve the unreachable problem caused by the

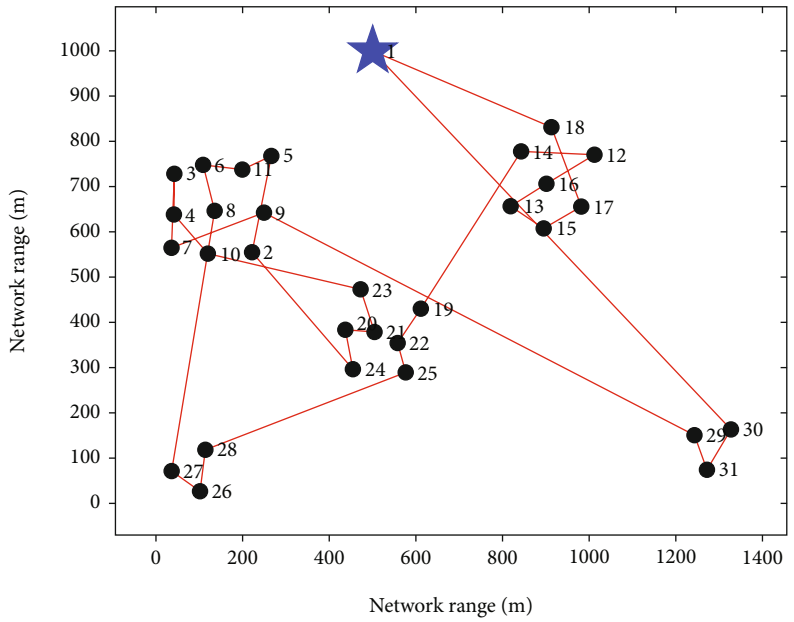


Figure 15: Optimal flight path of SA.

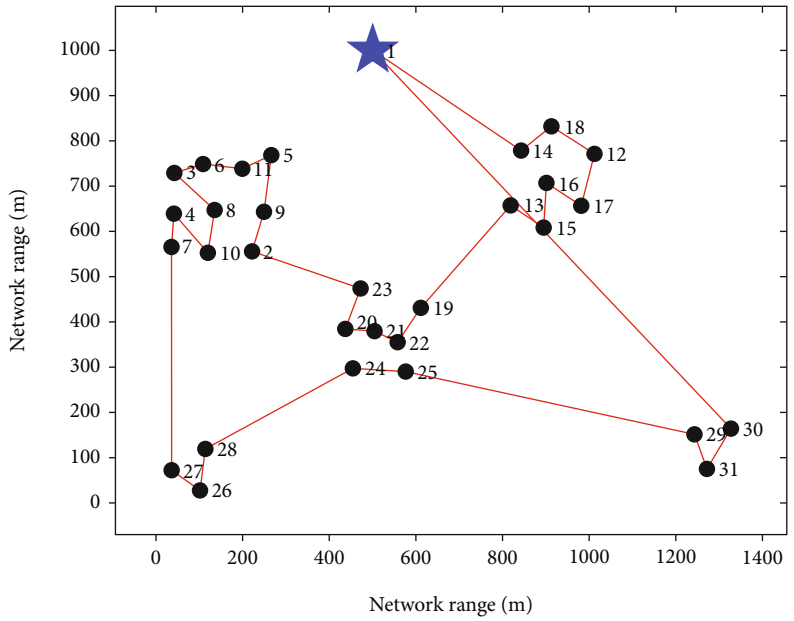


Figure 16: Optimal flight path of GA.

distance between the sensor node and the cluster head node being larger than the node communication distance. It is the DCSD-multi scheme.

The experimental scene is expanded. The DCSD-single scheme in the single-hop scenario and the DCSD-multi scheme in the multihop scenario are compared and tested in a 400 m ∗ 400 m monitoring area with 200 sensor nodes randomly distributed. Among them, in order to make multihop transmission occur, appropriately expand the range of each cluster in the DCSD-multi scheme in the multihop transmission scenario, that is, reduce the number of cluster head nodes in the network.

Because the flying height of the UAV will affect the distance between nodes, it will affect the energy consumption of information transmission. Therefore, the following comparative experiment considered the influence of the UAV's flight height on the experimental results.

Figures 12 and 13 show that when the number of cluster head nodes in the DCSD-single scheme is 16 and the number of cluster head nodes in the DCSD-multi scheme are 12 and 8, the comparison graph of 10% of the nodes die, 30% of the nodes die, half of the nodes die, and all of the nodes die for the first time and the comparison graph of the network life cycle with the UAV flight altitude change.

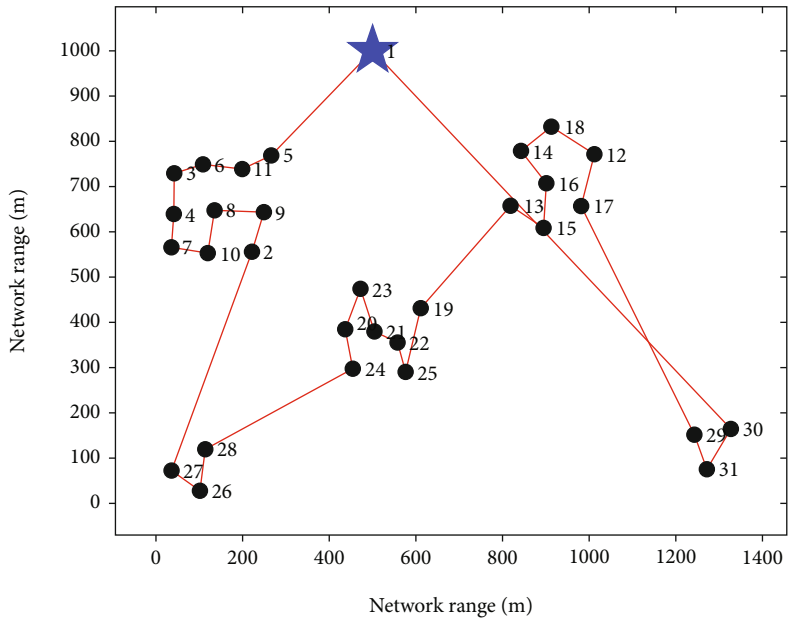


Figure 17: Optimal flight path of improved ACO.

Table 3: Performance comparison of four algorithms.

Algorithm name	Number of iterations to convergence	Shortest path length (m)
ACO	About 180	4697.07
SA	About 290	6389.05
GA	About 270	4802.35
Improved ACO	About 170	4410.41

Figure 13 shows the impact of UAV flight height on the performance of various schemes. As the flying height of the UAV increases, the cluster head node needs more transmission power to transmit data to the UAV node, which leads to an increase in the energy consumption of the cluster head node and reduces the life cycle of the network. Therefore, when the UAV is actually used to assist the sensor network for data collection, under the premise of ensuring safety, reducing the flying height of the UAV to a certain extent can delay the death time of the sensor nodes in the environmental monitoring area.

This chapter uses MATLAB R2019a to plan the UAV data collection path. Assume that there are five environmental monitoring areas separated from each other due to geographical environmental factors, which form a distributed wireless sensor networks through the flight transmission of UAV. According to the method of "store-carry-forward," starting from the data center, the UAV collects data from the cluster head nodes in each separated environmental monitoring area through the designated algorithm, and then according to the planned optimal path, the UAV finally transmits the data to the data center. In most ecological environmental monitoring scenarios, sensor nodes are deployed on the ground, so the height of each sensor node is ignored, and the network scenario is simplified to a two-dimensional coordinate plane. This scene has a data center and five separated sensor network environmental monitoring areas.

The algorithm evaluation indicators in this chapter mainly include the total path length of UAV data collection, the minimum number of iterations required for each algorithm to reach the shortest path, and the convergence speed of each algorithm. Suppose the data center coordinates are (500, 1000), the range of sensor network 1 is 300 m ∗ 300 m, and the number of nodes is 100. The range of sensor network 2 and sensor network 3 is 250 m ∗ 250 m, respectively, and the number of nodes is 70, respectively. The range of sensor network 4 and sensor network 5 is 150 m ∗ 150 m, respectively, and the number of nodes is 50, respectively. According to the cluster head election method described in Chapter 4, the coordinates of the sensor cluster head nodes in each monitoring area can be obtained during a certain round of data collection. The simulation experiment parameter settings of the improved ACO are shown in Table 1.

This chapter conducts simulation experiments on four intelligent algorithms. The node numbered 1 shown by the five-pointed star logo represents the data center. The cluster head nodes elected in each partitioned environmental monitoring area are represented by number 2-31, respectively. The coordinate of the data center and the coordinate information of the cluster head nodes elected in each partitioned sensor network are shown in Table 2.

Figures 14–17 show the results of using four algorithms for UAV data collection path planning, respectively. Figures 14–17 show the optimal flight path for UAV path planning using four algorithms, respectively. Among them, 1 is the data center, which is the starting point and end point of the UAV data collection process. Nodes 2-31 are cluster head nodes distributed in each partition monitoring area.

The comparison of the number of iterations and the shortest path length of the ant colony optimization,

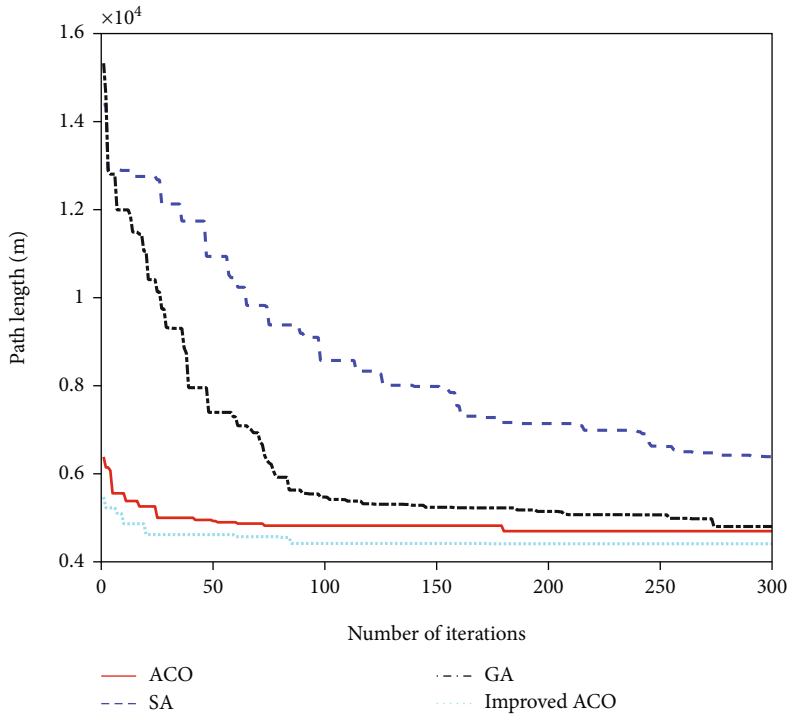


Figure 18: Comparison of iteration curves of four algorithms.

simulated annealing, genetic algorithm, and improved ACO to achieve convergence is shown in Table 3.

The comparison of the iterative curves of the four algorithms to reach the shortest path of UAV data collection is shown in Figure 18. It can be seen from the figure that in the experimental scenario proposed in this chapter, the quality of the solution is poor in the initial search execution stage of the simulated annealing and the convergence speed of the algorithm is slow when it reaches the shortest path. The genetic algorithm also has the problem of low solution quality in the initial search execution stage, and the convergence speed is slow when it reaches the shortest path. However, compared with simulated annealing, genetic algorithm greatly reduces the shortest path length. In comparison, the ant colony optimization and the improved ACO can obtain higher-quality solutions in a limited number of iterations; that is, they can converge to the shortest flight path of the UAV faster. Using the improved ACO to plan the UAV data collection path can get a better shortest path length and can get the convergent shortest path result at the fastest speed.

7. Conclusion and Outlook

This paper studies the data collection method of sensor network based on UAV and introduces the idea of edge computing into it. Distributed wireless sensor networks separated from each other are assisted by UAV technology for clustering and cluster head elections, which reduces the overall energy consumption of the distributed sensor network. Then, the improved ant colony optimization is used to plan the UAV data collection path. Through the "store-carry-forward" method, the UAV takes the shortest path to carry the sensed data and transmits it to the data center.

In future work, we will consider the use of UAVs to charge and collect data in the sensor network, consider the impact of different sensor node distribution patterns on the performance of the sensor network, and consider multi-UAVs to coordinate the data collection of the sensor network to further improve the efficiency of data collection to serve a wider monitoring area and complete the corresponding environmental monitoring tasks.

In addition, it is considered that the intelligent algorithm (improved ACO) can be changed into a reinforcement learning algorithm, and a reinforcement learning algorithm that conforms to its own scene can be set, so that the UAV can plan the optimal path by itself, thus improving the data collection efficiency of the sensor network.

Data Availability

The experimental data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (62077032), the Inner Mongolia Natural

Science Foundation (2020MS06023), and the Inner Mongolia Science and Technology Plan Project (2021GG0159).

References

[1] B. Yang and X. Bai, "Data collection strategy based on drone technology in wireless sensor networks," in *2020 16th International Conference on Mobility, Sensing and Networking (MSN)*, pp. 129–136, Tokyo, Japan, 2020.

[2] S. Huang and H. Chang, "A farmland multimedia data collection method using mobile sink for wireless sensor networks," *Multimedia Tools and Applications*, vol. 76, no. 19, pp. 19463–19478, 2017.

[3] K. G. Ngandu, K. Ouahada, and S. Rimer, "Smart meter data collection using public taxis," *Sensors*, vol. 18, no. 7, p. 2304, 2018.

[4] C. Li, *Research on Energy-Saving Strategy of Wireless Sensor Networks with a Mobile Agent Node*, Chongqing University, 2018.

[5] J. Chen, F. Yan, S. Mao et al., "Efficient data collection in large-scale UAV-aided wireless sensor networks," in *2019 11th International Conference on Wireless Communications and Signal Processing (WCSP)*, pp. 1–5, Xi'an, China, 2019.

[6] Y. Wang, Z. Hu, X. Wen, Z. Lu, and J. Miao, "Minimizing data collection time with collaborative UAVs in wireless sensor networks," *IEEE Access*, vol. 8, pp. 98659–98669, 2020.

[7] M. O. U. Zhiyu, Y. Zhang, F. A. N. Dian, L. I. U. Jun, and G. A. O. Feifei, "Research on the UAV-aided data collection and trajectory design based on the deep reinforcement learning," *Chinese Journal on Internet of Things*, vol. 4, no. 3, pp. 42–51, 2020.

[8] J. Zhang, "Research on multi-UAV scheduling algorithms for wireless sensor network data collection," *Modern Computer*, vol. 4, pp. 33–37, 2021.

[9] W. Chen, S. Zhao, R. Zhang, Y. Chen, and L. Yang, "UAV-assisted data collection with nonorthogonal multiple access," *IEEE Internet of Things Journal*, vol. 8, no. 1, pp. 501–511, 2021.

[10] X. Xu, H. Zhao, H. Yao, and S. Wang, "A Blockchain-enabled energy-efficient data collection system for UAV-assisted IoT," *IEEE Internet of Things Journal*, vol. 8, no. 4, pp. 2431–2443, 2021.

[11] G. Zhu, L. Guo, C. Dong, and X. Mu, "Mission time minimization for multi-UAV-enabled data collection with interference," in *2021 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1–6, Nanjing, China, 2021.

[12] R. Ma, R. Wang, G. Liu, H. H. Chen, and Z. Qin, "UAV-assisted data collection for ocean monitoring networks," *IEEE Network*, vol. 34, no. 6, pp. 250–258, 2020.

[13] D. Ebrahimi, S. Sharafeddine, P. Ho, and C. Assi, "UAV-aided projection-based compressive data gathering in wireless sensor networks," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1893–1905, 2019.

[14] Z. Du, C. Wu, T. Yoshinaga et al., "A routing protocol for UAV-assisted vehicular delay tolerant networks," *IEEE Open Journal of the Computer Society*, vol. 2, pp. 85–98, 2021.

[15] Z. Qi, Z. Shao, Y. S. Ping, L. M. Hiot, and Y. K. Leong, "An improved heuristic algorithm for UAV path planning in 3D environment," *2010 Second International Conference on Intelligent Human-Machine Systems and Cybernetics*, vol. 2, pp. 258–261, 2010.

[16] X. Chen, C. Wu, T. Chen et al., "Information freshness-aware task offloading in air-ground integrated edge computing systems," *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 1, pp. 243–258, 2022.

[17] C. Wu, Z. Liu, F. Liu, T. Yoshinaga, Y. Ji, and J. Li, "Collaborative learning of communication routes in edge-enabled multi-access vehicular environment," *IEEE Transactions on Cognitive Communications and Networking*, vol. 6, no. 4, pp. 1155–1165, 2020.

[18] P. Tong, J. Liu, X. Wang, B. Bai, and H. Dai, "Deep reinforcement learning for efficient data collection in UAV-aided Internet of Things," in *2020 IEEE International Conference on Communications Workshops (ICC Workshops)*, pp. 1–6, Dublin, Ireland, 2020.

[19] J. Zhang and Y. Sui, "Key technology of TDMA—the technology of time slot," *Applied Science and Technology*, vol. 28, no. 6, pp. 15–17, 2001.

[20] P. Nayak and A. Devulapalli, "A fuzzy logic-based clustering algorithm for WSN to extend the network lifetime," *IEEE Sensors Journal*, vol. 16, no. 1, pp. 137–144, 2016.

[21] Z. Li, Y. Tao, Y. Zhou, and L. Yang, "Energy-balanced multi-hop cluster routing protocol based on energy harvesting," *Computer Science*, vol. 47, no. S2, pp. 296–302, 2020.

[22] C. Gao, B. Feng, and L. Zhu, "Reviews of the meta-heuristic algorithms for TSP," *Control and Decision*, vol. 21, no. 3, pp. 241–247, 2006.

[23] H. Chen, J. Wu, J. Wang, and B. Chen, "Mechanism study of simulated annealing algorithm," *Journal of Tongji University (Natural Science)*, vol. 32, no. 6, pp. 802–805, 2004.

[24] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences*, vol. 2, p. 10, 2000.

[25] J. Mi, X. Wen, C. Sun, Z. Lu, and W. Jing, "Energy-efficient and low package loss clustering in UAV-assisted WSN using Kmeans++ and fuzzy logic," in *2019 IEEE/CIC International Conference on Communications Workshops in China (ICCC Workshops)*, pp. 210–215, Changchun, China, 2019.

[26] M. Shen, *Research on Wireless Sensor Network Communication Based on Unmanned Aerial Vehicle*, Hangzhou Dianzi University, 2019.

Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 1377225, 8 pages
https://doi.org/10.1155/2022/1377225

Research Article

Resource Optimization in MEC-Assisted Multirobot Cooperation Systems

Lanxin Qiu,[1,2] **Yi Zhang**,[1] **Yanbo Wang**,[1] **Yineng Shen**,[2] **Jiantao Yuan**,[3] **and Rui Yin**[3]

[1]*State Grid Zhejiang Electric Power Company Information & Telecommunication Branch, Hangzhou 310000, China*
[2]*Dept. of Information and Electrical Engineering, Zhejiang University, Hangzhou 310000, China*
[3]*Dept. of Information and Electrical Engineering, Zhejiang University City College, Hangzhou 310000, China*

Correspondence should be addressed to Yineng Shen; shenyineng@zju.edu.cn

Received 10 January 2022; Accepted 11 April 2022; Published 9 May 2022

Academic Editor: Carles Gomez

With prevalent utilization of *multirobot cooperation* (MRC) systems, people pay more attention to improve the system performance. Among them, the energy consumption and implementation latency of MRC systems are major concerns, and *mobile edge computing* (MEC) provides a potential way to solve these problems. Therefore, how to leverage MEC to get the balance between computing and communication consumption in MRC systems needs to be investigated urgently. In this paper, a MRC system deployed to accomplish multiple time-critical tasks by MEC technology is studied. The proposed MRC system includes a powerful *master robot* (MR) and several *slave robots* (SRs). As a scheduler, MR is responsible for allocating tasks to SRs and has more computing power. SRs are robots with sensors that interact with the environment. In this paper, we propose a strategy for task allocation and resource management in MRC systems. The results show that the proposed scheme can effectively reduce the total energy consumption in SRs.

1. Introduction

At present, there are many practical applications of multirobot systems, such as floor mopping robots, rescue robots after nature disasters, surgery by surgical robots, and manufacture by mechanical arms [1]. Obviously, multirobot collaboration can greatly improve efficiency compared to a single robot. Therefore, a very hot research direction is the MRC system [2]. However, in a MRC system, the difficulty in accomplishing the lantency-sensitive and computation-intensive tasks comes from diverse computation, communication and sensing capacities of the individual robots, and the limited battery budget [3]. On the one hand, multirobots need cooperate to accomplish the time-critical jobs. On the other hand, efficient cooperation between multirobot is definitely inseparable from efficient communication. Fortunately, MEC technology has emerged as a promising solution to enable the task offloading via wireless communication, which perfectly fits the MRC system structure.

As driven by the explosive growth on data traffic, *artificial intelligence* (AI) has revolutionized science and social life. However, AI also brought tremendous computation workloads [4]. Inevitably, robots also undertake more and more AI tasks in various applications, which imposes a certain computational burden on robot with limited size. To ease these burdens, many research have been done to accelerate computing efforts in the following three areas [2]: different types of robots, control architectures, and communication technologies.

In order to solve the above problems, both the field of robotics and the field of communication are making contributions. In robotics, the solution is to collaborate by adding more robots to overcome the resource constraints of a single robot. Firstly, in [5], the author introduces the controlled mobility to enable the sparse sensing, which can be exploited for energy-efficient and nonredundant sensing in MRC networks. Secondly, through efficient wireless communication technology, robots can share data and working status. In

[6], the authors come up an ad hoc based MRC system in which p-persistent real-time ALOHA is used as the channel competation strategy. By doin this, they achieve ultrareliability and ultralow latency communication. Furthermore, some researchers have noticed that with the development of wireless communication technology, it is a feasible way to deploy the computing tasks of robots to the cloud [7]. For instance, in [8], the authors utilize a distributed framework to build a visual SLAM robotic system, where cloud servers are responsible for map optimization and storing information to reduce computational workload. Furthermore, UAV is a segment of robotics research that has received great attention, and in [9], a joint offloading and trajectory design scheme that minimizes the sum of maximum delay is proposed in a MEC-based UAV system.

As an emerging computation offloading and wireless communication combination technology, MEC deploys cloud-like functions closer to the edge of networks to reduce latency. MEC trades off computational offloading and wireless communication, which plays an important role in reducing energy consumption and computing latency [10]. MEC has also been extensively used in many areas including *vehicle to vehicle* (V2V) [11], *unmanned aerial vehicle* (UAV) [12], and augmented reality [13]. In [14], an air-ground integrated multiaccess edge computing system has been investigated. The interaction among mobile users has been modelled as a stochastic game, which is transformed into a single-agent Markov decision process at each user. Then, an online deep reinforement learning scheme has been proposed to approximate the Q-factor and postdecision Q-factor, respectively, which is used to find the optimal solution at each user. In [15], computation offloading in beyond the fifth generation networks has been studied, where the combination of the wireless communications and multiaccess edge computing is considered. Multiagent Markov decision process has been applied to model the computation offloading problem, where a distributed learning framework is developed. Authors in [16] have extended the MEC technology to the unlicensed bands, where a context-aware communication approach is to efficiently integrate licensed and unlicensed spectrums by MEC technologies.

Based on above investigation, we can observe that MEC is suitable for computation task offloading in MRC systems. In [17], the authors propose an energy-efficient task offloading scheme for multiple devices for TDMA and OFDMA systems. Moreover, in [18], an algorithm based on *Alternating Direction Method of Multipliers* (ADMM) is proposed to maximize the revenue of *MEC system operator* (MSO), which is comprehensive considering offloading, resource allocation, and content caching strategies. To overcome network congestion and long latency in cloud computing, a multilevel resource management algorithm is designed in [19], in which cloud and edge servers cooperate to complete tasks. In the scenario of multiserver serving multiuser, [20, 21] have proposed centralized and distributed methods to derive the optimal task offloading strategies, respectively. In [22], we have propose a MEC-based MRC system and an algorithm to offload task and allocate the resource which focus on fairness and robustness. In [23], we have developed the optimal resource scheduling scheme to ensure that the task can be accomplished in time. Therefore, in our previous work [22, 23], there have two different models, where the optimal model guarantees the performance, and the fairness model guarantees that the SR is always "online." However, how to achieve the optimality and fairness is still an urgent problem to be solved.

This paper is devoted to proposing a resource allocation scheme applied to an MRC system, which minimizes the total energy consumption of SRs under a task implementation latency constraint. First, we consider a complicated realistic scenario where the SRs are in charge of the sensing and data collection and a MR is responsible for the task offloading and wireless communication resource management. Accordingly, a task implementation strategy in MEC-based MRC system is proposed, in which we divide each task into three parts. Then, an optimal and fair resource scheduling scheme is proposed to minimize the energy consumption of SR. The results show that the proposed scheme can effectively reduce the total energy consumption in SR.

2. System Model

In the paper, a time-critical task implementation process is proposed for MRC systems, and an MEC-based resource allocation scheme is studied. As shown in Figure 1, a MR M cooperates with K SRs, which denoted by a set of $\mathcal{S} = \{s_1, s_2, \cdots, s_K\}$. In addition, a powerful base station will handle some computing tasks for MR. The MR has more powerful computing ability than SRs, but SR has the ability to interact with the environment. Then, we let the MR lead multiple SRs on latency-sensitive computationally intensive tasks, such as AI applications and path planning.

As shown in Figure 2, the proposed task implementation process is divided to three stages. The first stage is called the sensing stage where each SR collects data required by the task under a duration of $T^{(ss)}$. The second is named as SR offloading stage. The SRs will process a certain amount of data locally and the rest of data will be offloaded to the MR with a time limitation $T_s^{(co)}$. In the MR offloading stage, which is the third stage, the MR also preform computation and offloading trade-offs to meet the latency $T_M^{(co)}$.

On the one hand, the MR needs to decide the amount of data each SR needs to collect, the transmit power of each SR, and the amount of data offloaded from each SR. The MR makes decisions based on the channel status of the SR and the remaining battery power, so the SR will feed back these information to the MR through the control channel at the beginning to ensure reliable communication. The MR then feeds back the decision to each SR via the control channel. Similarly, at the beginning of the third stage, the MR first measures and estimates the channel state information between the MR and the BS.

On the other hand, general working robots cannot be very large, so their battery and computing device capabilities are also limited. So we assume that SR has much less battery capacity and computing power than MR, and most of the energy of SR will be spent on sensing and collecting data

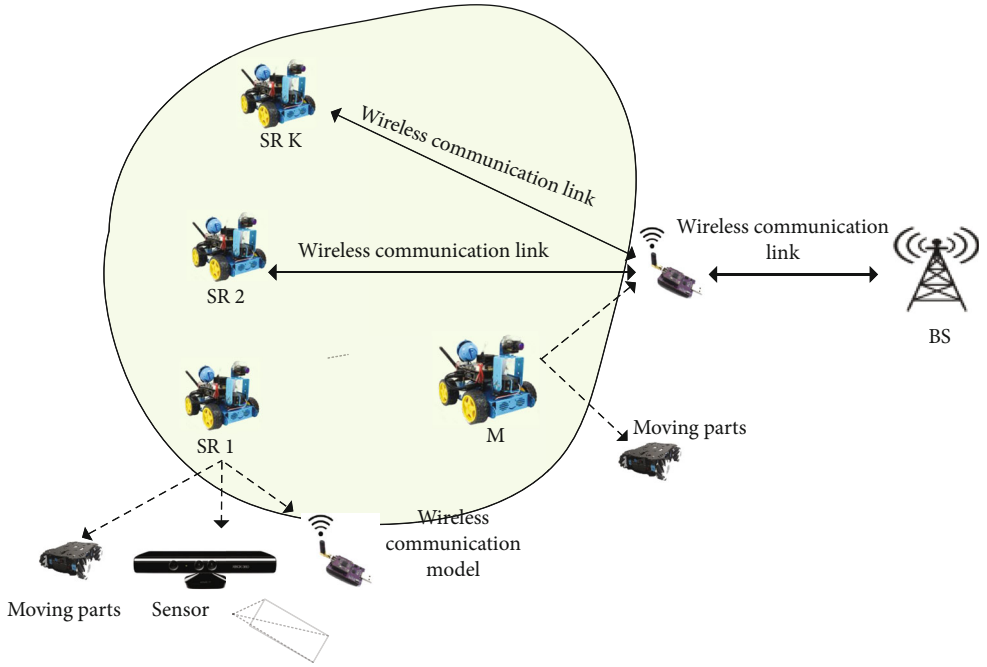


Figure 1: Multirobot cooperation system model.

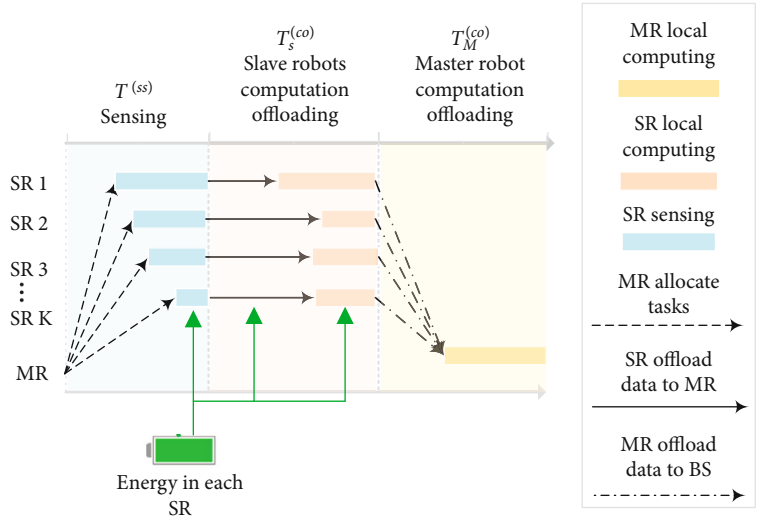


Figure 2: The time division structure and energy flow for system model.

in the first stage. In order to maintain the functionality of the MRC system for as long as possible, the power consumption of the SR should be minimized and balanced. To achieve this goal, the mathematical model of the system will be described in detail in the following subsections.

2.1. Data Collection Model. Firstly, MR will allocate different sensing time $t_k^{(ss)}$ to each SR based on the information provided by the SR, which is a variable. The amount of collected data bits at the k-th SR is denoted as

$$d_k = \frac{p_k^{(ss)} t_k^{(ss)}}{H_k}, \tag{1}$$

where H_k represents the energy consumption used to sense one bit [24], and $p_k^{(ss)}$ is a constant which denote as the sensing power consumption at the kth SR.

2.2. Computation-Offloading Model. Figure 2 also shows the direction of data flow and energy consumption in the MRC system. Let $d_k^{(ol)}$ denotes the offloaded data bits from the kth SR to the MR, and $d_M = \sum_{k \in \mathcal{S}} d_k^{(ol)}$ as the total received data from SRs.

Then, $(d_k - d_k^{(ol)})$ bits are the data that need to be processed locally. Moreover, let W_k represents the computation capability of the kth SR, which is the number of *central processing unit* (CPU) cycles executed per second, and W_M as

the computation capacity of the MR. Moreever, C_k as the number of CPU cycles is required at the kth SR to calculate 1-bit data, and C_M as the number of CPU cycles is required for computing 1-bit of data at the MR. To guarantee the latency constraint, the time spent on local computing should be less than a threshold $T_s^{(co)}$.

2.3. Communication Model. Consider the TDMA system. According to Shannon formula, the transmission data rate from the k-th SR to the MR is given by

$$r_k = B_k \log_2\left(1 + \frac{p_k^{(tr)} h_k}{N_k}\right), \tag{2}$$

where B_k is the channel bandwidth allocated to the kth SR, $p_k^{(tr)}$ is the transmission power consumed at SR k, h_k denotes the mean channel gain on the wireless channel between SR k and the MR, and N_k is the Gaussian channel noise, which is fixed. It is noteworthy that the channels between the SRs and MR experience both large fading due to the path loss and shadowing and fast fading due to the reflection and diffraction. Based on Equation (2), the time spent on offloading data from the kth SR to the MR can be written as

$$t_k^{(ol)} = \frac{d_k^{(ol)}}{r_k}. \tag{3}$$

Similarly, when the MR offloads the data to the BS, the transmission rate is given by

$$r_M = B_M \log_2\left(1 + \frac{p_M^{(tr)} h_M}{N_M}\right), \tag{4}$$

where B_M is the bandwidth allocated to the MR, $p_M^{(tr)}$ is transmission power allocation at the MR, h_M is the mean channel gain between the MR and the BS, and N_M is the noise power, which is fixed. According to Equation (4), the time spent on the transmission at the MR is given by $t_M^{(ol)} = D_M^{(ol)}/r_M$.

2.4. Energy Consumption Model. As shown in Figure 2, energy needs to be consumed in the three stages of the system. It is worth noting that we assume that the BS has sufficient computing and communication capabilities and energy. In this paper, we focus on the energy consumption of SRs, hence the computation time spent on the feedback from the BS and the energy cost at the BS is neglected.

First, According to Section 2.1, the energy consumption on sensing is given by $E_k^{(ss)} = p_k^{(ss)} t_k^{(ss)}$.

Next, the energy consumption on local computation at the kth SR is given by

$$E_k^{(co)} = \left(d_k - d_k^{(ol)}\right) C_k p_k^{(co)}, \tag{5}$$

where $p_k^{(co)}$ is the power consumption per CPU computing cycle at the kth SR. Similarly, when the MR executes the computation locally, the energy consumption is $E_M^{(co)} = (d_M - d_M^{ol}) C_M p_M^{(co)}$, where $p_M^{(co)}$ is the power consumption per CPU cycle of computing at the MR.

Finally, for convenience, we define a function $f(x) = N_k(2^{x/B} - 1)$, and according to Equation (3), the energy consumed on data transmission at the kth SR and the MR is given by $E_k^{(tr)} = p_k^{(tr)} t_k^{(ol)} = t_k^{(ol)}/h_k f(d_k^{(ol)}/t_k^{(ol)})$ and $E_M^{(tr)} = p_M^{(tr)} t_M^{(ol)} = t_M^{(ol)}/h_M f(D_M^{(ol)}/t_M^{(ol)})$, respectively.

Based on the above analysis, the total energy consumption at the kth SR and the MR during the task implementation is given by $E_k = E_k^{(ss)} + E_k^{(co)} + E_k^{(tr)}$ and $E_M = E_M^{(co)} + E_M^{(tr)}$, respectively.

3. Problem Formulation

The aim of the proposed MEC-based resource allocation scheme in MRC system is to accomplish time-critical tasks while maintaining the function of the system as long as possible. Then, the key problem is how to management sensing data and offload data in each SRs and MR. Furthermore, we define a weighted factor α_k associated with the kth SR, which can be expressed as $\alpha_k = \min(E_k^{\mathrm{Re}})/E_k^{\mathrm{Re}}, k \in \mathcal{S}$. The factor takes into account the fairness among all SRs. Accordingly, the optimization problem (P1) for the SRs is formulated as

$$\min_{\left\{\mathbf{t}_K^{(ss)}, \mathbf{d}_K^{(ol)}, \mathbf{t}_K^{(ol)}\right\}} \sum_{k=1}^{K} \alpha_k E_k \tag{6}$$

$$s.t. \sum_{k=1}^{K} d_k \geq D \tag{7a}$$

$$\frac{\left(d_k - d_k^{(ol)}\right)}{W_k} \leq T_s^{(co)}, k \in \mathcal{S} \tag{7b}$$

$$E_k^{(\mathrm{Re})} - E_k \geq 0, k \in \mathcal{S} \tag{7c}$$

$$T^{(ss)} \geq t_k^{(ss)} \geq 0, k \in \mathcal{S} \tag{7d}$$

$$T_s^{(co)} \geq t_k^{(ol)} \geq 0, k \in \mathcal{S} \tag{7e}$$

$$d_k \geq 0, k \in \mathcal{S} \tag{7f}$$

$$d_k^{(ol)} \geq 0, k \in \mathcal{S} \tag{7g}$$

where $E_k^{(\mathrm{Re})}$ denotes the remaining battery energy at the kth SR, $\mathbf{t}_K^s$ is the sensing time vector of SRs which denoted as $\mathbf{t}_K^{(ss)} = [t_i^{(ss)}]_{i=1}^K$, $\mathbf{d}_K^{(ol)}$ is the allocated offloading data vector of SRs which represented as $\mathbf{d}_K^{(ol)} = [d_i^{(ol)}]_{i=1}^K$, and $\mathbf{t}_K^{(ol)}$ is the time vector spent on data offloading with $\mathbf{t}_K^{(ol)} = [t_i^{(ol)}]_{i=1}^K$.

In problem (P1), the objective function is set to a min-max problem to improve fairness among SRs. Constraint (7a) is to guarantee the SRs collect the required number of

An optimal resource allocation scheme.
Initialize: System parameters, MR and BS; set $\eta_k = 0$; set $R = 0$ as the number of executed tasks.
While: MRC system can complete task, i.e., problem (P1) has a feasible solution: The Interior Point Method applied to the (SP1) to obtain the globally optimal solution $\{d_{R,k}^{*}, t_{R,k}^{(ss)*}\}$. And BCD method is used to solve the (SP2) to obtain the globally optimal solution $\{d_{R,k}^{(ol)*}, t_{R,k}^{(ol)*}\}$; *Gradient Descent (GD)* method is applied to update parameters η_k; jump to step 4 until convergence; compute total energy consumption of SRs, and update $E_{R,k}^{(\mathrm{Re})} = E_{R,k}^{(\mathrm{Re})} - E_{R,k}$, which represents the remaining energy of each SR after processing the current task; similarly, get the optimal solution $\{d_{R,M}^{(ol)*}, t_{R,M}^{(ol)*}\}$ by solving the (P2) using BCD method; compute energy consumption $E_{R,M}$ at the MR, and $E_{R,M}^{(\mathrm{Re})} = E_{R,M}^{(\mathrm{Re})} - E_{R,M}$; update $R = R + 1$.

Algorithm 1

data, (7b) ensures that each SR has an upper limit that spend on the local computing. Similarly, (7d) and (7e) represent the time constraints for the first and second stages, respectively. (7c) is to guarantee that the total power consumption at the kth SR is less than its remaining power.

Next, in the third stage, it needs to decide how much data should be offloaded to the BS to minimize the energy consumption. Accordingly, the problem (P2) is written as

$$\min_{\left\{t_M^{(ol)}, d_M^{(ol)}\right\}} \quad E_M \tag{8}$$

$$s.t. E_M^{(\mathrm{Re})} - E_M \geq 0 \tag{9a}$$

$$T_M^{(co)} \geq t_M^{(ol)} \geq 0 \tag{9b}$$

$$\frac{\left(d_M - d_M^{(ol)}\right)}{W_M} \leq T_M^{(co)} \tag{9c}$$

$$d_M \geq d_M^{(ol)} \geq 0 \tag{9d}$$

where (9a) is to guarantee that the total power consumption at the MR is less than its remaining battery power, (9b) represents the time constraints for the third stages, (9c) ensures that the time MR spend on the local computation should be less than $T_M^{(co)}$, and (9d) is to guarantee that the offloaded data bits are less than the total received data bits from SRs.

4. Proposed Algorithm

Through the analysis of the (P1), we find the energy consumption of SRs is coupled with constraints (7a) and (7c), which makes it complicated to solve the problem. To decouple these two constraints, the Lagrangian dual method is applied. Then, Lagrangian dual function corresponding to (6) is given by

$$L = \sum_{k=1}^{K} \alpha_k E_k + \sum_{k=1}^{K} \eta_k \left(E_k - E_k^{(\mathrm{Re})}\right), \tag{10}$$

where $\boldsymbol{\eta} = [\eta_1, \cdots, \eta_n]^T$ is Lagrangian multipliers.

According to (10), the dual problem is defined as

$$\max_{\{\boldsymbol{\eta} \geq 0\}} \left\{ \min_{\left\{\mathbf{t}_K^{(ss)}, \mathbf{D}_K^{(ol)}, \mathbf{t}_K^{(ol)}\right\}} L \right\}, \tag{11}$$

which is convex on $\boldsymbol{\eta}$.

After decoupling, we can split the problem (P3) into three subproblems. First, the subproblem (SP1) is to minimize the weighted energy consumption of SRs during the first stage, which is given by

$$\begin{aligned} \min_{\left\{\mathbf{t}_K^{(ss)}, \mathbf{d}_K\right\}} \quad & \sum_{k=1}^{K} (\alpha_k + \eta_k) E_k^{(ss)} \\ s.t. \quad & (6a), (6d), (6f) \end{aligned} \tag{12}$$

Next, (SP2) is to minimize the weighted energy consumption of SRs during the local computation and task offloading at the second stage, which is given by

$$\begin{aligned} \min_{\left\{\mathbf{d}_K^{(ol)}, \mathbf{t}_K^{(ol)}\right\}} \quad & \sum_{k=1}^{K} (\alpha_k + \eta_k) \left(E_k^{(co)} + E_k^{(tr)}\right) \\ s.t. \quad & (6b), (6e), (6g) \end{aligned} \tag{13}$$

With Lagrange multipliers, $\boldsymbol{\eta}$, (SP1), and (SP2) are all convex optimization problems. In particular, (SP1) is linear programming problems, which can be solved by the Interior Point Method. The *Block Coordinate Descent* (BCD) [25] optimization technique can be used to obtain the optimal solution of (SP2). After achieving the optimal solution of the subprime problem $E^*, E_k^{(ss)*}, E_k^{(co)*}, E_k^{(tr)*}$, the Lagrange dual problem is a linear programming problem, which is formulated as

$$\begin{aligned} \max_{\{\boldsymbol{\eta}\}} \quad & \left(\mathbf{t}_K^{(ss)}, \mathbf{d}_K^{(ol)}, \mathbf{t}_K^{(ol)}, \eta\right) \\ s.t. \quad & \eta_k \geq 0, k \in \mathcal{S}. \end{aligned} \tag{14}$$

The updating rule in the following Algorithm 1 can be applied to derive the optimal η.

Table 1: Simulation Parameters.

Number of SR			3
Number of MR			1
The initial battery energy of each SR			[3, 4, 1] J
The initial battery energy of MR			10 J
Low power mode threshold			0.2 J
T_s	800 ms	$T_s^{(co)}$	40 ms
B_k, B_M	10 MHz	$T_M^{(co)}$	10 ms
e_k^s	$\in[18, 22]$ nJ/bit	D	5×10^6 bits
C_k	$\in[100, 1000]$ cycles/bit	$p_k^{(ss)}$	$\in[0.05,0.13]$ W
F_k	$[100, \cdots, 800]$ MHz	F_M	2.4 GHz
$p_M^{(co)}$	1000×10^{-11} J/cycle	W_M	2.4×10^9 cycles/s
N_k	$\in[-86,-96]$ dBm	C_M	100 cycles/bit
W_k	$\in[100, \cdots, 800] \times 10^6$ cycles/s		
$p_k^{(co)}$	$\in[100,500] \times 10^{-11}$ cycles/bit		
Pass loss model	$15.3 + \alpha \times 10 \log_{10}(Distance)$, $\alpha = 3.75$, $\alpha = 5$,distance $\in$ [1,100] m		

Based on the Interior Point Method, the solution of (P2) can be achieved and the optimal task offloading and resource management scheme can be derived for the MR since it is a convex optimization problem. According to the above analysis, the optimal joint computation and communication resource scheme is concluded in Algorithm 1.

5. Numerical Results

In this section, the performance of proposed task and resource allocation scheme is investigated using computer simulation on MATLAB. We will refer to our scheme, i.e., Algorithm 1 as "ROOP." In order to reflect the performance of the algorithm, the robust scheme, which named as "MRC-RP," we proposed in [23], is considered. The key parameters used in the simulations are listed in Table 1.

5.1. ROOP Scheme Performance. Figure 3 shows the sensing data size allocation and offloading data size versus the number of accomplished tasks in ROOP. On the one hand, since we define a fairness factor α_k, the SR3 with the least battery will hardly participate in the task to save the battery energy. On the other hand, due to good channel conditions of SR2 in this simulation scenario, their transmission energy consumption per bit is lower than local computing. Therefore, SR2 are more inclined to offload more data to the MR. Furthermore, the time limit has a great constraint on the offloading of the SR1, since the SR1 has a bad channel state. Then, the results show that SR1 is not willing to offload.

Figure 4 demonstrates the remaining energy at each SR versus the number of implementing tasks. It can be observed that, in ROOP, since SR3 is assigned more tasks, it consumes more energy than MRC-RP. However, SR3 consumes less

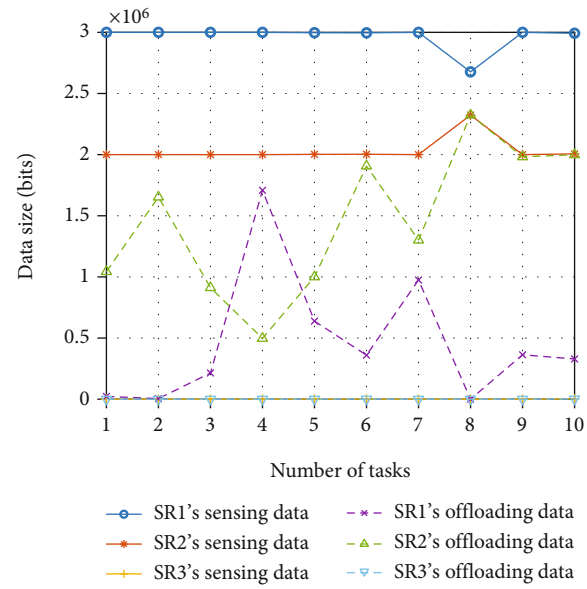


Figure 3: . The sensing data size allocation and data offloading.

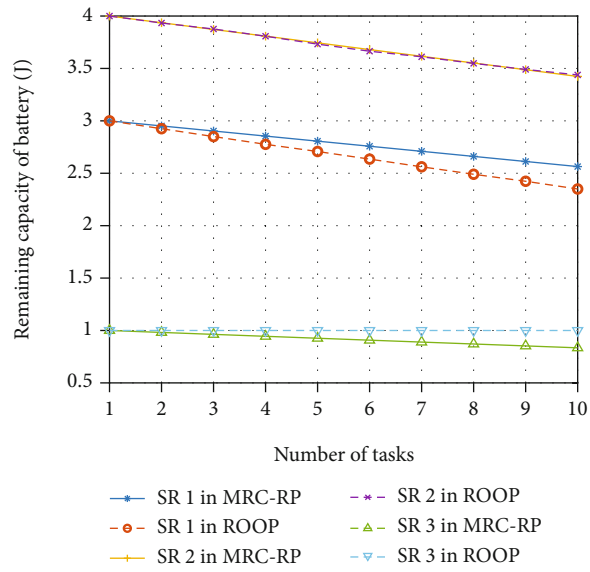


Figure 4: . The remaining battery energy corresponding to the three SRs.

energy to performing tasks than SR1 and SR2, which means the ROOP scheme is more friendly to the energy-less robots. We can observe in Figure 3 that the overall energy consumption in ROOP is smaller than MRC-RP. And as the number of accomplished tasks increases, more energy would be consumed and the remaining energy at the SRs decreases as well.

5.2. Performance Comparison. Figure 5 depicts the total energy consumption of SRs that complete different numbers of tasks. We consider the baseline *greedy offloading policy* (GOP) for comprehensive performance comparisons. The specific strategy of the GOP is that the SR will offload as many task bits as time permits. We observe that the total

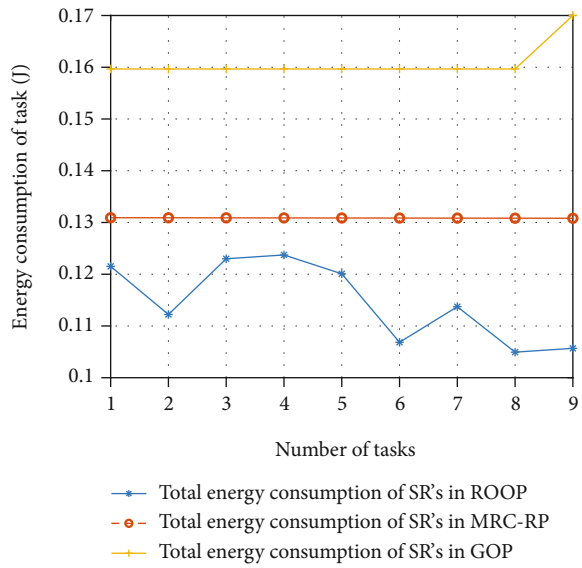


FIGURE 5: The comparison of energy consumption by ROOP, MRC-RP, and GOP.

energy consumption of SR for GOP and MRC-RP schemes is always much larger than ROOP when the number of tasks is accumulated, which verifies the effectiveness of our task and resource allocation scheme.

6. Conclusion

This work studies task and resource allocation for MRC system. First, a task implementation framework is proposed for MRC system based on MEC. Next, aiming to save the energy of SRs and MR and prolong the MRC system function time, we proposed a time-critical and computation-intensive resource allocation scheme for MEC-based MRC system, in which an MR acts as an edge server to provide computation and communication services to SRs. Simulation results revealed that proposed scheme greatly outperforms the baseline.

Data Availability

No data were used to support the study.

Conflicts of Interest

There is no conflict of interest regarding the publication of this paper.

Acknowledgments

This work is supported by the Science and Technology Project of State Grid Zhejiang Electric Power Co., Ltd. (2021ZK17).

References

[1] Y. Liu, X. Liu, X. Gao et al., "Robotic communications for 5g and beyond: challenges and research opportunities," 2020, http://arxiv.org/abs/2012.05093.

[2] Z. H. Ismail and N. Sariff, "A survey and analysis of cooperative multi-agent robot systems: challenges and directions," *Applications of Mobile Robots*, pp. 8–14, 2018.

[3] G. Bedi, G. K. Venayagamoorthy, R. Singh, R. R. Brooks, and K.-C. Wang, "Review of internet of things (IoI) in electric power and energy systems," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 847–870, 2018.

[4] H. Kim, H. Nam, W. Jung, and J. Lee, "Performance analysis of CNN frameworks for gpus," *IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*, vol. 2017, 2017, pp. 55–64, Santa Rosa, CA, USA, April 2017.

[5] Y. Mostofi, "Cooperative wireless-based obstacle/object mapping and see-through capabilities in robotic networks," *IEEE Transactions on Mobile Computing*, vol. 12, no. 5, pp. 817–829, 2013.

[6] K.-C. Chen and H.-M. Hung, "Wireless robotic communication for collaborative multi-agent systems," in *International Conference on Communications (ICC)*, pp. 1–7, Shanghai, China, May 2019.

[7] B. Kehoe, S. Patil, P. Abbeel, and K. Goldberg, "A survey of research on cloud robotics and automation," *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 2, pp. 398–409, 2015.

[8] L. Riazuelo, J. Civera, and J. M. Montiel, "C2TAM: a cloud framework for cooperative tracking and mapping," *Robotics and Autonomous Systems*, vol. 62, no. 4, pp. 401–413, 2014.

[9] Q. Hu, Y. Cai, G. Yu, Z. Qin, M. Zhao, and G. Y. Li, "Joint offloading and trajectory design for UAV-enabled mobile edge computing systems," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1879–1892, 2018.

[10] H. Lin, S. Zeadally, Z. Chen, H. Labiod, and L. Wang, "A survey on computation offloading modeling for edge computing," *Journal of Network and Computer Applications*, vol. 169, p. 102781, 2020.

[11] C. Wu, Z. Liu, F. Liu, T. Yoshinaga, Y. Ji, and J. Li, "Collaborative learning of communication routes in edge-enabled multi-access vehicular environment," *IEEE Transactions on Cognitive Communications and Networking*, vol. 6, no. 4, pp. 1155–1165, 2020.

[12] M. Zhao, W. Li, L. Bao, J. Luo, Z. He, and D. Liu, "Fairness-aware task scheduling and resource allocation in UAV-enabled mobile edge computing networks," *IEEE Transactions on Green Communications and Networking*, vol. 5, no. 4, pp. 2174–2187, 2021.

[13] Y. Siriwardhana, P. Porambage, M. Liyanage, and M. Ylianttila, "A survey on mobile augmented reality with 5G mobile edge computing: architectures, applications, and technical aspects," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 2, pp. 1160–1192, 2021.

[14] X. Chen, C. Wu, T. Chen et al., "Information freshness-aware task offloading in air-ground integrated edge computing systems," *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 1, pp. 243–258, 2022.

[15] X. Chen, C. Wu, Z. Liu, N. Zhang, and Y. Ji, "Computation offloading in beyond 5G networks: a distributed learning framework and applications," *IEEE Wireless Communications*, vol. 28, no. 2, pp. 56–62, 2021.

[16] C. Wu, X. Chen, T. Yoshinaga, Y. Ji, and Y. Zhang, "Integrating licensed and unlicensed spectrum in the internet of vehicles with mobile edge computing," *IEEE Network*, vol. 33, no. 4, pp. 48–53, 2019.

[17] C. You, K. Huang, H. Chae, and B.-H. Kim, "Energy-efficient resource allocation for mobile-edge computation offloading," *IEEE Transactions on Wireless Communications*, vol. 16, no. 3, pp. 1397–1411, 2016.

[18] C. Wang, C. Liang, F. R. Yu, Q. Chen, and L. Tang, "Computation offloading and resource allocation in wireless cellular networks with mobile edge computing," *IEEE Transactions on Wireless Communications*, vol. 16, no. 8, pp. 4924–4938, 2017.

[19] J. Ren, G. Yu, Y. He, and G. Y. Li, "Collaborative cloud and edge computing for latency minimization," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 5, pp. 5031–5044, 2019.

[20] M. Chen and Y. Hao, "Task offloading for mobile edge computing in software defined ultra-dense network," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 3, pp. 587–597, 2018.

[21] X. Chen, L. Jiao, W. Li, and X. Fu, "Efficient multi-user computation offloading for mobile-edge cloud computing," *IEEE/ACM Transactions on Networking*, vol. 24, no. 5, pp. 2795–2808, 2015.

[22] Y. Shen, R. Yin, H. Zhu, X. Chen, and C. Wu, "Resource management in MEC based muti-robot cooperation systems," in *International Conference on Information and Communication Technologies for Disaster Management (ICT-DM)*, pp. 30–37, Hangzhou, China, Dec. 2021.

[23] R. Yin, Y. Shen, H. Zhu, X. Chen, and C. Wu, "Time-critical tasks implementation in MEC based multi-robot cooperation systems," 2021, http://arxiv.org/abs/2111.11038.

[24] J. Zhu and S. Papavassiliou, "On the energy-efficient organization and the lifetime of multi-hop sensor networks," *IEEE Communications Letters*, vol. 7, no. 11, pp. 537–539, 2003.

[25] M. Hong, M. Razaviyayn, Z.-Q. Luo, and J.-S. Pang, "A unified algorithmic framework for block-structured optimization involving big data: with applications in machine learning and signal processing," *IEEE Signal Processing Magazine*, vol. 33, no. 1, pp. 57–77, 2016.

Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 6772467, 11 pages
https://doi.org/10.1155/2022/6772467

Research Article

FAHP-Based Reliability Evaluation of Distributed IoT Devices in a Distribution Power Grid

Xinhong You, Pengping Zhang, Shuai Li, Feng Wang, Guoqiang Su, and Shidong Zhang

State Grid Shandong Electric Power Research Institute, Jinan, China

Correspondence should be addressed to Xinhong You; sddky_coap@163.com

Received 3 March 2022; Revised 17 March 2022; Accepted 4 April 2022; Published 6 May 2022

Academic Editor: Celimuge Wu

Reliable operation of distributed internet of things (IoT) devices is essential for supporting two-way interaction between energy flow and information flow in a distribution power grid. Traditional reliability evaluation methods suffer from several challenges including complex evaluation indicator selection and inadaptability of fixed weights with actual reliability requirements. In this paper, we first establish a multilevel reliability evaluation indicator system, where the indicators are selected through joint consideration of comprehensiveness and effectiveness. Then, we propose a fuzzy analytic hierarchy process- (FAHP-) based reliability evaluation method, where a three-layer reliability evaluation architecture is established. Specifically, a 0.1-0.9 scaling method is adopted to establish the fuzzy judgment matrix and scale the important relationship among state-layer elements, which can be dynamically adjusted to generate the weights adapting with actual operation state. The scores of distributed IoT devices and corresponding reliability levels can be obtained through the weighted summation of the scores of each layer. Simulation results verify the effectiveness and accuracy of the proposed method through comparing with actual reliability value and two existing evaluation methods.

1. Introduction

With the integration of distributed renewable energy sources, flexible loads, and energy storage devices into the distribution power grid, two-way interaction between energy flow and information flow is essential to improve energy-supply balance and promote renewable energy consumption [1, 2]. Massive distributed internet of things (IoT) devices with strong interaction and perception capabilities are deployed in the distribution power grid to collect operation status and electrical parameters in real time, which provide data support for the optimization of distributed energy management [3]. Therefore, the reliability of distributed IoT devices is vital to the safe and stable operation of a distribution power grid. It is necessary to extract reasonable and valid analysis data from numerous evaluation indicators and build a reliability evaluation system to improve the operation reliability of distributed IoT devices [4, 5].

However, the performance evaluation indicators of distributed IoT devices span numerous categories, and the correlations among different categories of indicators are diverse [6, 7]. Several major critical technical challenges are summarized as follows.

Challenge 1. Difficulty in selecting evaluation indicators. Numerous evaluation indicators exist for distributed IoT devices in a distribution power grid, and the selection of evaluation indicators has a significant impact on the construction of the evaluation indicator system. Selecting insufficient indicators result in inaccurate reliability assessment, while selecting excessive indicators increase the complexity of the whole evaluation process. Therefore, it is a challenge to appropriately select the evaluation indicators to achieve accurate and concise reliability evaluation of distributed IoT devices.

Challenge 2. Difficulty in calculating weights of indicators. Various services of a distribution power grid impose

different reliability requirements of distributed IoT devices. When fixed weights are used to evaluate the reliability of the same indicator under different services or scenarios, the evaluation results cannot reflect the real reliability performance of the IoT devices. Therefore, how to determine the appropriate weights of indicators under different scenarios is another challenge.

There exist some works on the performance evaluation of IoT devices. In [8], Yuwen proposed a comprehensive evaluation indicator system to evaluate the safety performance of a communication system in a distribution power grid. In [9], Xiao et al. proposed a principal component analysis-based sensor performance evaluation method to achieve online monitoring of sensor faults. However, these works do not consider reliability evaluation, which are not suitable for a distribution power grid with high-reliability requirements on distributed IoT devices. Moreover, the expert evaluation method determines indicator weights according to individual expert preferences in performance evaluation, which cannot well handle the ambiguity of the indicators due to experts' personal subjective preferences.

The fuzzy analytic hierarchy process (FAHP) eliminates the influence of experts' personal subjective preferences on weight determination by fuzzy processing of indicator weights. It provides a solution for quantitative and qualitative analysis of multiobjective decision-making problems. In [10], Yuan proposed a comprehensive evaluation method of power quality based on incentive and punishment mechanism. FAHP was utilized to calculate the comprehensive value to evaluate power quality. In [11], Zeng et al. proposed a FAHP-based intelligent evaluation method for IoT devices in distribution station. A hierarchical structure model was established to quantitatively evaluate the importance of performance indicators. In [12], Li proposed a FAHP-based comprehensive evaluation system to evaluate multidimensional performance of navigation IoT devices, such as economy, reliability, and security. However, these works only consider a few evaluation indicators and ignore the impact of the order of magnitudes of different indicators, which are not suitable for the reliability evaluation of distributed IoT devices in a distribution power grid.

Motivated by the above challenges, we construct a reliability evaluation indicator system of distributed IoT devices in a distribution power grid and propose a FAHP-based reliability evaluation method. First, a three-layer reliability evaluation architecture including a target layer, indicator layer, and state layer is established [13]. Secondly, state-layer elements are classified according to the relationship between them and reliability, and corresponding membership functions are constructed to obtain the scores of state-layer elements. Then, the scores of indicator-layer evaluation indicators can be obtained through establishing the state-layer fuzzy judgment matrix, performing consistency check and consistency transmission, calculating the weights of state-layer elements, and calculating the weighted sum of weights and scores of elements. Similarly, the target-layer score can be obtained, which is utilized to determine the reliability level of distributed IoT devices and complete the reliability evaluation. According to the reliability levels and scores of three layers, the abnormalities of IoT devices can be inferred to provide guidance for maintenance [14]. The main contributions are introduced as follows.

Contribution 1. Multilevel reliability evaluation indicator system. A multilevel reliability evaluation indicator system for distributed IoT devices is proposed. The system includes four categories of first-level indicators and thirty-six second-level indicators, which are selected through the comprehensive consideration of the ability to reflect the actual operation state and the actual characteristics such as the difficulty level of data collection and the calculation complexity.

Contribution 2. FAHP-based weight calculation method. Considering that the importance of each state-layer element for reliability evaluation is constantly changing in the actual distribution power grid scenario, the proposed FAHP-based weight calculation method can dynamically adjust the relative importance of elements through adopting the 0.1-0.9 scaling method to establish the fuzzy judgment matrix, which improves the adaptability of element weight with actual reliability requirements.

The remainder of this paper is organized as follows. Section 2 introduces the reliability evaluation indicator system of distributed IoT devices in a distribution power grid. Section 3 presents FAHP-based reliability evaluation. The simulation results are presented in Section 4. Finally, Section 5 concludes this paper.

2. Reliability Evaluation Indicator System of Distributed IoT Devices in Distribution Power Grid

The reliability of distributed IoT devices in distribution power grid is mainly manifested in the ability of survival, effective communication, and accurate monitoring in a specific application environment and within a specified time period [15]. It also emphasizes the ability to deal with emergencies within a specific range and ensure its stable operation. The reliability evaluation indicator system needs to accurately describe all kinds of indicators affecting the reliability of distributed IoT devices [16]. At the same time, it should also specify the details of the hierarchical relationship and indicator weight of the indicator system [17]. Moreover, the evaluation accuracy of the reliability evaluation system is also directly affected by the selection strategy of evaluation indicators [18, 19]. On the one hand, the selected indicators are required to reflect the actual operation of IoT devices as much as possible, so that important indicators cannot be omitted [20]. On the other hand, the actual characteristics such as the difficulty level of data collection, the effectiveness of information, and the calculation complexity should be considered to improve the evaluation accuracy and reduce the evaluation complexity as much as possible.

There is no unified reliability evaluation standard of distributed IoT devices in a distribution power grid [21–23]. Following the principles of objectivity and comparability, we establish a multilevel reliability indicator system, including four first-level indicators and thirty-six second-level

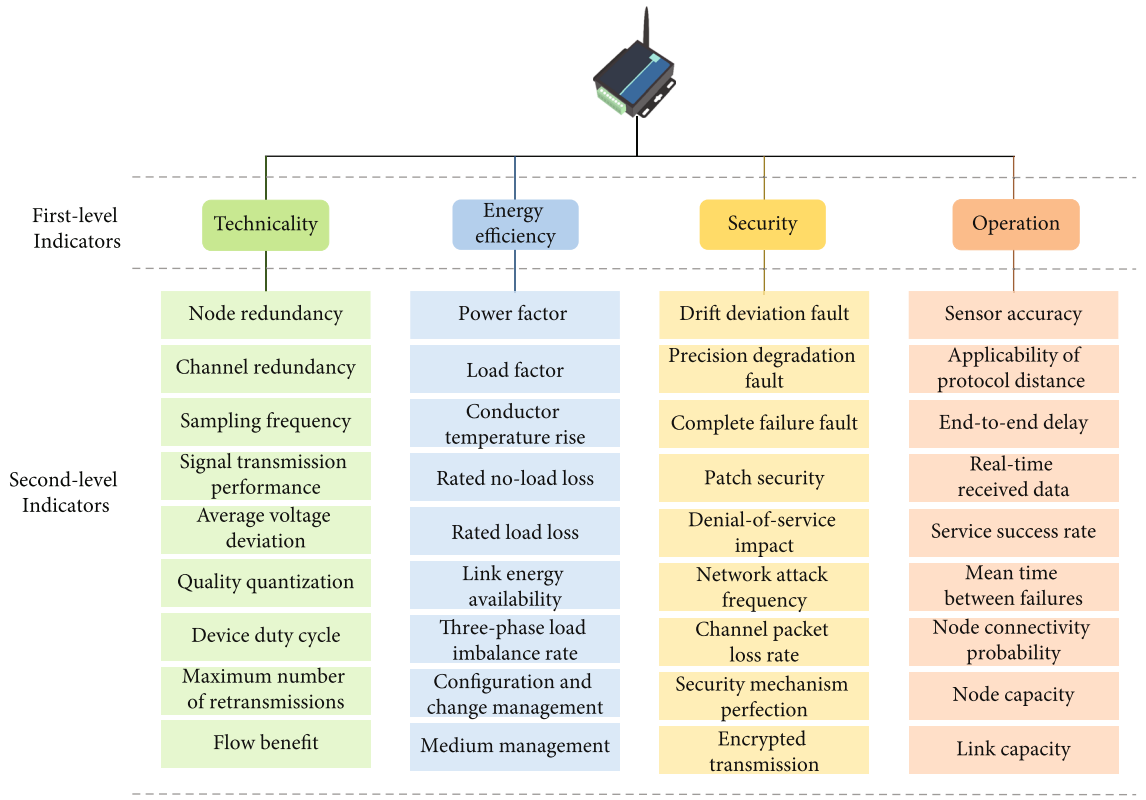


Figure 1: Multilevel reliability evaluation indicator system of distributed IoT devices in a distribution power grid.

indicators. As shown in Figure 1, the first-level indicators are divided into four categories, i.e., technicality evaluation indicator, energy efficiency evaluation indicator, security evaluation indicator, and operation evaluation indicator. The second-level indicators contained in each first-level indicator are provided as follows:

(i) Technicality evaluation indicator. The technicality evaluation indicator contains nine second-level indicators, i.e., node redundancy, channel redundancy, sampling frequency, signal transmission performance, average voltage deviation, quality quantization, device duty cycle, maximum number of retransmissions, and flow benefit

(ii) Energy efficiency evaluation indicator. The energy efficiency evaluation indicator contains nine second-level indicators, i.e., power factor, load factor, conductor temperature rise, rated no-load loss, rated load loss, link energy availability, three-phase load imbalance rate, configuration and change management, and medium management

(iii) Security evaluation indicator. The security evaluation indicator contains nine second-level indicators, i.e., drift deviation fault, precision degradation fault, complete failure fault, patch security, denial-of-service impact, network attack frequency, channel packet loss ratio, security mechanism perfection, and encrypted transmission

(iv) Operation evaluation indicator. The operation evaluation indicator contains nine second-level indicators, i.e., sensor accuracy, applicability of protocol distance, end-to-end delay, real-time received data, service success rate, mean time between failures, node connectivity probability, node capacity, and link capacity

The multilevel reliability evaluation indicator system proposed in this paper can be applied to more evaluation indicator scenarios by adjusting the categories of first-level indicators and second-level indicators.

3. FAHP-Based Reliability Evaluation of Distributed IoT Devices in Distribution Power Grid

3.1. Three-Layer Reliability Evaluation Architecture of Distributed IoT Devices. Based on the reliability evaluation indicator system proposed in Section 2, we establish a three-layer reliability evaluation architecture of distributed IoT devices, including the target layer, indicator layer, and state layer as follows:

(i) Target layer. The target layer is defined as the distributed IoT devices considering that the ultimate target is to evaluate the reliability of distributed IoT devices

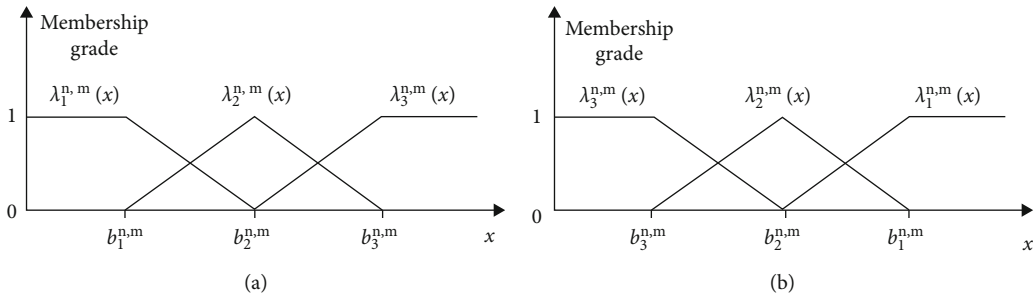


Figure 2: Membership functions.

(ii) Indicator layer. The indicator layer is composed of four types of evaluation indicators corresponding to the first-level indicators defined in Section 2, i. e., technicality evaluation indicator, energy efficiency evaluation indicator, security evaluation indicator, and operation evaluation indicator

(iii) State layer. The state layer consists of thirty-six elements corresponding to the second-level indicators defined in Section 2. Each type of evaluation indicator in the indicator layer includes nine state-layer elements, and the details are elaborated in Section 2

Based on the established three-layer reliability evaluation architecture, we denote the four types of evaluation indicators in the indicator layer as $\{C_n, n = 1, 2, 3, 4\}$, respectively. The state-layer elements included in the evaluation indicator type C_n are denoted as $\{e_n^m, m = 1, 2, \cdots, 9\}$. For instance, C_1 represents technicality evaluation indicators, and e_1^m, $m = 1, 2, \cdots, 9$, represent the node redundancy, channel redundancy, sampling frequency, signal transmission performance, average voltage deviation, quality quantization, device duty cycle, maximum number of retransmissions, and flow benefit, respectively.

3.2. State-Layer Scoring. The scoring method for state-layer elements in the three-layer reliability evaluation architecture of distributed IoT devices is illustrated as follows. Membership function is adopted to realize the mapping from the element values to the element scores, where the membership of an element to a certain fuzzy evaluation is calculated based on the element value, and then, the element score is calculated based on fuzzy evaluation scores and memberships. Membership function-based scoring method for state-layer elements can effectively reduce the influence of human subjective factors on the reliability evaluation of distributed IoT devices [24]. Considering the differences in the units and order of magnitudes of state-layer elements, membership functions with different parameters are constructed for different elements to realize unified scoring.

Based on the requirements on the operation state of different distributed IoT devices in the actual distribution power grid, the fuzzy evaluations for the reliability of a state-layer element are defined as poor, medium, and good. The membership functions corresponding to three fuzzy evaluations are denoted as $\lambda_1^{n,m}$, $\lambda_2^{n,m}$, and $\lambda_3^{n,m}$ for the state-layer element e_n^m [25]. Particularly, when the element value helps to improve the reliability of distributed IoT devices, the membership grade to the good fuzzy evaluation will be larger. Considering the relationship between element value and reliability, the state-layer elements are classified to positive correlation elements and negative correlation elements. Specifically, a larger positive correlation element value represents the higher reliability of the distributed sensor devices, and negative correlation elements are the opposite. We propose two membership functions for the above two kinds of elements, which are shown in Figure 2 and introduced as follows:

Membership function (a): the membership function for positive correlation elements.

$$\lambda_1^{n,m}(x) = \begin{cases} 1, & x \le b_1^{n,m}, \\ \dfrac{x - b_2^{n,m}}{b_1^{n,m} - b_2^{n,m}}, & b_1^{n,m} < x \le b_2^{n,m}, \\ 0, & x > b_2^{n,m}, \end{cases}$$

$$\lambda_2^{n,m}(x) = \begin{cases} 0, & x \le b_1^{n,m} \text{ or } x \ge b_3^{n,m}, \\ \dfrac{x - b_1^{n,m}}{b_2^{n,m} - b_1^{n,m}}, & b_1^{n,m} < x \le b_2^{n,m}, \\ \dfrac{x - b_3^{n,m}}{b_2^{n,m} - b_3^{n,m}}, & b_2^{n,m} < x \le b_3^{n,m}, \end{cases} \tag{1}$$

$$\lambda_3^{n,m}(x) = \begin{cases} 0, & x \le b_2^{n,m}, \\ \dfrac{x - b_2^{n,m}}{b_3^{n,m} - b_2^{n,m}}, & b_2^{n,m} < x \le b_3^{n,m}, \\ 1, & x > b_3^{n,m}. \end{cases}$$

Membership function (b): the membership function for negative correlation elements.

$$\lambda_1^{n,m}(x) = \begin{cases} 0, & x \le b_2^{n,m}, \\ \dfrac{x - b_2^{n,m}}{b_1^{n,m} - b_2^{n,m}}, & b_2^{n,m} < x \le b_1^{n,m}, \\ 1, & x > b_1^{n,m}, \end{cases}$$

$$\lambda_2^{n,m}(x) = \begin{cases} 0, & x \le b_3^{n,m} \text{ or } x \ge b_1^{n,m}, \\ \dfrac{x - b_3^{n,m}}{b_2^{n,m} - b_3^{n,m}}, & b_3^{n,m} < x \le b_2^{n,m}, \\ \dfrac{x - b_1^{n,m}}{b_2^{n,m} - b_1^{n,m}}, & b_2^{n,m} < x \le b_1^{n,m}, \end{cases} \quad (2)$$

$$\lambda_3^{n,m}(x) = \begin{cases} 1, & x \le b_3^{n,m}, \\ \dfrac{x - b_2^{n,m}}{b_3^{n,m} - b_2^{n,m}}, & b_3^{n,m} < x \le b_2^{n,m}, \\ 0, & x > b_2^{n,m}. \end{cases}$$

Here, x represents the actual element value, and $\lambda_1^{n,m}(x) + \lambda_2^{n,m}(x) + \lambda_3^{n,m}(x) = 1$. $b_1^{n,m}$, $b_2^{n,m}$, and $b_3^{n,m}$ are the parameters to describe the boundaries of poor, medium, and good fuzzy evaluations for element e_n^m, the values of which are determined by the relationship between reliability and element values [26]. Take the negative correlation element end-to-end delay e_4^3 as an example. We assume that the device is considered to be reliable when the end-to-end delay is lower than 50 ms and unreliable when the end-to-end delay exceeds 100 ms. Then, $b_3^{4,3}$, $b_2^{4,3}$, and $b_1^{4,3}$ can be set as 50 ms, 80 ms, and 100 ms, respectively.

The specific procedures of the state-layer element scoring for distributed IoT devices are summarized in the following:

(1) Determine the membership function based on the state-layer element types. Specifically, membership function (a) is utilized for positive correlation elements, and membership function (b) is utilized for negative correlation elements

(2) According to the selected membership function, calculate the membership grades of the current element value to the good, medium, and poor fuzzy evaluations, i.e., $\lambda_1^{n,m}(x)$, $\lambda_2^{n,m}(x)$, and $\lambda_3^{n,m}(x)$

(3) Score the current element value as

$$f_{n,m}(x) = \lambda_1^{n,m}(x)F_1 + \lambda_2^{n,m}(x)F_2 + \lambda_3^{n,m}(x)F_3, \quad (3)$$

where F_1, F_2, and F_3 are the scores corresponding to good, medium, and poor fuzzy evaluations, the values of which are determined according to the actual operation environment of the distribution power grid

3.3. Indicator-Layer Scoring. The scoring procedures for the indicator-layer evaluation indicators in the three-layer reliability evaluation architecture of distributed state IoT devices consist of establishment of state-layer fuzzy judgment matrix, consistency check, and consistency transformation, as well as weight calculation of state layer and scoring of indicator layer, which are elaborated as follows.

Table 1: 0.1-0.9 scaling method.

Scaling	Meaning
0.1	e_n^i is extremely important compared with e_n^j
0.2	e_n^i is strongly more important compared with e_n^j
0.3	e_n^i is obviously more important compared with e_n^j
0.4	e_n^i is slightly more important compared with e_n^j
0.5	e_n^i and e_n^j have equal importance
0.6	e_n^j is slightly more important compared with e_n^i
0.7	e_n^j is obviously more important compared with e_n^i
0.8	e_n^j is strongly more important compared with e_n^i
0.9	e_n^j is extremely important compared with e_n^i

3.3.1. Establishment of State-Layer Fuzzy Judgment Matrix. The state-layer fuzzy judgment matrix is defined to describe the important relationships between the state-layer elements belonging to a certain evaluation indicator of the indicator layer. Specifically, the state-layer fuzzy judgment matrix for the evaluation indicator C_n is denoted as $R_n^B = (r_{i,j}^{n,B})_{M\times M}$ and is given by

$$R_n^B = \begin{bmatrix} r_{1,1}^{n,B} & \cdots & \cdots & \cdots & r_{1,M}^{n,B} \\ \vdots & \ddots & \cdots & \cdots & \vdots \\ \vdots & \vdots & r_{i,j}^{n,B} & \vdots & \vdots \\ \vdots & \cdots & \cdots & \ddots & \vdots \\ r_{M,1}^{n,B} & \cdots & \cdots & \cdots & r_{M,M}^{n,B} \end{bmatrix}, \quad (4)$$

where M represents the number of state-layer elements contained in C_n. $r_{i,j}^{n,B}$ represents the importance of the element e_n^i relative to the element e_n^j in C_n, where $i, j = 1, 2, \cdots, M$.

In this paper, the 0.1-0.9 scaling method is adopted to scale the important relationship $r_{i,j}^{n,B}$. The specific scaling method is shown in Table 1.

For example, when e_n^i is extremely important compared with e_n^j, $r_{i,j}^{n,B} = 0.1$.

3.3.2. Consistency Check and Consistency Transformation. In the process of reliability evaluation of distributed IoT devices, the weight calculation of state-layer elements requires the consistency of the fuzzy judgment matrix. Consistency check and consistency transformation are introduced as follows:

Consistency check: if the fuzzy judgment matrix $R_n^B = (r_{i,j}^{n,B})_{M\times M}$ satisfies the consistency condition, i.e.,

$$r_{i,j}^{n,B} = r_{i,k}^{n,B} - r_{j,k}^{n,B} + 0.5, \quad \forall i, j, k = 1, 2, \cdots, M, \quad (5)$$

the fuzzy judgment matrix is called fuzzy consistency judgment matrix, which is denoted as $R_n^A = R_n^B$.

Consistency transformation: if the consistency condition is not satisfied, perform consistency transformation to transform the fuzzy judgment matrix R_n^B to the fuzzy consistency judgment matrix $R_n^A = (r_{i,j}^{n,A})_{M\times M}$, where

$$r_{i,j}^{n,A} = \frac{\sum_{j=1}^{M} r_{i,j}^{n,B} - \sum_{i=1}^{M} r_{i,j}^{n,B}}{2M} + 0.5. \tag{6}$$

3.3.3. Weight Calculation of State Layer and Scoring of Indicator Layer. The weights of state-layer elements are calculated according to the fuzzy consistency judgment matrix obtained by (6), which are integrated with the scores of state-layer elements obtained in Section 3.2 to calculate the scores of evaluation indicators in the indicator layer.

For the state-layer elements contained in the evaluation indicator C_n, the relationship between the weights and the fuzzy consistency judgment matrix R_n^A is given by

$$w_n^i - w_n^j = \frac{r_{i,j}^{n,A}}{a}, \tag{7}$$

where w_n^i and w_n^j represent the weights of state-layer elements e_n^i and e_n^j, respectively. a is a random parameter which satisfies $a \geq (M-1)/2$.

Then, the least square method is adopted to calculate the weights of the state-layer elements. Specifically, the weight of e_n^i is given by

$$w_n^i = \frac{1}{M} - \frac{1}{2a} + \frac{1}{Ma}\sum_{j=1}^{M} r_{i,j}^{n,A}. \tag{8}$$

Finally, based on the weights and scores of state-layer elements, the score of the evaluation indicator C_n is calculated as

$$X_n = \sum_{i=1}^{M} f_n^i \times w_n^i. \tag{9}$$

3.4. Target-Layer Scoring and Reliability Evaluation. Similar to the indicator-layer scoring, the procedures of target-layer scoring include the establishment of indicator-layer fuzzy judgment matrix, consistency check, and consistency transformation, as well as weight calculation of indicator layer and scoring of the target layer. The reliability evaluation of distributed IoT devices is performed based on the target-layer score. The details are introduced as follows.

3.4.1. Establishment of Indicator-Layer Fuzzy Judgment Matrix. According to the scores of the evaluation indicators obtained in subsection 3.3, establish the indicator-layer fuzzy judgment matrix R_P based on variable weight method to describe the important relationship between the indicator-layer evaluation indicators, which is given by

Table 2: Corresponding relationship between the target-layer score and the reliability levels.

Label	Reliability level	Target-layer score
	Poor	0-20
	Medium	20-50
	Good	50-85
	Excellent	85-100

$$R_P = \begin{bmatrix} r_{1,1}^P & \cdots & \cdots & \cdots & r_{1,M'}^P \\ \vdots & \ddots & \cdots & \cdots & \vdots \\ \vdots & \vdots & r_{i,j}^P & \vdots & \vdots \\ \vdots & \cdots & \cdots & \ddots & \vdots \\ r_{M',1}^P & \cdots & \cdots & \cdots & r_{M',M'}^P \end{bmatrix}, \tag{10}$$

where $r_{i,j}^P = X_j/(X_i + X_j)$, and M' represents the number of evaluation indicators.

3.4.2. Consistency Check and Consistency Transformation. Perform consistency check on R_P based on (5). If R_P does not satisfy the consistency condition, transform R_P to the indicator-layer fuzzy consistency judgment matrix $R'_P = (r_{i,j}^{P'})_{M'\times M'}$, where

$$r_{i,j}^{P'} = \frac{\sum_{j=1}^{M'} r_{i,j}^P - \sum_{i=1}^{M'} r_{i,j}^{P'}}{2M'} + 0.5. \tag{11}$$

3.4.3. Weight Calculation of Indicator Layer and Scoring of Target Layer. Similar to (8), the weight of the evaluation indicator C_n can be calculated as

$$w_n = \frac{1}{M'} - \frac{1}{2a'} + \frac{1}{M'a'}\sum_{j=1}^{M}{}' r_{i,j}^{P'}, \tag{12}$$

where a' is a random parameter which satisfies $a' \geq (M'-1)/2$.

Based on the weights and scores of indicator-layer evaluation indicators, the target-layer score is calculated as

$$S = \sum_{n=1}^{M}{}' w_n X_n. \tag{13}$$

3.4.4. Reliability Evaluation. The reliability level is determined based on the target-layer score to achieve the reliability evaluation of distributed IoT devices. According to the characteristics of the operation environment of the distribution power grid, the reliability levels include excellent, good, medium, and poor [27]. The corresponding relationship between the target-layer score and the reliability levels is shown in Table 2. In other practical application scenarios with higher requirements on the reliability of distributed IoT devices, the reliability levels can be further finely

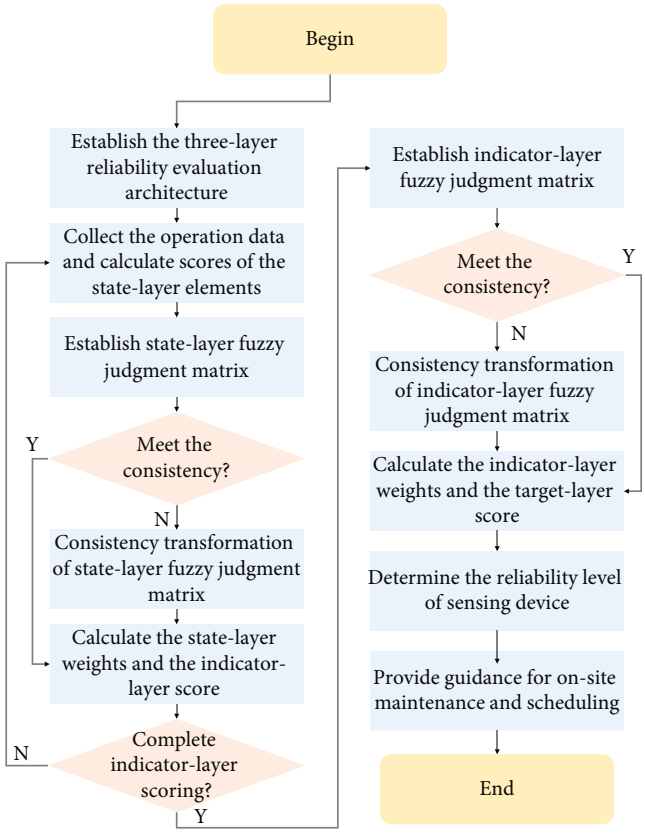


Figure 3: Implementation process of FAHP-based reliability evaluation of distributed IoT devices.

divided. The on-site inspectors can judge the reliability levels of the distributed IoT devices according to the final target-layer score [28]. In addition, for the devices with poor reliability level, the state-layer and indicator-layer scores can also provide maintenance guidance for on-site inspectors.

3.5. Implementation Process of FAHP-Based Reliability Evaluation. The implementation process of FAHP-based reliability evaluation of distributed IoT devices is shown in Figure 3 and is introduced as follows:

(i) *Step 1.* Establish the three-layer reliability evaluation architecture of distributed IoT devices, and determine the composition of each layer

(ii) *Step 2.* Select different membership functions for different types of state-layer elements. Collect the current operation data of state-layer elements, calculate the membership grades to different fuzzy evaluations, and calculate scores of the state-layer elements based on (3)

(iii) *Step 3.* Establish a state-layer fuzzy judgment matrix based on the 0.1-0.9 scaling method

(iv) *Step 4.* Perform consistency check and consistency transformation based on (5) and (6) to obtain the state-layer fuzzy consistency judgment matrix

Table 3: The scores of state-layer elements of technicality evaluation indicator.

State-layer elements	Score
Node redundancy	84
Channel redundancy	92
Sampling frequency	78
Signal transmission performance	94
Average voltage deviation	86
Quality quantification	90
Device duty cycle	88
Maximum number of retransmissions	98
Traffic benefit	93

(v) *Step 5.* Calculate the weights of state-layer elements based on (8), and calculate the score of the indicator-layer evaluation indicators based on (9)

(vi) *Step 6.* Judge whether all evaluation indicators of the indicator layer have been scored. If not, return to *Step 2* to calculate the next evaluation indicator. If yes, proceed to the next step

(vii) *Step 7.* Establish indicator-layer fuzzy judgment matrix based on (10), and carry out consistency check and consistency transformation to obtain the indicator-layer fuzzy consistency judgment matrix

(viii) *Step 8.* Calculate the weights of evaluation indicators based on (12), and calculate the target-layer score based on (13)

(ix) *Step 9.* Determine the reliability levels and complete the reliability evaluation of distributed IoT devices based on Table 2. According to the reliability levels and scores of three layers, analyze the operation state and abnormal faults, and provide guidance for on-site maintenance and scheduling

4. Simulation Results

To verify the accuracy and practicability of the proposed FAHP-based reliability evaluation method, the on-site operation data of an IoT device in the distribution power grid of Shandong Province, China, are collected to perform reliability evaluation.

4.1. Simulation Verification. In this subsection, we introduce the simulation verification process of the FAHP-based reliability evaluation method. Due to the space limitation, we only take the technicality evaluation indicator C_1 and its contained state-layer elements as an example.

4.1.1. Membership Function Selection and State-Layer Scoring. Judge the type of each state-layer element of the technicality evaluation indicator and select the corresponding membership function. For instance, the signal transmission performance is directly proportional to the reliability, which is a positive correlation element, and thus,

membership function 1 is selected. The node redundancy is inversely proportional to the reliability, which is a negative correlation element, and thus, membership function 2 is selected. The boundary parameters are set based on the characteristics of each element [29]. Then, the scores of state-layer elements of technicality evaluation indicator can be obtained based on (3), as shown in Table 3.

4.1.2. State-Layer Fuzzy Judgment Matrix of Technicality Evaluation Indicator. Based on the 0.1-0.9 scaling method, the state-layer fuzzy judgment matrix of C_1 is given by

$$R_1^B = \begin{bmatrix} 0.5 & 0.0156 & 0.2 & 0.2222 & 0.0144 & 0.0148 & 0.0145 & 0.1152 & 0.0982 \\ 0.99844 & 0.5 & 0.9403 & 0.9474 & 0.4791 & 0.4865 & 0.4809 & 0.4754 & 0.4923 \\ 0.8 & 0.0597 & 0.5 & 0.5333 & 0.0552 & 0.0567 & 0.0556 & 0.0744 & 0.0567 \\ 0.7778 & 0.0526 & 0.4667 & 0.5 & 0.0486 & 0.05 & 0.049 & 0.0388 & 0.0432 \\ 0.9856 & 0.5209 & 0.9448 & 0.9514 & 0.5 & 0.5074 & 0.5018 & 0.5148 & 0.5164 \\ 0.9852 & 0.5135 & 0.9433 & 0.95 & 0.4926 & 0.5 & 0.4944 & 0.4898 & 0.4934 \\ 0.9855 & 0.5191 & 0.9444 & 0.9510 & 0.4982 & 0.5056 & 0.5 & 0.5351 & 0.5126 \\ 0.8848 & 0.5246 & 0.9256 & 0.9612 & 0.4852 & 0.5102 & 0.4649 & 0.5 & 0.4876 \\ 0.9018 & 0.5077 & 0.9433 & 0.9568 & 0.4836 & 0.5066 & 0.4874 & 0.5124 & 0.5 \end{bmatrix}. \tag{14}$$

According to (5), R_1^B does not meet the consistency condition. Therefore, it is transformed into a fuzzy consistency judgment matrix based on (6), which is given by

$$R_1^A = \begin{bmatrix} 0.5 & 0.24414 & 0.44463 & 0.45379 & 0.23621 & 0.24071 & 0.23574 & 0.24727 & 0.24418 \\ 0.75586 & 0.5 & 0.70049 & 0.70965 & 0.49207 & 0.49656 & 0.4916 & 0.50312 & 0.50004 \\ 0.55537 & 0.29951 & 0.5 & 0.50916 & 0.29158 & 0.29608 & 0.29112 & 0.30264 & 0.29956 \\ 0.54621 & 0.29035 & 0.49084 & 0.5 & 0.28242 & 0.28692 & 0.28196 & 0.29348 & 0.29039 \\ 0.76379 & 0.50793 & 0.70842 & 0.71758 & 0.5 & 0.50449 & 0.49953 & 0.51106 & 0.50797 \\ 0.75929 & 0.50344 & 0.70392 & 0.71308 & 0.49551 & 0.5 & 0.49504 & 0.50656 & 0.50348 \\ 0.76426 & 0.5084 & 0.70888 & 0.71804 & 0.50047 & 0.50496 & 0.5 & 0.51152 & 0.50844 \\ 0.75273 & 0.49688 & 0.69736 & 0.70652 & 0.48894 & 0.49344 & 0.48848 & 0.5 & 0.49692 \\ 0.75582 & 0.49996 & 0.70044 & 0.70961 & 0.49203 & 0.49652 & 0.49156 & 0.50308 & 0.5 \end{bmatrix}. \tag{15}$$

4.1.3. Weights of State-Layer Elements of Technicality Evaluation Indicator. The weights of state-layer elements of technicality evaluation indicator are calculated based on (8). The results are shown in Table 4.

4.1.4. Scores of Evaluation Indicators. The score of the technicality evaluation indicator can be calculated based on (9), which is given by $X_1 = 89.85$.

Similarly, the scores of the energy efficiency evaluation indicator, security evaluation indicator, and operation

Table 4: Weights of state-layer elements.

Weight	Value	Weight	Value
ω_1^1	0.0651853	ω_1^2	0.1291497
ω_1^3	0.0790283	ω_1^4	0.0767381
ω_1^5	0.1311325	ω_1^6	0.1300089
ω_1^7	0.1312492	ω_1^8	0.1283686
ω_1^9	0.1291394		

Table 5: Scores of evaluation indicators.

Evaluation indicator	Score
Technicality evaluation indicator C_1	X_1=89.85
Energy efficiency evaluation indicator C_2	X_2=82.47
Security evaluation indicator C_3	X_3=83.26
Operation evaluation indicator C_4	X_4=87.34

evaluation indicator can be calculated. The specific results are shown in Table 5.

4.1.5. Reliability Evaluation. The indicator-layer fuzzy judgment matrix is given by

$$R_P = \begin{bmatrix} 0.5 & 0.4786 & 0.481 & 0.4929 \\ 0.5214 & 0.5 & 0.5024 & 0.5143 \\ 0.519 & 0.4976 & 0.5 & 0.512 \\ 0.5071 & 0.4857 & 0.488 & 0.5 \end{bmatrix}. \quad (16)$$

According to (5), R_P does not meet the consistency condition. Therefore, it is transformed into a fuzzy consistency judgment matrix R'_P based on (6), which is given by

$$R'_P = \begin{bmatrix} 0.5 & 0.4952 & 0.4958 & 0.4984 \\ 0.5048 & 0.5 & 0.5005 & 0.5032 \\ 0.5042 & 0.4995 & 0.5 & 0.5027 \\ 0.5016 & 0.4968 & 0.4973 & 0.5 \end{bmatrix}. \quad (17)$$

Calculate the weights of evaluation indicators based on (12). The results are shown in Table 6.

Then, the target-layer score is calculated as $S = 85.71$. According to Table 2, the reliability level of this distributed IoT device is excellent. However, the scores of the energy efficiency evaluation indicator and security evaluation indicator are relatively low, which indicate that the device may be abnormal. Through maintenance, it is found that the three-phase load of this device is unbalanced, and a small probability of drift deviation exists, which is consistent with the reliability evaluation results and proves the accuracy of the proposed method.

Table 6: The weights of evaluation indicators.

Weight	Value	Weight	Value
ω_1	0.248233	ω_2	0.251417
ω_3	0.251067	ω_4	0.249283

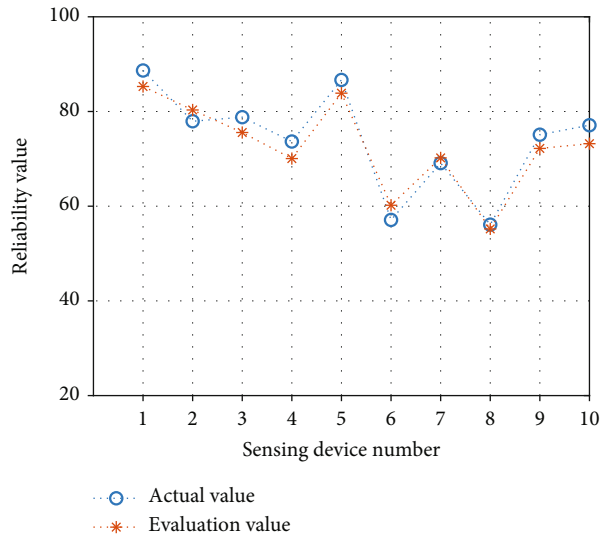


Figure 4: Reliability evaluation values of different IoT devices.

4.2. Simulation Comparison. In this section, the on-site operation data of ten distributed IoT devices in the distribution power grid of Shandong Province, China, are collected. We perform reliability evaluation on these ten devices and compare the evaluation results with the actual reliability value to verify the effectiveness. In addition, the proposed method is compared with two evaluation methods, i.e., principal component analysis (PCA) [30] and analytic hierarchy process (AHP) to verify the accuracy [31]. PCA performs correlation analysis on the indicators to obtain comprehensive indicators and complete reliability evaluation. AHP establishes a judgment matrix based on qualitative analysis to calculate the indicator weights and conduct reliability evaluation.

4.2.1. Reliability Evaluation of Distributed IoT Devices. Figure 4 shows the actual reliability values and the evaluation values, i.e., target-layer scores, of ten distributed IoT devices. It can be seen that the proposed method can achieve accurate reliability evaluation. The difference between the evaluation values and actual values is small, and the obtained reliability levels based on the evaluation values are accurate according to Table 2. The evaluation results indicate the superiority of the proposed method in the processing of indicators, which can effectively reduce the influence of the evaluator's subjective factors and objectively and truly reflect the reliability of the distributed IoT devices.

4.2.2. Accuracy Comparison. Figure 5 shows the reliability evaluation accuracy of different evaluation methods. Compared with PCA and AHP, FAHP can increase the evaluation accuracy by 9.86% and 6.96%, respectively. The reason

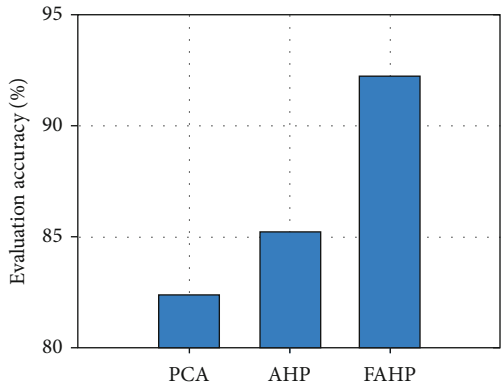


Figure 5: Accuracy comparison.

is that FAHP utilizes membership functions to obtain the scores of state-layer elements, which avoids the influence of subjective factors caused by artificial scoring. In addition, FAHP can flexibly adjust the boundary parameters of membership functions to adapt to different state-layer elements, different operation requirements, and different environments. AHP cannot get rid of the influence of subjective factors, resulting in a lower evaluation accuracy. The accuracy of PCA is the lowest because PCA can hardly distinguish the difference among multiple indicators.

5. Conclusions

In this paper, we addressed the reliability evaluation problem for distributed IoT devices in the distribution power grid and proposed the FAHP-based reliability evaluation method to overcome the influence of subjective factors and realize accurate reliability evaluation. Compared with PCA and AHP, simulation results indicate that the proposed FAHP-based method increases the evaluation accuracy by 9.86% and 6.96%, respectively. In the future work, we will further consider establishing more accurate membership functions based on the refinement characteristics of state-layer elements.

Data Availability

The [data type] data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the Science and Technology Project of State Grid Shandong Electric Power Company under grant number 2020A-010.

References

[1] Z. Zhou, M. Dong, K. Ota, G. Wang, and L. Yang, "Energy-efficient resource allocation for D2D communications underlaying cloud-RAN-based LTE-A networks," *IEEE Internet of Things Journal*, vol. 3, no. 3, pp. 428–438, 2016.

[2] J. Hu, X. Liu, M. Shahidehpour, and S. Xia, "Optimal operation of energy hubs with large-scale distributed energy resources for distribution network congestion management," *IEEE Transactions on Sustainable Energy*, vol. 12, no. 3, pp. 1755–1765, 2021.

[3] J. Liu, Z. Zhao, J. Ji, and M. Hu, "Research and application of wireless sensor network technology in power transmission and distribution system," *Intelligent and Converged Networks*, vol. 1, no. 2, pp. 199–220, 2020.

[4] A. Azizivahed, A. Arefi, S. Ghavidel et al., "Energy management strategy in dynamic distribution network reconfiguration considering renewable energy resources and storage," *IEEE Transactions on Sustainable Energy*, vol. 11, no. 2, pp. 662–673, 2020.

[5] K. Thirugnanam, M. Moursi, V. Khadkikar, H. Zeineldin, and M. Hosani, "Energy management of grid interconnected multi-microgrids based on P2P energy exchange: a data driven approach," *IEEE Transactions on Power Systems*, vol. 36, no. 2, pp. 1546–1562, 2021.

[6] Z. Zhou, H. Liao, B. Gu, K. Hup, S. Mumtaz, and J. Rodriguez, "Robust mobile crowd sensing: when deep learning meets edge computing," *IEEE Network*, vol. 32, no. 4, pp. 54–60, 2018.

[7] Z. Zhou, K. Ota, M. Dong, and C. Xu, "Energy-efficient matching for resource allocation in D2D enabled cellular networks," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 6, pp. 5256–5268, 2017.

[8] T. YuWen, "Research on performance evaluation method of navigation sensor in anti-lost clothing," *Wool Textile Journal*, vol. 48, pp. 103–108, 2020.

[9] F. Xiao, S. Wang, X. Xu, and G. Ge, "Automatic commissioning of AHU sensors using principle component analysis," *Building Science*, vol. 6, pp. 34–39, 2008.

[10] J. Yuan, *Research of Power Quality Dynamic Comprehensive Evaluation Method on Regional Power Grid*, South China University of Technology, Guangdong, 2012.

[11] Z. Zeng, J. Gan, W. Zeng, Q. Wang, and J. Yi, "A condition evaluation method of isolation circuit breaker based on triangular fuzzy analytic hierarchy process," in *In Proceedings of the 2021 5th International Conference on Power and Energy Engineering (ICPEE)*, vol. 2-4, pp. 22–25, Xiamen, China, December 2021.

[12] J. Li, "Study on the comprehensive performance evaluation of distribution network communication system," *Smart power.*, vol. 38, pp. 61–65, 2010.

[13] X. Chen, C. Wu, T. Chen et al., "Information freshness-aware task offloading in air-ground integrated edge computing systems," *IEEE Journal on Selected Areas in Communicatio.*, vol. 40, pp. 243–258, 2021.

[14] H. Liao, Z. Zhou, B. Ai, and M. Guizani, "Learning-based energy-efficient channel selection for edge computing-empowered cognitive machine-to-machine communications," in *In Proceedings of the 2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, vol. 25-28, pp. 1–6, Antwerp, Belgium, May 2020.

[15] T. Balachandran, A. Manoharan, V. Aravinthan, and C. Singh, "Component-level reliability evaluation model for cyber power devices," *IEEE Transactions on Smart Grid*, vol. 12, no. 1, pp. 692–703, 2021.

[16] Z. Zhou, Z. Wang, H. Liao, S. Mumtaz, L. Oliveira, and V. Frascolla, "Learning-based URLLC-aware task offloading for internet of health things," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 2, pp. 396–410, 2021.

[17] M. Berk, O. Schubert, H. Kroll, B. Buschardt, and D. Straub, "Reliability assessment of safety-critical sensor information: does one need a reference truth?," *IEEE Transactions on Reliability*, vol. 68, no. 4, pp. 1227–1241, 2019.

[18] L. Gao, C. Wu, T. Yoshinaga, X. Chen, and Y. Ji, "Multi-channel blockchain scheme for internet of vehicles," *IEEE Open Journal of the Computer Society*, vol. 2, pp. 192–203, 2021.

[19] H. Liao, Z. Zhou, X. Zhao et al., "Learning-based context-aware resource allocation for edge-computing-empowered industrial IoT," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4260–4277, 2020.

[20] N. Cha, C. Wu, T. Yoshinaga, Y. Ji, and A. Yau, "Virtual edge: exploring computation offloading in collaborative vehicular edge computing," *IEEE Access*, vol. 9, pp. 37739–37751, 2021.

[21] S. Xiang and J. Yang, "Reliability evaluation and reliability-based optimal design for wireless sensor networks," *IEEE Systems Journal*, vol. 14, no. 2, pp. 1752–1763, 2020.

[22] W. Dargie, "A quantitative measure of reliability for wireless sensor networks," *IEEE Sensors Letters*, vol. 3, no. 8, pp. 1–4, 2019.

[23] M. Berk, O. Schubert, H. Kroll, B. Buschardt, and D. Straub, "Exploiting redundancy for reliability analysis of sensor perception in automated driving vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, pp. 5073–5085, 2019.

[24] Z. Wang, "A representable uninorm-based intuitionistic fuzzy analytic hierarchy process," *IEEE Transactions on Fuzzy Systems*, vol. 28, no. 10, pp. 2555–2569, 2020.

[25] Z. Du, C. Wu, T. Yoshinaga et al., "A routing protocol for UAV-assisted vehicular delay tolerant networks," *IEEE Open Journal of the Computer Society*, vol. 2, pp. 85–98, 2021.

[26] Z. Ge and Y. Liu, "Analytic hierarchy process based fuzzy decision fusion system for model prioritization and process monitoring application," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 1, pp. 357–365, 2019.

[27] X. Mi, X. Wu, M. Tang et al., "Hesitant fuzzy linguistic analytic hierarchical process with prioritization, consistency checking, and inconsistency repairing," *IEEE Access*, vol. 7, pp. 44135–44149, 2019.

[28] Z. Deng and J. Wang, "Multi-sensor data fusion based on improved analytic hierarchy process," *IEEE Access*, vol. 8, pp. 9875–9895, 2020.

[29] A. Hawbani, E. Torbosh, X. Wang, P. Sincak, L. Zhao, and A. Al-DUbai, "Fuzzy-based distributed protocol for vehicle-to-vehicle communication," *IEEE Transactions on Fuzzy Systems*, vol. 29, no. 3, pp. 612–626, 2021.

[30] L. Tang and R. Li, "A novel QoS evaluation method in electric power communication networks," in *In Proceedings of the 2015 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, vol. 15-17, pp. 2039–2043, Zhangjiajie, China, August 2015.

[31] J. Luo, J. Yu, Q. Chen, J. Liao, Y. Huang, and J. Xu, "Analysis of bandwidth and site relationship for power communication network based on analytic hierarchy process," in *In Proceedings of the 2019 IEEE 3rd International Conference on Circuits, Systems and Devices (ICCSD)*, vol. 23-25, pp. 115–118, Chengdu, China, August 2019.

Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 6807257, 14 pages
https://doi.org/10.1155/2022/6807257

Research Article

Blockchain-Based Incentive Mechanism for Spectrum Sharing in IoV

Hongning Li,[1] Jingyi Li,[2] Hongyang Zhao,[3] Shunfan He,[4] and Tonghui Hu[2]

[1]*Xidian Guangzhou Institute of Technology, Guangzhou, Guangdong 511370, China*
[2]*Xidian University, Xi'an, Shaanxi 710071, China*
[3]*CEPREI, Guangzhou, Guangdong 511370, China*
[4]*South Central University for Nationalities, Wuhan, Hubei 430073, China*

Correspondence should be addressed to Hongyang Zhao; zhaohy@ceprei.com

Received 17 December 2021; Accepted 30 March 2022; Published 28 April 2022

Academic Editor: Celimuge Wu

In this paper, we design a blockchain-based incentive mechanism for the problem of low-level participation of primary users caused by location privacy leakage during spectrum data sharing in the Internet of Vehicles (IoV). First, we propose a K-anonymous location protection scheme for multiuser cooperation, which can protect the location privacy of primary users by generalizing their location information through the construction of anonymous areas. Then, we design an incentive mechanism, which performs reporting and adjudication strategy through the transaction stored in blockchain. Simulation results indicate that the proposed scheme can effectively prevent the privacy leakage of primary users' location and encourage them to actively participate in spectrum sharing in IoV.

1. Introduction

With the development of information technology, 6G will further realize the Internet of everything, establish multilevel and full-coverage seamless connection, and serve the key areas of multi-industry integration such as communication, transportation, and automobile. The vehicle networking system is being developed more quickly with the new generation of information and communication technology. 6G needs to support high-level security to meet the requirements of intelligent vehicle systems. The growing number of vehicles has significantly increased the consumption of spectrum resources. In fact, spectrum resources are divided into various frequency bands in a specific form given by government agencies, which are allocated to users with permission by issuing licenses. However, the existing spectrum management methods lead to some frequency bands being idle for most of the time, and the overall utilization rate of spectrum resources is very low. Take the USA as an example. A large number of investigations by the Federal Communication Commission show that the usage of spectrum resources is extremely unbalanced. Some authorized frequency bands are very crowded, while most of the others are idle [1]. Therefore, how to use spectrum effectively has become an urgent problem to be solved.

The 6G white paper points out that the full and efficient utilization of spectrum resources in different frequency bands can be realized through recultivation, aggregation, and sharing. It meets the spectrum needs of the 6G era. Most of the existing methods for obtaining free spectrum are based on the perception of secondary users, but the accuracy may be affected by malicious users. How to encourage primary users to actively participate in spectrum sharing and improve the accuracy of available spectrum information is an urgent problem to be solved.

Incentive mechanism can guarantee the needs of participating users through special forms of interest division, which is an effective way to stimulate users to participate in spectrum sharing. In spectrum sharing, to get benefits, primary users with permission can share the bands when they have no communication requirements. Other users that can use idle band shared by primary users are called

secondary users. To get spectrum information, Feng et al. propose a monetary incentive mechanism based on reverse auction to encourage secondary users to participate in spectrum sensing [2]. Li et al. adopt a pricing mechanism based on maximizing expected utility to encourage users to participate in perception. Ying et al. [3] use cooperative spectrum sensing schemes based on the evolutionary game and Stenberg game model to improve detection performance [4]. However, most of the current research considers using spectrum sensing technology to obtain idle spectrum information and use idle bands for opportunities, while little research is done on the active sharing of primary users. Elnahas et al. [5] propose an auction mechanism with time-varying valuation information to maximize auction revenue to encourage primary users to join the market. Literature [6] proposes to increase auction revenue in a dynamic secondary market to improve spectrum utilization. In fact, the participation of primary users can effectively improve spectrum utilization efficiency.

The effective implementation of spectrum sharing in IoV depends on the active participation of all users in the network. How to encourage the primary users to actively participate in spectrum sharing is one of the important issues that need to be studied in IoV.

In addition, the primary users need to submit a certain range of location information to a third party (such as a spectrum distribution center) in spectrum sharing in IoV. The more precise the location provided, the more conducive to the allocation and use of free spectrum. Untrustworthy third parties can infer their personal sensitive information from the primary users' spectrum status and sharing license information, causing hidden dangers to user privacy and security [7], thereby reducing the primary users' enthusiasm for participating in spectrum sharing. Due to the lack of protection of the location information and effective incentive mechanism for primary users in IoV, primary users have no incentive to participate in spectrum sharing.

At present, there are many blockchain-based technologies and methods applied in privacy protection. In 2016, Yuan et al. used blockchain technology to build a secure and trusted distributed autonomous transportation system for the first time [8]. Benjamin et al. proposed a distributed storage-based vehicle networking system based on Ethereum to achieve secure communication between vehicles [9]. In the literature [10–12], to provide reliable reference and credible data for law enforcement agencies involved in information exchange or traffic accident evidence collection, a distributed data storage is constructed using blockchain technology. According to the literature [13–16], blockchain distributed storage can enhance the reliability of data, and users in the blockchain system use pseudonyms, which cuts off the connection between user names and their real identities and prevents malicious nodes from obtaining users' real identity. In [17], blockchain and in-vehicle IoT features and related research questions are discussed. Besides, in literature [18], a multichannel blockchain solution is applied to the blockchain. It can be seen that the current research uses the blockchain to solve the problem of privacy leakage. Therefore, the application of blockchain in the field of privacy protection can be used as an effective means to solve the problem of privacy leakage of main users in the spectrum sharing in IoV.

Therefore, the paper proposes an incentive mechanism based on location privacy protection (IMLPP), which uses the blockchain to protect primary users' location information and encourages them to actively participate in spectrum sharing. The incentive mechanism is designed to improve the utilization rate of the spectrum and further solve the problem of the shortage of spectrum resources. In the proposed mechanism, the distributed K-anonymous scheme based on blockchain is used to generalize the location information of primary users, which ensures that even if the opponents can obtain the spectrum allocation information, the real location of primary users cannot be inferred. The main contributions of the paper are as follows:

(1) We propose a privacy-preserving scheme based on blockchain to generalize primary users' location during spectrum sharing. In this scheme, users with a certain requirement can be selected to cooperate with primary users to construct anonymous areas. The information of construction of anonymous area is stored in blockchain as transaction, and it can be used as evidence of users' behavior

(2) We propose an incentive mechanism to encourage primary users to participate in spectrum sharing. The honesty degree is proposed to measure the integrity of users. Each user in the network has an initial honesty degree, which is updated according to the users' behavior. With deposit payment, honesty degree evaluation algorithm, and users' behavior constraint in the blockchain, the incentive mechanism can effectively encourage primary users to participate in spectrum sharing and constraint users' behavior

2. Spectrum Sharing Incentive Mechanism Based on Location Privacy Protection

2.1. System Model. This paper considers the spectrum sharing model of IoV as shown in Figure 1, which includes a fusion center and multiple vehicle users. Communication is enabled among users and between users and the fusion center. The fusion center issues spectrum sensing tasks, calculates spectrum data, and allocates idle spectrum to secondary users. Primary users share their idle spectrum and protect location information by issuing request for location information protection and constructing anonymous areas with the assistance of other users in the network.

In this paper, the primary users who participate in spectrum sharing and need to protect their location information are called the requesting user, and users that provide encrypted location information to help the requesting user construct an anonymous area are called cooperative users. Requesting users use the location information provided by cooperative users to construct anonymous areas to meet the needs of privacy protection.

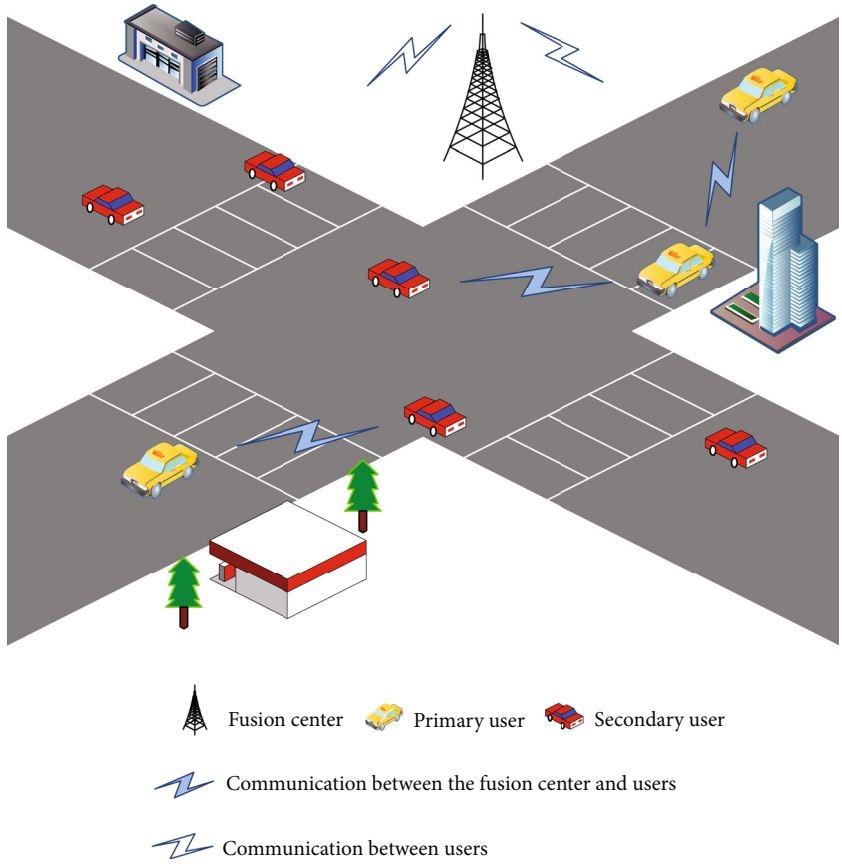


Figure 1: System model.

To construct a reliable anonymous area, cooperative users' behavior should be constrained. In this paper, we define two types of illegal behaviors, one is requesting users who disclose the location information of cooperative users, and the other one is cooperative users who provide false locations. The process of assisting a requesting user to construct anonymous area is regarded as a transaction. The requesting user ID, the cooperative user ID, and the location information of the cooperative user are taken as transaction bill information and then encrypted and recorded in the blockchain (the blockchain is a private chain in IoV). This process will generate a certain amount of virtual currency (called mining) in the blockchain system.

2.2. Incentive Mechanism Based on Location Privacy Protection. In this section, an incentive mechanism based on location privacy protection (IMLPP) is proposed, which is shown in Figure 2. The mechanism uses K-anonymous scheme based on blockchain with cooperative users to generalize primary users' location information. In this mechanism, an honesty degree evaluation mechanism is designed to provide a basis for selecting between requesting users and cooperative users. By paying the deposit, reporting and adjudicating with the transaction bill as the evidence, users' location information can be protected. On this basis, the honesty degree and virtual currency in the blockchain are taken as incentives for the primary users to participate in spectrum sharing.

IMLPP scheme is divided into four sections, honesty degree mechanism, anonymous area construction, report and adjudication strategy, and incentive mechanism.

2.2.1. Honesty Degree Mechanism. In the honesty degree mechanism, honesty degree is used to measure the credibility of users, as the basis for mutual choice in the transaction, to meet the user's personalized security requirements for location privacy, and as the reference basis for the fusion center to allocate spectrum. Specifically, requesting users want the cooperative users with high honesty degree to participate in anonymous area construction to ensure the accuracy of the location provided by cooperative users. Cooperative users also tend to cooperate with requesting users with high honesty degree to ensure that location information is not disclosed. Secondary users with high honesty degree will be allocated spectrum with high probability.

The honesty degree evaluation algorithm is the basis of honesty degree update. Assuming that m_0 and m_1 are constant coefficients, m_0 and m_1 can be any positive number, and the value of m_0 and m_1 has no effect on the results of this experiment. We consider $m_0 = 20$, $m_1 = 20$ in this paper. B is a Boolean variable, and if the user has illegal behavior, $B = 0$, and on the contrary, $B = 1$. The initial honesty degree H_0 of all users is 60, and the upper limit of the honesty degree is 200. We assume that the current honesty degree of user U_i is H_i, and the honesty degree evaluation algorithm is shown in Algorithm 1.

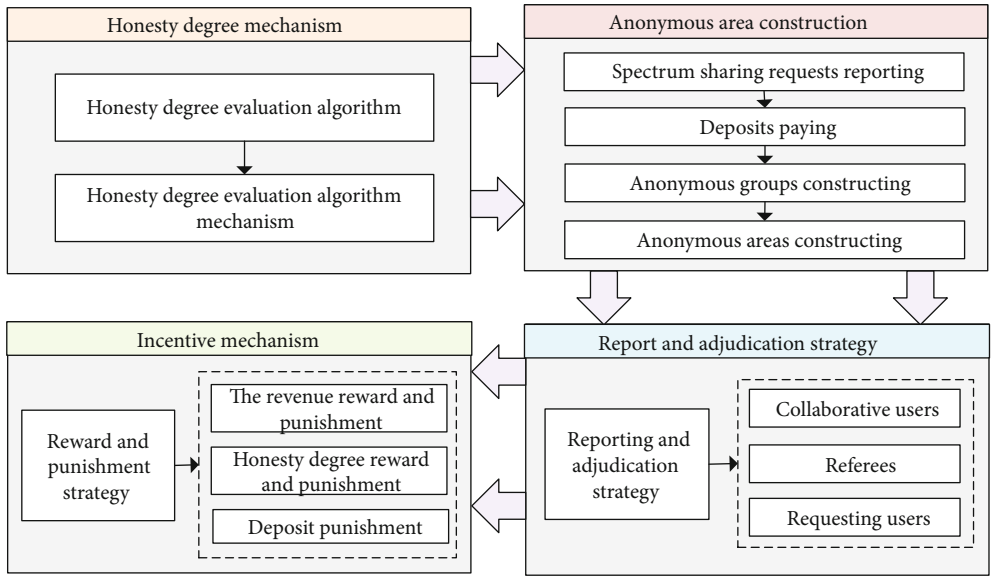


Figure 2: IMLPP.

According to the honesty degree evaluation algorithm, if the user's honesty degree is higher, the more honesty degree will be deducted when the user commits illegal acts, and the more slowly of the user's honesty degree increases.

2.2.2. Anonymous Area Construction. This section gives the detail of anonymous area construction, which uses distributed K-anonymous scheme to protect primary users' location information. It contains spectrum sharing requests, deposits paying, anonymous groups constructing, and anonymous areas constructing. Among them, the anonymous group is a set of users who are willing to participate in the construction of the anonymous area and meet the requirements.

To illustrate, we take a primary user PU_i as an example. PU_i sends a request to the smart contract:

$$\text{request} = \left\{ID_{PU_i}, H_{PU_i}, H_U, (K-1)\right\}, \tag{1}$$

where ID_{PU_i} is the only identifier of PU_i in the blockchain system. H_{PU_i} is PU_i's honesty degree. H_U is the lower limit of the honesty degree of cooperative users. $(K-1)$ is the number of cooperative users to meet different requirements of different requesting users for location privacy.

After receiving the request, the smart contract determines whether to assist in constructing the anonymous area according to PU_i's honesty degree H_{PU_i}:

(1) When $\mathrm{H_{PU_i}} < 40$, the request is rejected.

Then, the smart contract calculates and returns the deposit that PU_i has to pay:

$$D_{PU_i} = \frac{m_2}{H_{PU_i}}, \tag{2}$$

where m_2 is the income to be generated from this mining. It can be seen from Formula (2) that the higher the honesty degree, the lower the deposit PU_i need to pay.

After paying the deposit, PU_i's location protection request is broadcasted in the network, and other users in the network choose whether to participate in the anonymous area construction according to PU_i's honesty degree H_{PU_i}. In order to guarantee the construction of anonymous area, this paper introduces willingness list $\text{wish} = \{U_1 : H_{U_1}, U_2 : H_{U_2}, \cdots, U_i : H_{U_i}\}$, which includes users' honesty degree and their serial numbers.

When the user U_i is willing to participate in the anonymous area construction, it sends a request to the smart contract. Then, the smart contract will put U_i into the willingness list. If U_i's honesty degree $H_{U_i} \geq H_U$, the smart contract returns the deposit D_{U_i}, and U_i will join the anonymous group after paying the deposit D_{U_i}, which meets the following requirements:

$$D_{U_i} = \frac{m_2}{K * H_{U_i}}. \tag{3}$$

If K-1 cooperative users join the anonymous group, the anonymous group is successfully constructed. If the anonymous group construction fails because K or H_U value is too high, the smart contract will send the wish list to PU_i. PU_i adjusts H_U and K according to the wish list and sends to the smart contract to reconstruct the anonymous group.

After the anonymous group is successfully constructed, all cooperative users $U_i (i = 1, 2, \cdots, K-1)$ in the group send PU_0 location information bills $Bill_{LOC_{U_i}}$, which meets the following requirements:

$$Bill_{LOC_{P_i}} = \left\{ID_{U_i}, P_{PU_i}\left(E_{U_i}\left(Loc_{U_i}\right)\right)\right\}, \tag{4}$$

where ID_{U_i} is the cooperative user U_i's identity, Loc_{U_i} is U_i's

Input: Current honesty degree H_i;
Output: Updated honesty degree $H_{i'}$.
①For each H_i do:
② if $B=0$:
//The user has illegal behavior
③ $H_{i'}=H_i - H_i/m_1$
④ else if $B=1$ and $H_i < 200$:
//The user is honest and the current honesty degree level is not up to the upper limit
⑤ $H_{i'} = H_i + m_0/H_i$
⑥ if $H_{i'} > 200$:
//Updated fidelity exceeds the upper limit
⑦ $H_{i'} = 200$
⑧ else:
//The user is honest and the honesty degree reaches the upper limit
⑨ $H_{i'} = 200$

Algorithm 1: Honesty degree evaluation algorithm.

location information, and $E_{U_i}(Loc_{U_i})$ is the encrypted ciphertext of location information using the U_i's SU1 private key and PU_i's public key.

PU_i uses his private key and U_i's public key to decrypt the ciphertext $E_{U_i}(Loc_{U_i})$ and obtain U_i's location information Loc_{U_i}, which is used to construct the location anonymous area.

We assume PU_i's identity is ID_{PU_i}, and the location information is Loc_{PU_i}. Before using the location privacy protection scheme, PU_i submits location information that is shown in Table 1, and the fusion center can directly obtain PU_i's location information.

With the location privacy protection scheme, a multilocation information anonymous area is submitted to the fusion center by PU_0. As shown in Table 2, the probability that the fusion center can correctly analyze the location information of the primary user is only $1/K$.

After the anonymous area is constructed, PU_i submits the anonymous area together with the spectrum sharing license to the fusion center. Then, $P_{PU_i}(E_{U_i}(Loc_{U_i}))$, ID_{PU_i}, and ID_{U_i} are written into the transaction bill by PU_i for broadcasting throughout the network, which is shown in Table 3. The users with honesty degree greater than 60 in IoV jointly participate in the calculation competition to write the transaction bill on the block and add the block to the blockchain.

Since the size of the anonymous area is much larger than the moving distance of vehicles during the time when the anonymous area is constructed, the error caused by the vehicle movement is ignored in this paper.

2.2.3. Report and Adjudication Strategy. For the possible users' illegal behaviors in this scheme, this paper proposes a strategy for judging and punishing illegal behaviors, which is called report and adjudication strategy. In addition, we give the concept of referees to refer to those users who participate in adjudicating illegal behavior. Firstly, the reporting and adjudication strategy of requesting users and cooperative users are defined as follows.

(1) Definition I (Reporting and Adjudication Strategy).

(i) In the reporting and adjudication strategy a_1, we define the reporting and adjudication strategy of cooperative users. When U_i discovers that his location information is leaked by PU_i, U_i sends the smart contract a request to report PU_i, and provides evidence of PU_i 's illegal behavior. Then the request is broadcasted in the network. The first 50 users (referees) in the network to respond carry out verification and adjudication. Referees retrieve transaction bills in the blockchain, verify the report information according to the transaction bills, and determine whether support the reporting based on the evidence

(ii) In the adjudication strategy a_2, we define the reporting and adjudication strategy of the requesting users. When PU_i finds that the security of the constructed anonymous area is reduced due to the provision of false location information by U_i, PU_i uses his private key to decrypt $P_{PU_i}(E_{U_i}(Loc_{U_i}))$ in the transaction bill, and obtain $E_{U_i}(Loc_{U_i})$. Then, $E_{U_i}(Loc_{U_i})$ and related evidence (such as the location is no man's land, etc.) are sent to the smart contract for reporting, which is broadcasted in the network. After verifying the report information, the referees use U_i 's public key to decrypt the ciphertext $E_{U_i}(Loc_{U_i})$ to get the location information Loc_{U_i}. Finally, referees determine whether support the report based on Loc_{U_i} and the evidence

According to reporting and adjudication strategy, after the user initiates a report, if there are more than 25 referees who support the report, it will be determined that the

Table 1: Position table before generalization.

User	Location information
ID_{PU_i}	Loc_{PU_i}

Table 2: Anonymous area.

User	Location information
ID_{PU_i}	$Loc_{U_1}, Loc_{U_2}, Loc_{U_3}, ..., Loc_{PU_i}, ..., Loc_{U_{k-1}}$

Table 3: Transaction bill.

User	Location information
ID_{PU_0}	–
ID_{U_1}	$P_{PU_i}(E_{U_1}(Loc_{U_1}))$
ID_{U_2}	$P_{PU_i}(E_{U_2}(Loc_{U_2}))$
...	...
$ID_{U_{K-1}}$	$P_{PU_i}(E_{U_{K-1}}(Loc_{U_{K-1}}))$

reported user has illegal behavior; otherwise, the report will be invalid.

Considering that the referee may make an adjudication without verification, which will affect the report result, this paper puts forward the adjudication strategies for the referee's illegal behaviors.

For the referee J_i, if the adjudication is wrong for T consecutive times, J_i would be adjudicated as an illegal user, and the T value meets

$$T = \left[H_{J_i}^{m_4}\right] + m_5, \tag{5}$$

where H_{J_i} is J_i's honesty degree. The value range of m_4 is between 0 and 1, and m_5 can be other positive numbers. In subsequent simulation experiments, we consider m_4= 0.5, m_5=2.

From Formula (5), the value of T is related to the honesty degree of the user. The higher the honesty degree of the user, the better the inclusiveness to the user, and the more times the error can be decided.

2.2.4. Incentive Mechanism. To encourage the primary user to participate in spectrum sharing, all users in the network to participate in anonymous area construction and adjudication and restrict users' behavior; this paper proposes reward and punishment mechanisms in different scenarios.

(1) Definition II (Responsivity). The ratio of the number of users responding to a primary user's request for anonymous area construction information to the total number of users in the network is called the response rate.

For the primary users, the higher the honesty degree, the higher the response rate. Only by improving the honesty degree can the higher response rate be obtained. For secondary users, only by improving honesty degree can they have higher priority in spectrum allocation. Therefore, in addition to the virtual currency in the blockchain, honesty degree is also used as an incentive for users. In this scheme, we propose a reward and punishment mechanism to reward users and punish users who have illegal behavior, which consists of reward and punishment strategies in three aspects, namely income, deposit, and honesty degree.

(2) Definition III (Reward and Punishment Strategy).

(i) In the reward and punishment strategy b_1, the revenue reward and punishment are defined. Users who participate in anonymous area construction or spectrum sharing will get virtual currency rewards, and users who have illegal behaviors have lower income in the penalty round (we set the penalty round to 10 rounds).

After the transaction bill is linked up, the miners look for whether there is a penalty transaction bill for PU_i's and U_i's illegal behaviors in the blockchain. Assume that m_2 is the virtual currency generated by the miner through mining, and the miner obtains virtual currency is $m_2/3$:

(1) If no penalty transaction bill for PU_i's illegal behavior is found in the blockchain, the miner will assign PU_i virtual currency C_{PU_i}, which meets the following requirements:

$$C_{PU_i=}\frac{m_2}{3}. \tag{6}$$

(2) If no penalty transaction bill for U_i's illegal behavior is found in the blockchain, the miner will assign U_i virtual currency C_{U_i}, which meets the following requirements:

$$C_{U_i=}\frac{m_2}{3K}. \tag{7}$$

(3) When PU_i's illegal behavior is found and it exists in the lth block $block_l$, assume that N is the current number of blocks, C_i is the income when the user has no illegal behavior, and C_i' is the actual income of the user this time:

(a) If $N - L \le 10$, then the miner assigns virtual currency to the user:

Table 4: Simulation parameter table.

Parameters	Meaning	Default
N	Number of users	10000
A	Proportion of primary users	30%
B	Proportion of secondary users	50
C	Percentage of attackers	20%
Cycle	Number of simulations	0 ~ 200
Block	Current block length	100
M	Number of transactions stored per block	100
K	Number of users participating in anonymous area construction	2 ~ 37

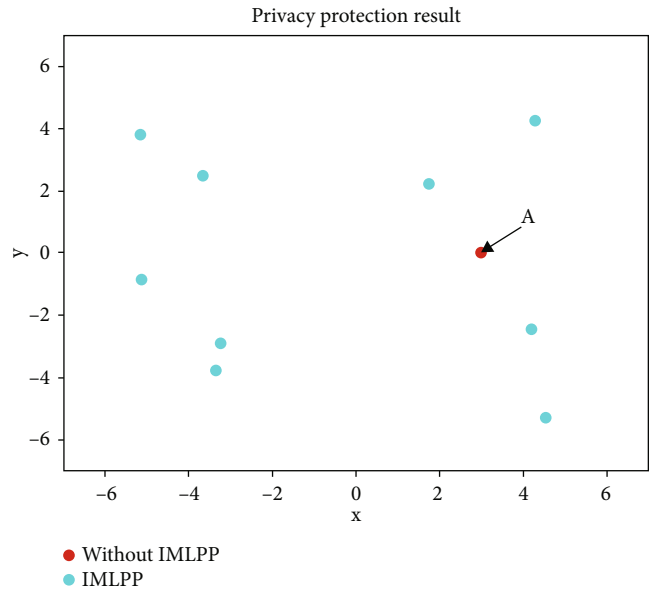


Figure 3: Anonymous region of $K = 10$.

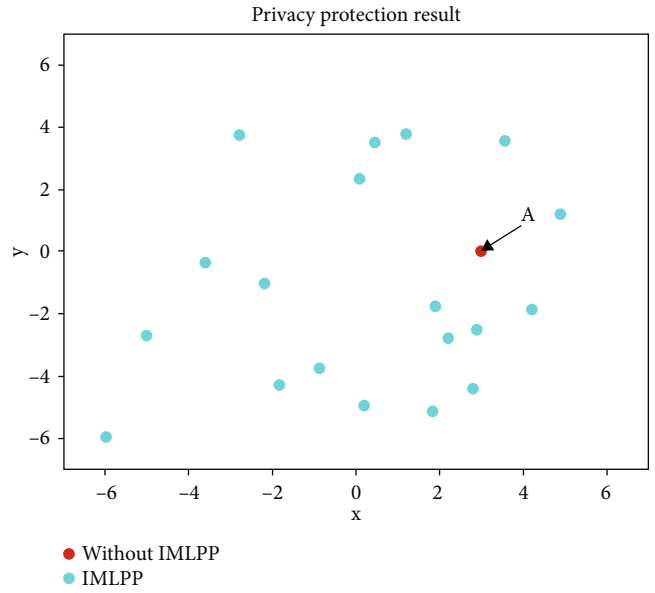


Figure 4: Anonymous region of $K = 20$.

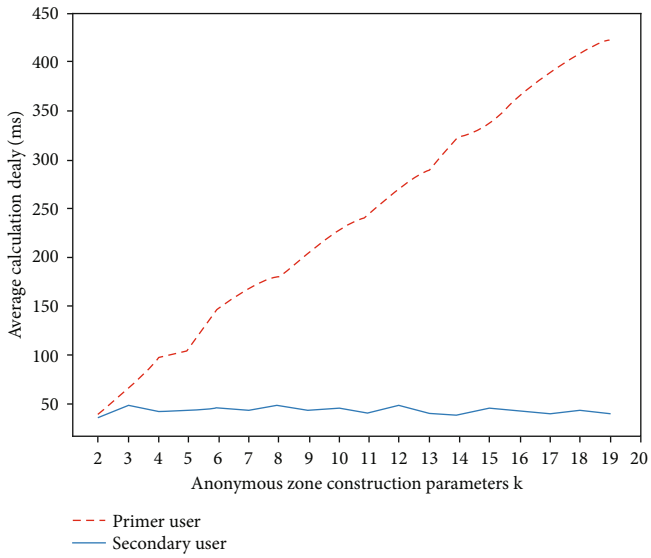


Figure 5: Average calculation delay.

$$C_i' = \frac{C_i}{2}. \tag{8}$$

(b) If $N - L < 10$, then the miner assigns virtual currency to the user:

$$C_i' = C_i. \tag{9}$$

(ii) In the reward and punishment strategy b_2, honesty degree reward and punishment and deposit punishment are defined. The honesty degree is updated according to the honesty degree evaluation algorithm. If users participate in anonymous area construction, share spectrum, or have illegal behavior, their honesty degree will be updated. Besides, the deposit paid by illegal users will be used as compensation for privacy victims

After the transaction bill is linked, PU_i's and U_i's honesty degree H_i will be updated as follows according to the honesty degree evaluation algorithm:

$$H_i = H_i + \frac{20}{H_i}. \tag{10}$$

If there is a user who has illegal behavior during the construction of the anonymous area, the penalty transaction bill will be broadcast and the user will be punished:

(1) If U_i is adjudicated to have illegal behavior, the deposit paid by U_i will be used as PU_i's compensation, and U_i's honesty degree H_{U_i}will be updated as follows according to the honesty degree evaluation algorithm:

$$H_{U_i} = H_{U_i} - \frac{H_{U_i}}{20}. \tag{11}$$

(2) If PU_i is adjudicated to have illegal behavior, the deposit paid by PU_iwill be used as compensation, and PU_i's honesty degree H_{PU_i}will be updated according to the honesty degree evaluation algorithm:

$$H_{PU_i} = H_{PU_i} - \frac{H_{PU_i}}{20}. \tag{12}$$

The following is an introduction to the reward and punishment mechanism of referees.

Assume that a referee J_i's honesty degree is H_i:

(1) If after J_i participating in the ruling, J_i is not determined to be user who has illegal behavior, and J_i's honesty degree will increase to

$$H_{J_i} = \left(H_{J_i} + \frac{20}{H_{J_i}}\right). \tag{13}$$

(2) If J_i's adjudicated to be an illegal user after participating in the ruling, J_i's honesty degree will be reduced to

$$H_{J_i} = \left(H_{J_i} - \frac{H_{J_i}}{20}\right). \tag{14}$$

The reward and punishment mechanisms reward the primary users who participate in spectrum sharing, the cooperative users who participate in the construction of anonymous areas, and the referees who participate in the adjudication, and punish the illegal users, which not only play an incentive role, but also can effectively restrain the user behavior.

3. Simulation Experiment and Analysis

3.1. Simulation Environment. In this section, we conduct a simulation analysis on the proposed IMLPP scheme to verify its impact on location privacy protection and spectrum sharing incentives in spectrum sharing in IoV. The parameter settings of simulation environment are shown in Table 4.

3.2. Simulation Analysis and Results of Location Privacy Protection. In this paper, a distributed K-anonymous

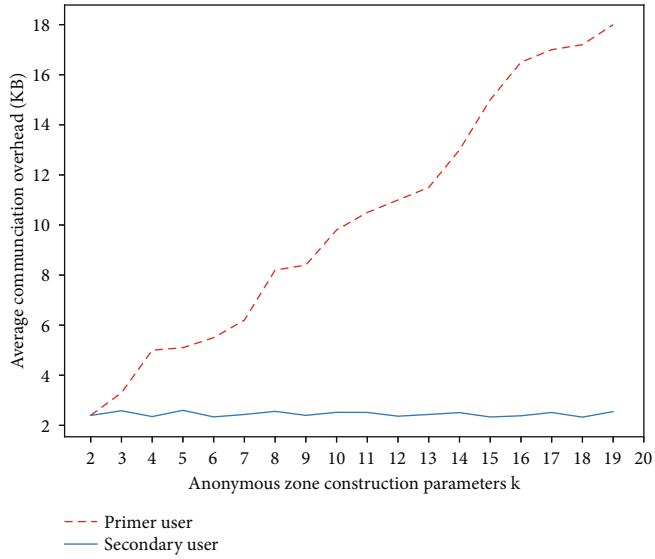


Figure 6: Average communication overhead.

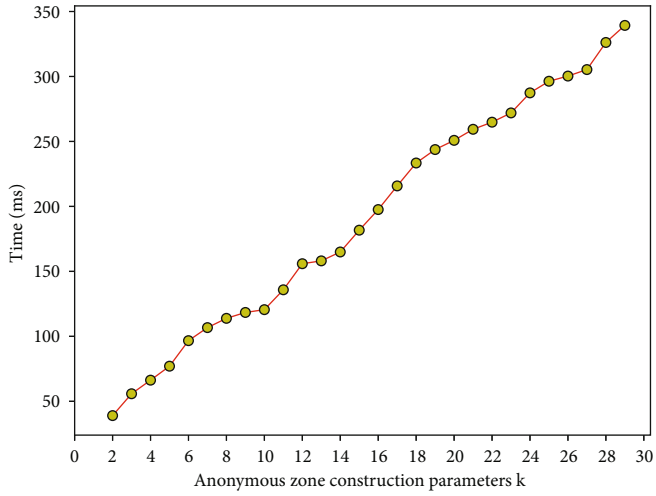


Figure 7: Time consumption for different K values.

algorithm is designed by using blockchain technology, which protects the location privacy security of the primary user in the spectrum sharing of the vehicle network and solves the privacy security threat of the primary user caused by spectrum sharing.

3.2.1. Privacy Protection of Constructing Anonymous Areas. This part of the experiment analyzes the privacy protection effect of the privacy protection scheme on the primary user. The vehicle user running in the vehicle network is regarded as a point moving in a two-dimensional plane coordinate system, and the coordinates of the point represent the position of the user. As shown in Figure 3, before using IMLPP scheme, the user's position is red point A, which can be directly obtained by attackers. After using this scheme, when $K = 10$, the user's position A (3, 0) is generalized to an anonymous area composed of 10 points, and the probability that the attacker can correctly analyze the position of point A is only 1/10. When $K = 20$, as shown in Figure 4, the probability that the attacker gets the location of A is 1/20. The larger the K value, the safer the user's location privacy. We use Java to perform simulation experiments and use Python to plot and analyze the experimental result data.

3.2.2. Influence of Parameter K on Average Computing Delay and Communication Overhead. In this part, the calculation delay and communication overhead of users in the process of anonymous area construction are analyzed experimentally.

We select different K values for simulation experiments; the value of K ranges from 2 to 19 and obtains the user's computing delay and communication overhead, as shown in Figures 5 and 6. It can be seen from the figure that the K value will affect the computational delay and communication overhead required by the requesting users, and the cooperative users will not be affected by it.

This is because when the requesting user receives the location bill of the cooperative user, the requesting user

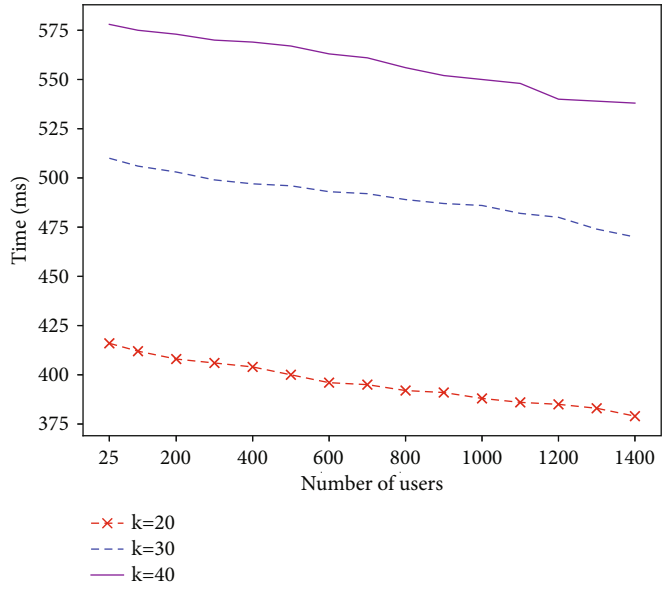


Figure 8: Time consumption when the number of users in the network varies.

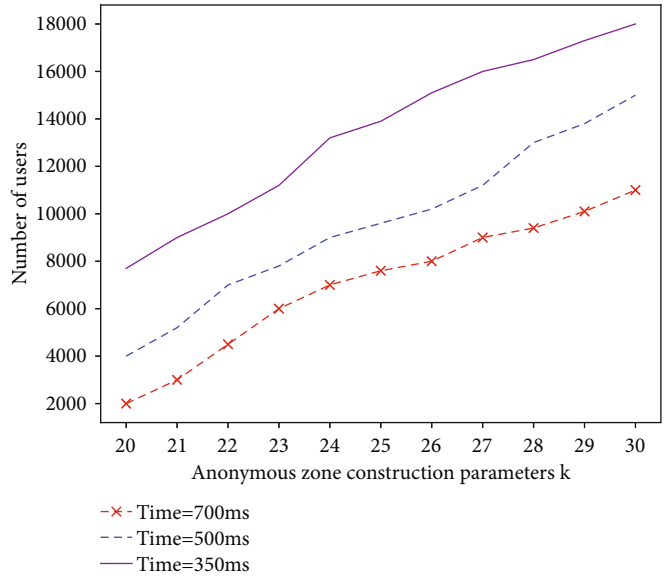


Figure 9: Relationship between users and K value in the network at a limited time.

needs to decrypt the location information using the public key of the cooperative user, while the cooperative user only needs to send the location bill to the requesting user. Therefore, with the increase of K value, the calculation delay required by the requesting user increases, and the cooperative user will not be affected by it, as shown in Figure 5.

In addition, during anonymous area construction, as the number of cooperative users participating in the anonymous area construction increases, the number of location information bills that the requesting user needs to receive increases, and the amount of information that needs to be processed increases, while the cooperative user is not affected. Therefore, as shown in Figure 6, the communication overhead of the requesting user increases with the value of K, while the communication overhead of the cooperative user is not affected by the change of the value of K.

In addition, we control the number of users in the network to be 10,000 and select different K values for simulation experiments. The K value ranges from 2 to 30, and the generation time of the anonymous area is obtained, as shown in Figure 7. The figure shows, when the number of users in the network is fixed, the time for constructing the anonymous area will increase with the increase of the K value, but the larger the K value, the better the location privacy of the primary user can be protected. In addition, when the value of K is fixed, as shown in Figure 8, the number of users in the network is inversely proportional to the time for constructing the anonymous area, and the more users in the

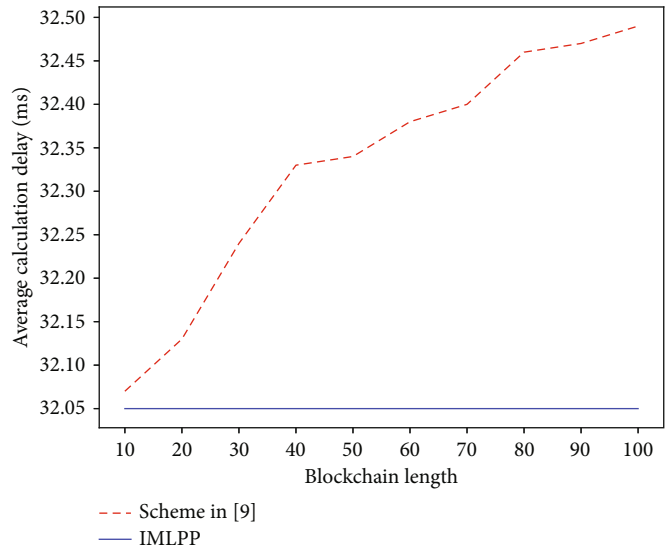


Figure 10: The effect of blockchain length on unauthorized users.

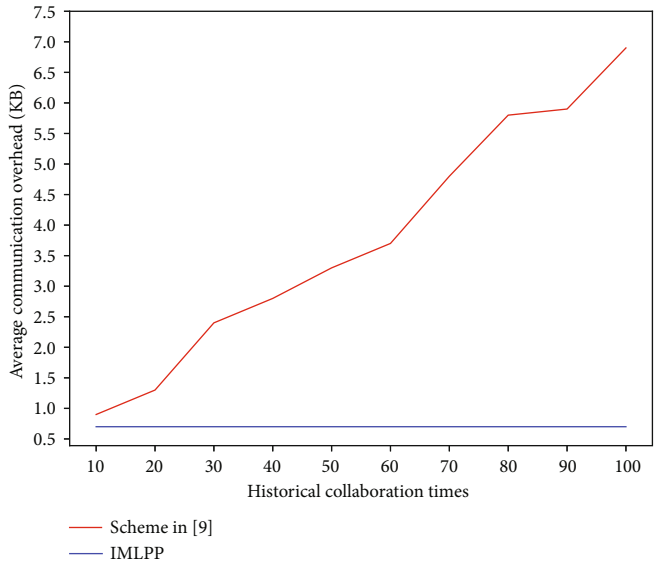


Figure 11: The impact of historical collaboration times on the communication overhead required by requesting users.

network, the less time it takes to construct the anonymous area. However, within a limited time, as shown in Figure 9, the larger the K value required by the primary user, the higher the number of users in the network.

3.2.3. The Influence of Blockchain Length. In this scheme, after receiving an anonymous area construction request sent by an authorized user, it is only necessary to choose whether to participate in its spectrum sharing according to its integrity, and the length of the blockchain will not affect unauthorized users, while in the scheme [19], in order to verify whether there is location privacy leakage or fraudulent behavior in the history of the requesting user, the collaborating user needs to download and query the transaction bills stored in the entire blockchain. Therefore, as shown in Figure 10 in the scheme [19], with the increase in the length of the distributed anonymous area cooperative construction blockchain, the computing delay required by users in the anonymous area construction process is also increasing, and the length of the blockchain will not affect this scheme. Therefore, this scheme can reduce the computational experiment well.

3.2.4. The Impact of Historical Collaboration Times. In scheme [19], the user's ID will be used as an index to retrieve all historical transaction bills containing the ID in the blockchain system, so that each user in the network can trace the historical behavior of requesting users and cooperative users. As shown in Figure 11, as the number of times that the requesting user participates in the construction of the anonymous area as a collaborator increases, the number of transaction bill numbers that the requesting user needs to provide also increases, resulting in the requesting user needing to construct the anonymous area. The communication

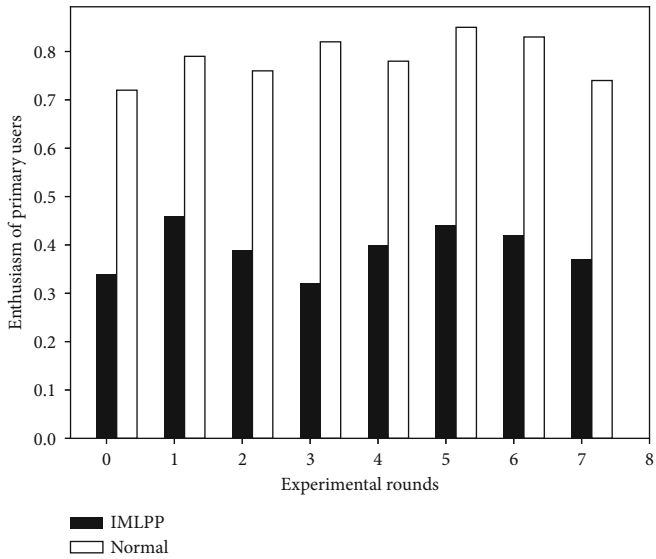


Figure 12: Enthusiasm of primary users to participate in spectrum sharing.

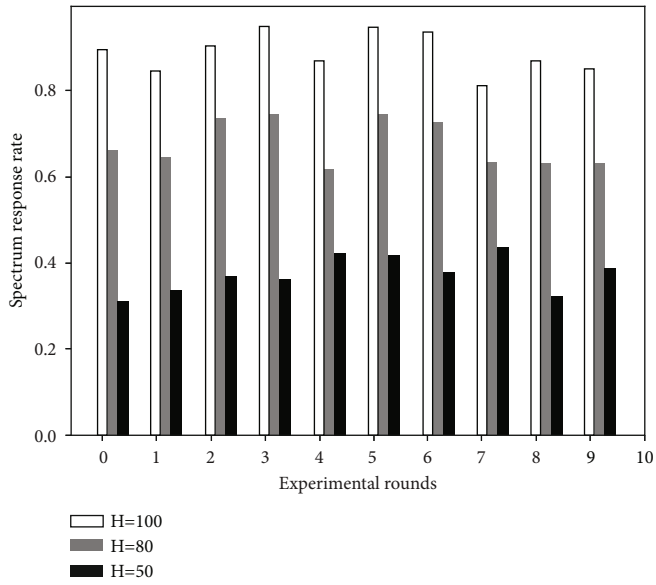


Figure 13: Response rates of primary users with different honesty degree.

overhead also increases, and the integrity evaluation algorithm used in this paper makes the user do not need to provide the transaction bill number, so the number of historical cooperation will not affect the communication overhead required in the construction of the user's anonymous area. This scheme can reduce the communication overhead very well.

3.3. Simulation Results and Analysis of Spectrum Sharing Excitation. This part of the experiment analyzes the incentive effect of incentive mechanism on primary users. In an environment without incentive mechanism, primary users are divided into three types: (1) always actively share their idle spectrum, (2) sometimes share their free spectrum, and (3) do not participate in spectrum sharing. Set the initial proportion of class I users with idle spectrum to 20%, class II users to 60%, and class III users to 20%. Assuming that all primary users in the current network have idle spectrum, the proportion of primary users willing to participate in spectrum sharing to the total number of primary users in the network is taken as the positive rate of spectrum sharing. As shown in Figure 12, in the absence of incentives, only the first and second types of primary users will participate in spectrum sharing, and due to the low enthusiasm of the second type of primary users, the positive rate of spectrum sharing is between 0.3 and 0.5. Under the environment of incentive mechanism, the second and third types of primary users will also actively participate in spectrum sharing to obtain virtual currency

rewards and improve honesty degree, so the positive rate of spectrum sharing is between 0.7 and 0.9.

As shown in Figure 13, in the histogram, from left to right are the response rates of the primary users with honesty degree of 100, 80, and 50 in the location privacy protection scheme. The higher the honesty degree of the primary users, the higher the response rate. This is because the higher the honesty degree, the more credible the users are, and the more users are willing to participate in their location privacy protection.

4. Concluding Remarks

This paper proposes an incentive mechanism called IMLPP, which uses a blockchain-based K-anonymity scheme to construct a K-anonymity area that meets the needs of the primary user to protect their location information in spectrum sharing. On this basis, honesty degree and virtual currency are used to motivate users. The proposed scheme can effectively generalize primary users' location information, meet their personalized privacy protection needs, and encourage them to actively participate in spectrum sharing. In addition, both requesting users and cooperative users need to pay deposit, which restricts the user's behavior.

Data Availability

The data of secure computation protocols and algorithms used to support the findings of this study are available from the corresponding author upon request.

Additional Points

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is partly supported by the National Key Research and Development Program of China under Grant 2021YFB2700600 and 2019YFC0118800, the National Natural Science Foundation of China under Grant 62132013 and 61903384, the Key Research and Development Programs of Shaanxi under Grant 2021ZDLGY06-03, and High-Level Innovation Research Institute Project under Grant 2021B0909050008.

References

[1] M. Rajendran and M. Duraisamy, "Distributed coalition formation game for enhancing cooperative spectrum sensing in cognitive radio ad hoc networks," *IET Networks*, vol. 9, no. 1, pp. 12–22, 2020.

[2] F. Jingyu, Y. Jinwen, Z. Ruitong, and Z. Wenbo, "Internet of things spectrum sharing incentive mechanism against location privacy leakage," *Computer Research and Development*, vol. 57, no. 10, pp. 2209–2220, 2020.

[3] L. Xiaohui, Z. Qi, and W. Xianbin, "Privacy-aware crowdsourced spectrum sensing and multi-user sharing mechanism in dynamic spectrum access networks," *IEEE Access*, vol. 7, pp. 32971–32988, 2019.

[4] Y. Xuhang, S. Roy, and R. Poovendran, "Pricing mechanisms for crown-sensed spatial-statistics-based radio mapping," *IEEE Transactions on Cognitive Communications and Networking*, vol. 3, no. 2, pp. 242–254, 2017.

[5] O. Elnahas, M. Elsabroute, O. Muta, and H. Furukawa, "Game theoretic approaches for cooperative spectrum sensing in energy-harvesting cognitive radio networks," *IEEE Access*, vol. 6, pp. 11086–11100, 2018.

[6] Y. Changyan, J. Cai, and G. Zhang, "Spectrum auction for differential secondary wireless service provision with time-dependent evaluation information," *IEEE Transactionson Wireless Communications*, vol. 16, no. 1, pp. 206–220, 2017.

[7] X. Dong, T. Zhang, D. Lu, G. Li, Y. Shen, and J. Ma, "Preserving geo-distinguishability of the primary user in dynamic spectrum sharing," *IEEE Transactions on Veterinary Technology*, vol. 68, no. 9, pp. 8881–8892, 2019.

[8] Y. Yuan and F. Y. Wang, "Towards blockchain based intelligent transportation systems," in *2016 IEEE 19th international conference on intelligent transportation systems (ITSC)*, pp. 2663–2668, Rio de Janeiro, Brazil, 2016.

[9] B. Leiding, P. Memarmoshrefi, and D. Hogrefe, "Self-managed and blockchain based vehicular ad-hoc networks," in *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct*, pp. 137–140, Heidelberg, Germany, 2016.

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

[11] M. Cebe, E. Ergin, K. Akkaya, H. Aksu, and S. Uluagac, "Block 4forensic: an integrated lightweight blockchain framework for forensics applications of connected vehicles," *IEEE Communications Magazine*, vol. 56, no. 10, pp. 50–57, 2018.

[12] M. Li, L. Zhu, and X. Lin, "Efficient and privacy preserving carpooling using blockchain assisted vehicular fog computing," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4573–4584, 2019.

[13] L. Liu, M. Zhao, M. Yu, M. A. Jan, D. Lan, and A. Taherkordi, "Mobility-Aware Multi-Hop Task Offloading for Autonomous Driving in Vehicular Edge Computing and Networks," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–14, 2022.

[14] C. Tan, X. Li, T. H. Luan, B. Gu, Y. Qu, and L. Gao, "Digital twin based remote resource sharing in internet of vehicles using consortium blockchain," in *2021 IEEE 94th Vehicular Technology Conference (VTC2021-Fall)*, pp. 1–6, Norman, OK, USA, 2021.

[15] Y. Qu, S. R. Pokhrel, S. Garg, L. Gao, and Y. Xiang, "A blockchained federated learning framework for cognitive computing in industry 4.0 networks," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 4, pp. 2964–2973, 2021.

[16] L. Liu, J. Feng, Q. Pei et al., "Blockchain-enabled secure data sharing scheme in mobile-edge computing: an asynchronous advantage actor–critic learning approach," *IEEE Internet of Things Journal*, vol. 8, no. 4, pp. 2342–2353, 2021.

[17] C. Peng, C. Wu, L. Gao, J. Zhang, K. L. Alvin Yau, and Y. Ji, "Blockchain for vehicular internet of things: recent advances and open issues," *Sensors*, vol. 20, no. 18, p. 5079, 2020.

[18] L. Gao, C. Wu, T. Yoshinaga, X. Chen, and Y. Ji, *Multi-channel blockchain scheme for internet of vehicles*, 2021.

[19] L. Hai, L. Xinghua, L. Bin et al., "A distributed K-anonymous location privacy protection scheme based on blockchain," *Journal of Computer Science*, vol. 42, no. 5, pp. 942–960, 2019.

Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 3091495, 11 pages
https://doi.org/10.1155/2022/3091495

Research Article

A Detection Algorithm for Audio Adversarial Examples in EI-Enhanced Automatic Speech Recognition

Ying Huang and Jie Liu

School of Information Engineering, Xi'an University, China

Correspondence should be addressed to Ying Huang; yhuang@xawl.edu.cn

Received 15 January 2022; Accepted 19 March 2022; Published 21 April 2022

Academic Editor: Celimuge Wu

Benefiting from the development of big data, edge computing, and deep learning, splendid breakthroughs have been made in automatic speech recognition (ASR) in recent years. Since then, more and more smart products have chosen speech as the interface for human-computer interaction, which causes popularity of edge intelligence (EI) enhanced automatic speech recognition. While people are enjoying the social changes brought by speech recognition technology, a factor of instability quietly emerged called audio adversarial example which is a type of audio deliberately generated by attackers via adding subtle perturbations to the original audio signal. The added perturbations which sound like certain noise that cannot be precepted by human but will cause ASR system make wrong transcription. Three detection algorithms for audio adversarial examples are proposed in this thesis, namely, the robust detection algorithm based on WER (word error rate), the feature detection algorithm based on ADR (adversarial ratio), and the collaborative detection algorithm based on neural network. The experiment results show that three detection algorithms proposed in this thesis have a great discrimination on audio adversarial examples and achieve high AUC scores. Among them, the cooperative detection is the best and the feature detection is the worst. In addition, we found that robust detection algorithm tends to have a higher accuracy score but a lower recall score, while feature detection algorithm tends to have the converse performance. Moreover, since the proposed collaborative detection method combines the advantages of the robust detection and feature detection methods, it presents a better performance with respect to accuracy, recall, and F1 score.

1. Introduction

With the evolution of deep learning, big data, and cloud computing technologies, the accuracy of speech recognition has improved substantially. The hardware cost of speech data storage is also continually dropping. These two trends have led to more and more smart products using speech as the interface of human-computer interaction, which has resulted in more opportunities for the intelligent speech industry [1]. According to statistics from Frost and Sullivan, the market for China's intelligent speech industry has gone from only 2.87 billion yuan in 2014 to 21.65 billion yuan in 2019, at a compound annual growth rate of 53.2%. Sullivan forecasts that China's intelligent speech market will reach 65.51 billion yuan in 2023.

With the merge of edge computing and artificial intelligence, intelligent speech applications enhanced by EI are now used in scenarios such as smart homes, smart cars, smart medical devices, and smart customer service [2]. Internet companies, intelligent speech technology companies, and smart speech start-ups are all players in the global market for intelligent speech products.

The increased availability of intelligent speech devices has helped users realize the value of instinctive expression when it comes to productivity, and consequently, users' lifestyles are changing. However, speech adversarial samples have emerged as a stumbling block. The adversarial sample in speech recognition is defined as a type of audio generated by an attacker in order to deliberately add subtle disturbances to the original audio. In terms of acoustic

characteristics, the speech adversarial sample has slightly more noise than the original audio, which the human ear is not sensitive to. However, it will cause an automatic speech recognition (ASR) system to transcribe errors. The widespread use of ASR systems in intelligent speech devices gives attackers more opportunities to do this. For example, attackers can generate adversarial samples against a certain type of ASR system in advance and then use social media to disseminate the adversarial samples. Adversarial samples could be input into the speech interface of a smart car, and then transcribed into a series of altered driving instructions, posing great danger to life and property. Therefore, it is very important to study the adversarial examples and defense mechanisms in speech recognition.

Nilaksh Das and Madhuri Shanbhogue et al. in [3] implemented the first interactive experimental tool, called Adagio, for audio adversarial samples, which can attack and defend the end-to-end Deep Speech model in real time visually and auditorily. In [4], Iustina Andronic et al. also discussed the possibility of MP3 compression as a defense against adversarial samples. Krishan Rajaratnam et al. [5] discussed the effect of combining audio preprocessing methods on speech classification models, using six preprocessing methods, including MP3 compression, AAC compression, bandpass filtering, and audio translation. Zhuolin Yang et al. [6] pointed out that speech is a time-domain signal with inherent time-dependent characteristics and that the introduction of antinoise can lead to the destruction of this dependence. Based on this assumption, we propose the concept of using temporal dependency (TD) for detection, which uses the ratio of the longest common prefix of partial and full transcription to the length of the entire text as a detection indicator. Tejas Jayashankar et al. [7] first proposed applying the concept of dropout [8] to the detection of audio adversarial samples. Victor Akinwande and Celia Cintas [9] introduced a novel idea, which regards the detection of adversarial samples as anomalous pattern (AP) detection in the ASR model space. The author assumes that the adversarial sample will cause abnormal activation of some nodes in the neural network. Based on this, the author uses the subset scan method to search for the most abnormal subset of data observations and then uses nonparametric scan statistics. This method quantifies the abnormality of the subset as a numerical score between 0 and 1, specifically the Berk–Jones test statistics [10] method. Qiang Zeng et al. [11] combined the fact that different ASR systems use different architectures, parameters, and training datasets to cause differences in the same audio transcription with the idea of multiversion programming [12], and proposed a novel method of adversarial sample detection, called MVP-EARS. This method uses ready-made ASR algorithms to determine whether the audio is an adversarial sample. Saeid Samizade et al. [13] proposed for the first time that the detection of adversarial samples is a binary classification problem. Based on this, this paper proposes to convolutional neural network (CNN) detection, which involves using the CNN model to train the detection method of adversarial samples and benign samples.

The main work of this paper in voice adversarial sample defense is as follows:

(1) We propose a robust detection algorithm based on word error rate (WER) which is based on the fact that adversarial samples are obtained by adding a small amount of noise to a normal sample. The algorithm detects the audio using spectral subtraction for noise reduction, then uses the WER to measure the impact of noise reduction on the audio, and then trains a classifier to differentiate adversarial samples from benign samples. The proposed approach is superior to other approach in theory and experimental results

(2) We propose a feature detection algorithm based on adversarial effect derived from the fake sample to improve the method of directly using the entire speech feature for detection. In order to characterize the adversarial nature of a certain frame and a certain speech, the proposed approach attempts to incorporate the characteristic of voice sample into neural networks, and finally, a classifier is trained to distinguish adversarial samples from the normal samples

(3) Aware of the single and linear characteristics of the above two methods, we pro-pose a neural network-based collaborative detection algorithm and introduce a binary neural network model to fit the nonlinear relationship between WER, adversarial degree, and adversarial samples, in order to further improve the security and discrimination capability of the detection algorithm. The robust detection algorithm based on WER and the calculation method based on adversarial degree are combined with the waveform characteristics of the voice itself to extract voice features. This combined method can better restore the audio itself and also provides a benign input feature for the calculation of the neural network, thereby ensuring the accuracy of the calculation results and a high recall rate

2. Related Work

2.1. Attack Model. ASR technology converts human speech into text [14–16]. From speech signals to text characters, ASR technology spans multiple basic and cutting-edge disciplines such as acoustics and linguistics, signal processing, computers, and artificial intelligence. Although research on speech recognition began as early as the 1950s, due to its complexity, the accuracy of speech recognition was not very high until the emergence of neural networks and the rise of end-to-end technology [17, 18]. Since then, the accuracy of speech recognition rate has been advancing rapidly. Compared with the traditional DNN-HMM [19] hybrid model, the end-to-end ASR system omits the steps of aligning text- and context-sensitive phonemes and can directly start training from the neural network without multiple iterations.

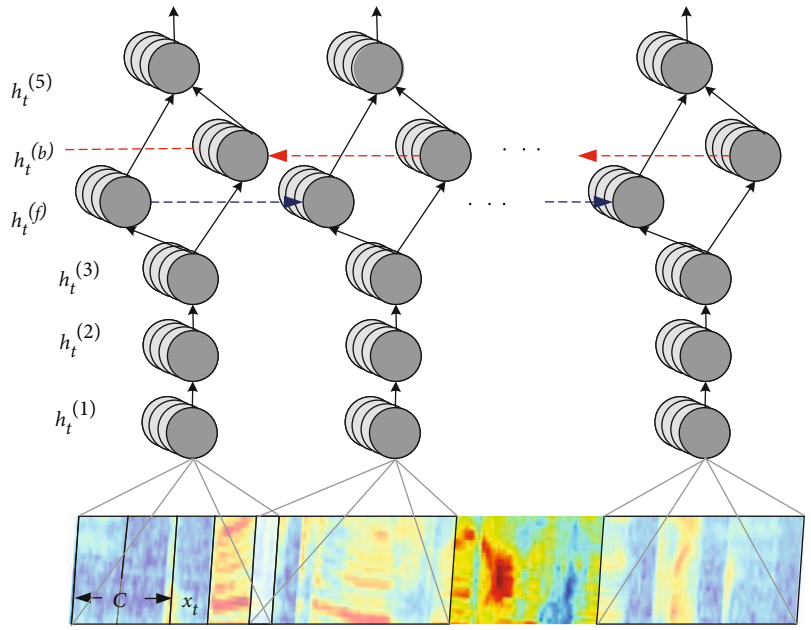


Figure 1: Deep Speech network model.

All the theoretical research and experiments in this paper are based on Deep Speech, an open source, end-to-end ASR system [20]. The network structure is shown in Figure 1. The MFCC feature of speech is used as input. The core is an RNN model with correctionist temporal classification (CTC) loss [21] as the loss function; the output is the probability distribution of the character sequence.

The Deep Speech model consists of 5 hidden layers. For input x, we use h^l to denote the l^{th} layer and h^0 to denote the input. The first 3 layers are fully connected layers. For the first layer, the input at time t is not only the characteristics x_t of time t, but also the characteristics of its front and back C frames, totaling $2C + 1$ frames. The first 3 layers are calculated by

$$h_t^l = g\left(W^l h_t^{l-1} + b^l\right), \tag{1}$$

where $g(z) = \min(\max(z, 0))$ and the maximum value is limited on the basis of ReLU, so it is also called Clipped ReLU. The fourth layer is a two-way RNN, as shown in

$$\begin{aligned} h_t^f &= g\left(W^4 h_t^3 + W_\tau^f h_{t-1}^f + b^4\right), \\ h_t^b &= g\left(W^4 h_t^3 + W_\tau^b h_{t+1}^b + b^4\right). \end{aligned} \tag{2}$$

The most common RNN is used here instead of LSTM/GRU in order to make the network structure simple and consistent and to facilitate the optimization of calculation speed. In this two-way RNN, the parameters input to the hidden unit are shared (including bias), and the RNN in each direction has its own hidden unit and hidden unit parameters. h^f is calculated from time 1 to time T, and h^b is calculated from time T in turn. The fifth layer will add the two outputs of the fourth layer bidirectional RNN as its input, as shown in

$$\begin{aligned} h_t^4 &= h_t^f + h_t^b, \\ h_t^5 &= g\left(W^5 h_t^4 + b^5\right). \end{aligned} \tag{3}$$

The last layer is a fully connected layer without activation function, which uses softmax to turn the output into a probability corresponding to each character, as shown in

$$h_{t,k}^6 = \hat{y}_{t,k} = P(c_t = k|x) = \frac{\exp\left(W_k^6 h_t^5 + b_k^6\right)}{\sum_j \exp\left(W_j^6 h_t^5 + b_j^6\right)}. \tag{4}$$

After calculating $P(c_t = k|x)$, CTC can be used to calculate $L(\hat{y}, y)$ and find the gradient of L to the parameter.

2.2. CW Attack. The CW attack is a white-box targeted attack against the ASR system, derived from the literature of Nicholas Carlini and David Wagner [22]. In this method, we propose to improve the CTC loss function y introducing the L2 norm of noise distortion and using the Adam optimizer to simultaneously optimize CTC loss and distortion to achieve a balance between distortion and CTC Loss. The loss function is shown by

$$\begin{aligned} &\text{minimize } |\delta|_2^2 + c \cdot l(x + \delta, t) \\ &\text{such that } \mathrm{dB}_x(\delta) \le \tau \\ &\mathrm{dB}_x(\delta) = \mathrm{dB}(\delta) - \mathrm{dB}(x) \end{aligned}, \tag{5}$$

where δ is the added noise; c is the weight; x is the original audio; l is the CTC loss function; t is the target transcription

text; $dB_x(\delta)$ is the distortion of the noise δ relative to the original waveform x, measured in decibels (dB); τ is the maximum distortion constant; and $dB(\cdot)$ is the logarithmic scale which is used to measure the relative loudness of audio or noise samples; the calculation method is shown in

$$dB(x) = \max_i 20 \cdot \log_{10}(x_i), \tag{6}$$

where x_i represents the value of the i^{th} sampling point of the waveform x.

Why can CW attacks be used to generate adversarial examples? It should be noted that the CTC loss function reflects the relationship between the target transcription text and its corresponding audio. When the audio does not match the text, the CTC loss is larger. Therefore, reducing the CTC loss of the audio to is equivalent to increasing the CTC loss of the audio and the original transcribed text, but the direction of its increase is to approach. In addition, to be able to converge as quickly as possible, we use the fast gradient thinking in the FGSM algorithm. In each iteration, loss is used to derive the added noise to obtain the gradient that makes the loss function change the fastest, and the disturbance noise is "updated" along this gradient direction until the transcription target is reached.

Figure 2 shows a comparison of the audio waveform before and after the CW attack. The original audio content is "but everything had changed," and the CW attack is successfully transcribed as "nothing is impossible." It is clear that the CW attack modifies the entire waveform. The pronunciation segment of the adversarial sample has a greater similarity to the original waveform, while the silent segment has a significant gap. In addition, compared with the original audio, we can hear obvious TV-like snowflake noise.

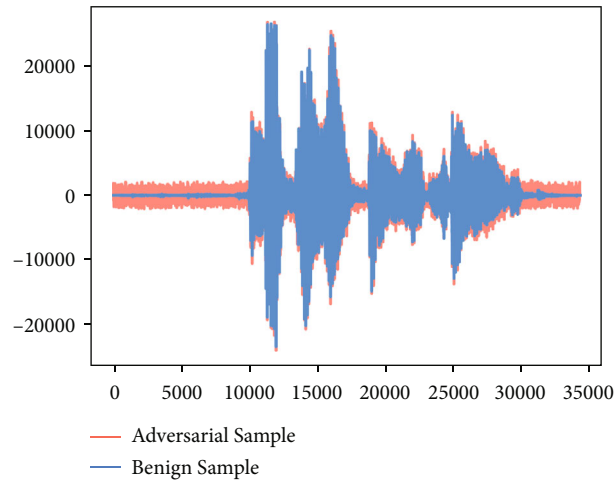


Figure 2: Audio waveform comparison before and after CW attack.

3. Algorithms

In this section, we introduce our proposed audio adversarial sample detection model. As shown in Figure 3, the model includes seven components.

(i) Speech interface module: Corresponding to the upper left corner of Figure 3, this module is responsible for detecting the legality of input audio files, that is, whether the sampling rate and format meet the specifications, and then converting the speech into the form of a one-dimensional vector

(ii) Noise reduction module: The core of the robust detection algorithm, this module is responsible for noise reduction and storage of audio. Here, it imitates the code style of the audio adversarial sample attack library and encapsulates all the noise reduction-related code into the denoise.py file. Called through the interface of the denoise input audio, the function returns the save path of the audio after noise reduction

(iii) Feature extraction module: This module, which forms the core of the feature detection algorithm, is responsible for extracting the feature vector of the filter banks of the audio. In addition, the module is responsible for the extraction of MFCC features; that is, another DCT operation is performed on the basis of the filter banks for Deep Speech voice recognition system input

(iv) Speech recognition system: This system is responsible for transcribing the input MFCC features into human-understandable text

(v) WER calculation module: (described in Section 3.1)

(vi) Adversity calculation module: (described in Section 3.2)

(vii) Two-class neural network: (described in Section 3.3)

3.1. Robust Detection Algorithm Based on Word Error Rate. The core of the robust detection algorithm based on WER is spectral subtraction noise reduction. In this method, the first spectral subtraction noise reduction is performed on the audio to be detected [23], and then, the audio is transcribed before and after noise reduction through the ASR system to calculate the WER of the audio to be detected. Finally, according to the differentiation of adversarial samples based on WER, a classifier is designed for detection. The algorithm principle and process are as follows:

According to the generation process of adversarial samples, let $y(n)$ be an audio adversarial sample with added antinoise, and then $y(n)$ is composed of original audio $x(n)$ and additive noise $d(n)$; that is, the form of the additive model is shown in

$$y(n) = x(n) + d(n). \tag{7}$$

The Fourier transform on both sides of the equation is shown in

$$Y(\omega) = X(\omega) + D(\omega). \tag{8}$$

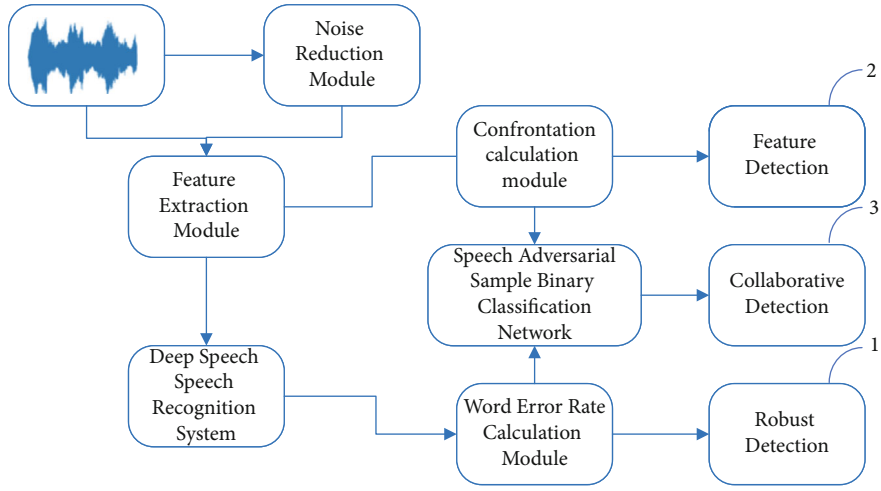


Figure 3: Speech adversarial sample detection model.

If expressed by the power spectrum, the form of the additive model is shown in

$$|Y(\omega)|^2 = |X(\omega)|^2 + |D(\omega)|^2 + 2\,\mathrm{Re}\left\{X(\omega)\overline{D(\omega)}\right\}. \quad (9)$$

Here, $2\,\mathrm{Re}\{X(\omega)\overline{D(\omega)}\}$ is called the cross term. Due to the vibration of the vocal organs, the speech signal is usually nonstationary. But if only one of the frames is intercepted, assuming that it is 10 to 30 ms, the speech in this frame has a stationary characteristic. In the same way, the noise signal is also stable or slowly changing at the microscopic scale. Therefore, it is considered that the mean value of the additive noise $d(n)$ is 0, and is not related to $x(n)$; that is, the cross term is 0. The above formula is simplified as

$$|X(\omega)|^2 = |Y(\omega)|^2 - |D(\omega)|^2. \quad (10)$$

In the speech signal, it is generally considered that there is no speech activity in the first few frames, so the first few frames can be regarded as pure noise signals; that is, the noise spectrum $|D(\omega)|^2$ can be estimated using these frames. Because the phase of the speech signal will not affect humans' understanding of speech, after obtaining the amplitude spectrum of the original audio, the phase of the speech adversarial sample can be used to approximate the speech phase of the original audio. At this time, an approximation original audio can be obtained in theory.

Further, we use the audio before and after noise reduction to obtain the reference text and the predicted text through the ASR system. We hope that the impact of noise reduction can be reflected in the differences in the text. To measure the inconsistency between two paragraphs of text, this method uses WER.

WER is generally used to compare predicted text and reference text in units of characters and to quantify the difference between the two texts. It is typically used to measure the performance of an ASR system and is a key indicator in the field of speech recognition. Its calculation formula is shown in

$$\mathrm{WER} = \frac{S+D+I}{N} = \frac{S+D+I}{S+D+C}. \quad (11)$$

Note that S is the number of words that need to be replaced in the reference text, D is the number of words that need to be deleted in the reference text, I is the number of words that need to be inserted into the reference text, and C is the correct number of words in the reference text. As a result, $N = S + D + C$ is the character length of the reference text.

The numerator of the WER is equivalent to the edit distance of two paragraphs of text [24]. Editing distance is defined as the minimum operation required to change from one text to another. The executable operations include replacing a character, deleting a character, and inserting a character. The industry has a classic dynamic programming solution to this problem, and its state transition equation is shown in

$$\mathrm{DP}_{i,j} = \min \begin{cases} \mathrm{DP}_{i-1,j-1} + 0 & \text{if } h_i = r_j \\ \mathrm{DP}_{i-1,j-1} + 1 & (\text{Substitution}) \\ \mathrm{DP}_{i,j-1} + 1 & (\text{Insertion}) \\ \mathrm{DP}_{i-1,j} + 1 & (\text{Deletion}) \end{cases}, , \quad (12)$$

where h is the predicted text, r is the reference text, and DP is the matrix used for state transition which dimension is $|h| \times |r|$.

3.2. Feature Detection Algorithm Based on Adversarial Degree. The core of the feature detection algorithm based on adversarial degree is the extraction and application of filter banks features. In this method, we first extract the filter banks of the audio to be detected and then calculate the antagonism of the audio to be detected based on the filter

banks. Finally, according to the degree of discrimination of adversarial samples, a classifier is designed for detection. The algorithm principle and process are as follows.

3.2.1. Pre-Emphasis. The first step of the algorithm is to apply a pre-emphasis filter to the signal. Compared with the low frequency, the high frequency usually has a smaller amplitude, so the pre-emphasis filter can be used to balance the spectrum and amplify the high frequency. In addition, the pre-emphasis can also avoid numerical problems during the Fourier transform operation and improve the signal-to-noise ratio (SNR). The specific calculation formula is shown in

$$y(t) = x(t) - \alpha x(t-1), \tag{13}$$

where x is the speech signal, y is the signal after pre-emphasis, and α is the pre-emphasis coefficient, which is generally selected as 0.95 or 0.97.

3.2.2. Framing. After pre-emphasis, the signal needs to be divided into short-time frames. Under normal circumstances, the frequency in the speech signal is not static, and the Fourier transform of the entire speech signal will lose the frequency contour of the signal. Therefore, the signal is also processed in units of frames, and then, the approximate value of the signal frequency profile is obtained by merging adjacent frames. In speech processing, the frame size is usually set to 25 ms.

3.2.3. Window Adding. After the signal is cut into frames, a window function such as a Hamming window needs to be applied to each frame to offset the assumption of unlimited data made by the fast Fourier transform (FFT) and reduce spectrum leakage. The form of the Hamming window is shown in

$$w[n] = 0.54 - 0.46\cos\left(\frac{2\pi n}{N-1}\right), \tag{14}$$

where $0 \le n \le N-1$ and N is the length of the window.

3.2.4. Fourier Transform and Power Spectrum. Next, we perform short-time Fourier transform on each frame, which is also called N-point FFT. N is usually 256 or 512. The power spectrum calculation formula is shown in

$$P = \frac{|\mathrm{FFT}(x_i)|^2}{N}, \tag{15}$$

where x_i is the i^{th} frame audio signal.

3.2.5. Filter Banks. Finally, the Mel-level triangular filter (usually 40 filters) is applied to the power spectrum to extract the filter banks features. The Mel scale imitates the human ear's perception of sound; that is, it has a higher discriminative power at a lower frequency and a lower discriminatory power at a higher frequency. The conversion formula of Hertz f and Mel m is shown in

$$\begin{aligned} m &= 2595\log_{10}\left(1 + \frac{f}{700}\right), \\ f &= 700\left(10^{m/2595} - 1\right). \end{aligned} \tag{16}$$

Each filter in the Mel filter bank is triangular, with a response of 1 at the center frequency, and linearly decreases to 0 toward the center frequency of the adjacent filters. The filters in the Mel filter bank are shown in

$$H_m(k) = \begin{cases} 0 & k < f(m-1) \\ \dfrac{k - f(m-1)}{f(m) - f(m-1)} & f(m-1) \le k < f(m) \\ 1 & k = f(m) \\ \dfrac{f(m+1) - k}{f(m+1) - f(m)} & f(m) < k \le f(m+1) \\ 0 & k > f(m+1) \end{cases}, \tag{17}$$

where m is the subscript of the filter bank; in this method $1 \le m \le 40$, $f(m)$ is the center frequency of the m^{th} triangular filter, and $H_m(k)$ represents the response of the m^{th} triangular filter at k Hz. Because the human ear's perception of sound is not linear, it is necessary to use the log function for nonlinear processing at the end.

3.2.6. Confrontation. Statistical observation of the filter banks of benign and adversarial samples reveals that the probability of positive values in the filter banks features of adversarial samples is significantly higher than that of benign audio samples. In addition, the longer the noise duration, the greater the noise amplitude, and the greater the probability of a positive value. Based on this, we propose the concept of adversarial frames. Frames with nonnegative filter banks feature values are regarded as adversarial frames, which indicate the degree of disbelief in this frame.

Furthermore, the speech signal is counted in units of frames in filter banks, and the concept of adversarial degree is proposed, which represents the proportion of adversarial frames in the audio. The greater the proportion, the greater the degree of disbelief in the audio, that is, the greater the possibility of treating it as a confrontational sample. The smaller the proportion, the more authentic the audio is. The calculation method of antagonism is shown in

$$\mathrm{ADR} = \frac{\sum_i (f \ge 0 \forall f \in \mathrm{fea}_i)}{N}, \tag{18}$$

where fea is the feature matrix of the audio filter banks and N is the first dimension of fea, which is related to the audio duration.

3.3. Collaborative Detection Algorithm Based on Neural Network. All single detection algorithms may bring about the problem of insufficient robustness, and the algorithm proposed in this paper is no exception. An attacker can deliberately reduce a single index to carry out more advanced secondary attacks. In addition, the binary classification in the real scene is usually not linearly separable, and all the methods that use linear classification will inevitably result in the lack of a certain performance index of accuracy or recall. Therefore, to further improve the robustness of the model and the algorithm's discrimination against adversarial samples, we combine the methods proposed in Sections 3.1 and 3.2 to provide a better detection method.

Here, we regard WER and adversarial degree as the characteristics of artificially extracted speech samples and then use neural network for nonlinear fitting training to achieve the effects of classification and detection.

In this paper, a lightweight binary neural network is selected. In addition to the input layer and the output layer, it only includes two hidden layers and the corresponding dropout layer, as shown in Table 1.

Table 1: Two-class neural network architecture.

Layer (type)	Output shape
dense_1 (dense)	(None, 64)
activation_1 (activation)	(None, 64)
dropout_1 (dropout)	(None, 64)
dense_2 (dense)	(None, 64)
activation_2 (activation)	(None, 64)
dropout_2 dDropout)	(None, 64)
dense_3 (dense)	(None, 1)
activation_3 (activation)	(None, 1)

4. Implementation and Evaluation

4.1. Databases. The Common Voice [25] corpus is an initiative from Mozilla, which contains six files with tab-separated values (TSV files) and a single clips subdirectory that contains all of the audio data, where each of the six TSV files represents a different segment of the voice data, with all six having the following column headers: [client_id, path, sentence, up votes, down_votes, age, gender, accent]. It is a collection of self-recorded voices uploaded by many users on the Common Voice website. The text content comes from many public domains, such as blog posts submitted by users, old books, movies, and other public speeches. According to Mozilla, the main purpose of the project is to train and test the ASR system. The goal is to help teach machines how to speak, but Mozilla also encourages its use for other purposes.

The Common Voice corpus is divided into three parts. The "valid" subset is the audio that has been heard by at least two people and that most of the listeners think matches the text. The "invalid" subset contains the audios that do not match their corresponding text judged by at least 2 persons. And the remaining audios form the subset named "other". Furthermore, "valid" and "other" are divided into three parts: "dev" is used for development and experimentation, "train" is used for speech recognition training, and "test" is used for testing WER.

Considering the cost in time and hardware, this paper finally selected 8071 audio files of "cv-valid-dev" and "cv-valid-test" as the preliminary screening of the dataset.

The audio files of the Common Voice corpus are all in .mp3 format. Therefore, first, the format conversion of the preliminary audio files is required, and then Deep Speech is used for transcription, and the WER index is tested. According to the results, Deep Speech reached an average WER of 7.33% and performed well on the preliminary screening dataset. Finally, this paper screened out 1,200 speech samples with a WER of 0 and the length of the transcribed text not exceeding 57 as the experimental benign sample dataset and used the CW attack to generate the corresponding adversarial sample dataset. If the length of the transcribed text of the benign sample does not exceed 37, the CW attack target is set to "nothing is impossible"; otherwise, the attack target is set to "if winter comes can spring be far behind?"

4.2. Environment. The hardware environment of the experiment in this article is shown in Table 2.

The software environment of the experiment in this paper is shown in Table 3.

In Table 3, Deep Speech code is Mozilla's code implementation of Deep Speech's speech recognition model, and Deep Speech model is a trained model file that stores the weights, biases, gradients, and other variable values of the model. We need to pay attention to compatibility when using the Python package. The adapted version number is given here. CUDA and cuDNN are drivers that need to be installed when using Nvidia graphics cards. We can also install the TensorFlow version, if it is the correct version.

4.3. Indicators. In order to better introduce the two-category index, a confusion matrix is first introduced here. The confusion matrix is shown in Table 4.

4.3.1. ACC. ACC indicates the proportion of samples with correct predictions to the total samples. The ACC calculation formula is shown in

$$\mathrm{ACC} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}}. \tag{19}$$

4.3.2. AUC. The area under the curve (AUC) represents the area under the ROC curve. Here, in order to better understand the ROC curve, we first introduce the true positive rate (TPR) and false positive rate (FPR). TPR represents the proportion of all positive samples in the dataset that are correctly predicted. The calculation formula is shown in

$$\mathrm{FPR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}. \tag{20}$$

Table 2: Hardware environment.

Items	Parameter
CPU	Intel Xeon E3-1230v5
RAM	16GB DDR4
GPU	NVIDIA Quadro K420
Storage	1 T SSD

Table 3: Software environment.

(a)

Component	Version
Ubuntu	16.04
Python	3.5.2
Deep Speech code	0.4.1
Deep Speech model	0.4.1
CUDA	9.0
cuDNN	7.0

(b)

Python packages	Version
pandas	0.24.0
numpy	1.16.4
Keras	2.2.4
ds-ctcdecoder	0.4.1
tensoflow-gpu	1.12.0
scipy	1.4.1

Table 4: Confusion matrix.

		Actual result	
		P	N
Prediction	P	TP	FP
	N	FN	TN

where T and F represent true and false, and P and N represent positive and negative.

FPR represents the proportion of negative samples that are predicted to be positive samples. The calculation formula is shown in

$$\text{FPR} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{21}$$

Every time a threshold is set, a set of TPR and FPR values can be obtained. Therefore, the score of each sample in the test set is set as a threshold, so that multiple sets of TPR and FPR values can be obtained. At this time, FPR is used as the abscissa and TPR as the ordinate to draw the ROC curve.

4.3.3. Precision. Precision represents the proportion of positive samples that are correctly predicted to all positive samples. The calculation formula is shown in

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \tag{22}$$

4.3.4. Recall. Recall represents the proportion of positive samples that are correctly predicted to all positive samples. It basically has the same meaning as the true rate, except that the name is different. The calculation formula is shown in

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{23}$$

4.3.5. F1 Score. F1 score is defined as the harmonic mean of precision and recall. The calculation formula is shown in

$$F_1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{24}$$

4.4. Result and Analysis. Figures 4 and 5 show the comparison diagrams of WER and adversarial degree distribution of benign samples and adversarial samples generated by CW attacks. According to the WER distribution map, it is clear that the WER of the benign samples is concentrated in the range of 0 to 0.1, while the WER of nearly every adversarial sample is greater than 0.1. Furthermore, according to the adversarial degree distribution map, it is clear that the WER of the adversarial samples is concentrated in the range of 0.9 to 1.0, while the benign samples have a wider distribution range, but most of them are less than 0.9. Therefore, the WER and adversarial degree have a good degree of success in differentiating the adversarial samples generated by the CW attack. In terms of the distribution ratio, the differentiation capability of adversarial degree is slightly weaker than that of the WER.

Figure 6 shows the joint distribution diagram of WER and adversarial degree of the benign sample and the adversarial sample generated by the CW attack. The boundary between the benign sample and the adversarial sample is relatively clear, with points crossing only sporadically. It can be concluded that the collaborative detection algorithm is very successful at differentiating the adversarial samples generated by the CW attack.

Figure 7 shows the ROC curves of the three detection algorithms against CW attacks. The upper left corners of the three curves are infinitely close to the $(0, 1)$ point, and the AUC value is greater than 0.99, indicating that these three algorithms perform very well in detecting CW attacks. In addition, the ROC curve of the collaborative detection completely covers the other two curves, indicating that the performance of the collaborative detection algorithm is better than that of a single detection. Because a single detection has a high degree of discrimination against the adversarial samples generated by the CW attack, the improvement of the discrimination degree by coordinated detection is limited. The ROC curves of robust detection and feature

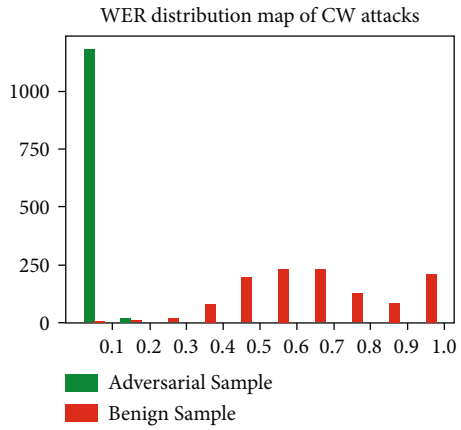


Figure 4: WER distribution map of CW attacks.

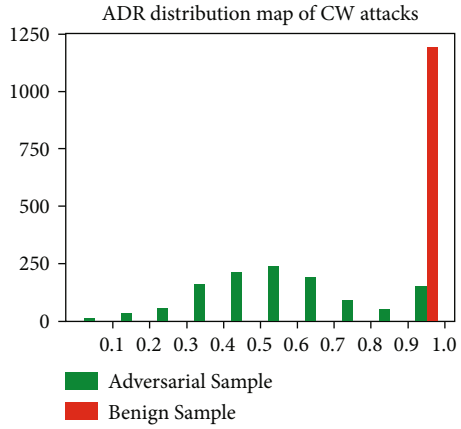


Figure 5: ADR distribution map of CW attacks.

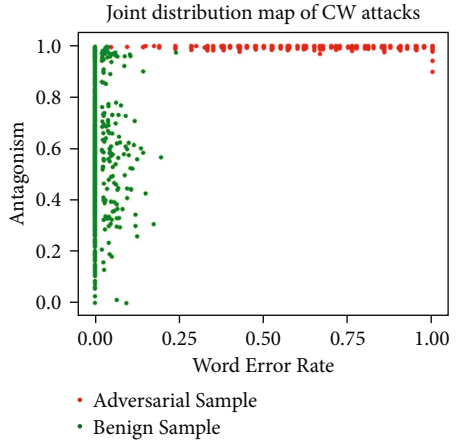


Figure 6: Joint distribution map of CW attacks.

detection overlap. In terms of the AUC value, the robust detection performs the best.

Next, we will further discuss the algorithm in terms of its specific performance on the test set, that is, ACC, accuracy, recall, and F1 score. Table 5 shows the CW attack detection

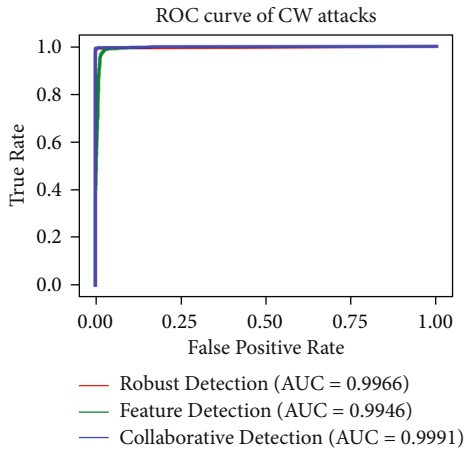


Figure 7: ROC curve of CW attacks.

Table 5: Indicators of CW attacks.

	Robust detection	Feature detection	Collaborative detection
ACC	0.9950	0.9817	*0.9967*
AUC	0.9966	0.9946	*0.9991*
Precision	0.9983	0.9738	*0.9983*
Recall	0.9667	0.9900	*0.9967*
F1 score	0.9822	0.9818	*0.9975*

indicators. By comparing the three detection algorithms, we can draw three conclusions.

(1) The three detection algorithms all have a good degree of discrimination against adversarial samples in CW attack scenarios, with the collaborative detection algorithm the best, followed by the robust detection and the feature detection. Because the collaborative algorithm also detects the resistance robust detection characteristics, such as WER, and the feature-sensitive characteristics of neural networks, its success rate is higher than that of the other two. Feature changes often affect the robustness of the system, so in terms of index values, robustness, and detection methods are better than feature detection methods

(2) Robust detection algorithms based on WER tend to have a higher accuracy rate, but the recall rate is low. Feature detection algorithms based on adversarial degree tend to have a higher recall rate, but the accuracy rate is low. It shows that the robust detection algorithm based on the suberror rate has higher accuracy in the retrieval accuracy rate than the feature detection based on adversarial degree. This also confirms the conclusion (1) from another aspect, that is, the robust detection method better than feature detection methods

(3) The collaborative detection algorithm based on neural network improves the discrimination capability

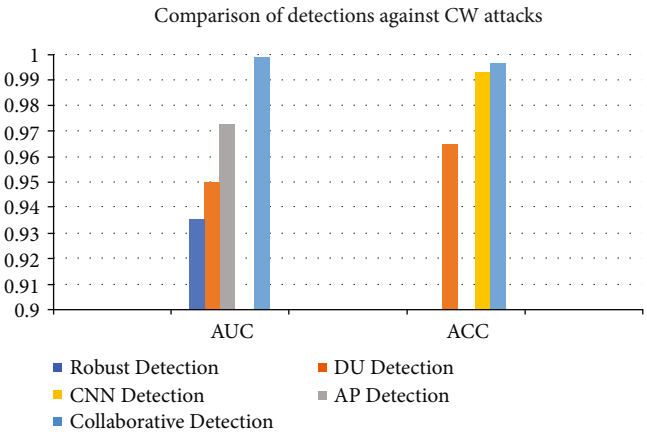


Figure 8: Comparison of detections against CW attacks.

of voice adversarial samples more than single detection. In addition, the collaborative detection algorithm integrates the advantages of robust detection and feature detection, making it have a higher accuracy rate and recall rate, as well as a more balanced detection capability

4.5. Comparisons. In this section, we compare the best performing collaborative detection algorithm with the existing detection algorithms, which are TD detection, DU detection, AP detection, and CNN detection. The comparison results are shown in Figure 8. Where there is no bar, it indicates that the author did not use the corresponding indicators.

Figure 8 shows that the detection scheme proposed in this paper achieves higher scores in both AUC and ACC indicators than the existing detection schemes, indicating that the collaborative detection algorithm has a stronger capability to detect CW attacks.

5. Conclusions

In recent years, thanks to China's favorable policies for artificial intelligence and the relatively mature technologies of speech recognition, big data, and cloud computing, the country's intelligent voice industry has experienced a period of rapid development. However, the popular ASR systems are suffering from the severe threat of audio adversarial samples. Adding even a slight disturbance to original audios, that is difficult to be detected by listeners, will make these systems output erroneous transcriptions. This poses a serious threat to the security of smart voice devices, which is the focus of this article's research.

This paper proposes three detection schemes for detecting adversarial samples: a robust detection algorithm based on word error rate, a feature detection algorithm based on adversarial degree, and a collaborative detection algorithm based on neural network. The robust detection algorithm is based on WER from the perspective of generating voice confrontation samples. It proposes the idea of using spectral subtraction noise reduction to destroy the artificially added perturbation in the confrontation sample and then uses WER as a measurement standard for detection. From the perspective of voice features, the feature detection algorithm based on adversarial degree proposes two concepts: detecting the filter banks feature of the speech frame as a unit and adversarial frame and adversarial degree. The algorithm uses these as the detection criteria. Considering the problems that might be caused by single linear detection, the neural network-based collaborative detection algorithm combines WER and adversarial degree to jointly detect voice adversarial samples by training a neural network.

The experimental results show that all three detection algorithms display good discrimination against CW attacks, with the collaborative detection performance the best, followed by robust detection and then feature detection. The results also show that robust detection algorithms tend to have higher accuracy, but the recall rate is low. The feature detection algorithms tend to have higher recall, but the accuracy is low. The collaborative detection algorithm integrates the advantages of robust detection and feature detection. While improving the overall discrimination, it also has a higher accuracy rate and recall rate, as well as a more balanced detection ability, which proves the necessity of joint detection.

Although the results show that the research in this paper has achieved good results, it should be noted that the adversarial samples studied in this paper are directly input to the voice interface and cannot form an attack effect after being broadcast in the air. However, the latest work [26, 27] shows that although there are many restrictions, there are already adversarial samples that can be played and then attacked. Therefore, it is important to continue research on the defense of voice adversarial samples. In addition, the author believes that it is very valuable to use each frame of speech as a unit of detection, but due to time limitations, this could not be addressed in this paper. Future research will explore this.

Data Availability

The Common Voice [25] corpus is an initiative from Mozilla. It is a collection of self-recorded voices uploaded by many users on the Common Voice website. The text content comes from many public domains, such as blog posts submitted by users, old books, movies, and other public speeches. According to Mozilla, the main purpose of the project is to train and test the ASR system. The goal is to help teach machines how to speak, but Mozilla also encourages its use for other purposes.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] J. Feng, L. Liu, Q. Pei, and K. Li, "Min-max cost optimization for efficient hierarchical federated learning in wireless edge networks," *IEEE Transactions on Parallel and Distributed Systems*, p. 1, 2022.

[2] L. Liu, M. Zhao, M. Yu, M. A. Jan, D. Lan, and A. Taherkordi, "Mobility-aware multi-hop task offloading for autonomous driving in vehicular edge computing and networks," *IEEE*

Transactions on Intelligent Transportation Systems, pp. 1–14, 2022.

[3] N. Das, M. Shanbhogue, S. T. Chen, L. Chen, M. E. Kounavis, and D. H. Chau, "Adagio: Interactive experimentation with adversarial attack and defense for audio," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 677–681, Springer, Cham, 2018.

[4] I. Andronic, L. Kürzinger, E. R. Chavez Rosas, G. Rigoll, and B. U. Seeber, "MP3 compression to diminish adversarial noise in end-to-end speech recognition," in *Speech and Computer. SPECOM 2020. Lecture Notes in Computer Science, vol 12335*pp. 22–34, Springer, Cham.

[5] K. Rajaratnam, K. Shah, and J. Kalita, "Isolated and ensemble audio preprocessing methods for detecting adversarial examples against automatic speech recognition," http://arxiv.org/abs/1809.04397.

[6] Z. Yang, B. Li, P. Y. Chen, and D. Song, "Characterizing audio adversarial examples using temporal dependency," http://arxiv.org/abs/1809.10875.

[7] T. Jayashankar, J. L. Roux, and P. Moulin, "Detecting audio attacks on ASR systems with dropout uncertainty," http://arxiv.org/abs/2006.01906.

[8] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[9] V. Akinwande, C. Cintas, S. Speakman, and S. Sridharan, "Identifying audio adversarial examples via anomalous pattern detection," http://arxiv.org/abs/2002.05463.

[10] R. H. Berk and D. H. Jones, "Goodness-of-fit test statistics that dominate the Kolmogorov statistics," *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete*, vol. 47, no. 1, pp. 47–59, 1979.

[11] Q. Zeng, J. Su, C. Fu et al., "A multiversion programming inspired approach to detecting audio adversarial examples," in *2019 49th annual IEEE/IFIP international conference on dependable systems and networks (DSN)*, pp. 39–51, IEEE, Portland, OR, USA, 2019.

[12] A. Avizienis and L. Chen, "On the implementation of N-version programming for software fault-tolerance during program execution," in *International Computer Software and Applications Conference (COMPSAC)*, pp. 149–155, IEEE, Chicago, USA, 1977.

[13] S. Samizade, Z. H. Tan, C. Shen, and X. Guan, "Adversarial example detection by classification for deep speech recognition," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3102–3106, Barcelona, Spain, 2020.

[14] Y. Gaur, W. S. Lasecki, F. Metze, and J. P. Bigham, "The effects of automatic speech recognition quality on human transcription latency," in *Proceedings of the 13th International Web for All Conference*, pp. 1–8, New York, 2016.

[15] H. Ibrahim and A. Varol, "A study on automatic speech recognition systems," in *2020 8th International Symposium on Digital Forensics and Security (ISDFS)*, pp. 1–5, Beirut, Lebanon, 2020.

[16] X. Lu, S. Li, and M. Fujimoto, "Automatic speech recognition speech-to-speech translation," in *SpringerBriefs in Computer Science*, Y. Kidawara, E. Sumita, and H. Kawai, Eds., pp. 21–38, Springer, Singapore, 2020.

[17] D. Wang, X. Wang, and S. Lv, "An overview of end-to-end automatic speech recognition," *Symmetry*, vol. 11, no. 8, p. 1018, 2019.

[18] L. Lu, X. Zhang, and S. Renais, "On training the recurrent neural network encoder-decoder for large vocabulary end-to-end speech recognition," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5060–5064, Shanghai, China, 2016.

[19] J. Novoa, J. Wuth, J. P. Escudero, J. R. Fredes, R. Mahu, and N. B. Yoma, "DNN-HMM based automatic speech recognition for HRI scenarios," in *The 13th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 150–159, IEEE/ACM, Chicago, USA, 2018.

[20] A. Hannun, C. Case, J. Casper et al., "Deep speech: scaling up end-to-end speech recognition," http://arxiv.org/abs/1412.5567.

[21] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd International Conference on Machine Learning*, pp. 369–376, New York, 2006.

[22] N. Carlini and D. Wagner, "Audio adversarial examples: targeted attacks on speech-to-text," in *2018 IEEE Security and Privacy Workshops (SPW)*, pp. 1–7, San Francisco, CA, USA, 2018.

[23] M. Berouti, R. Schwartz, and J. Makhoul, "Enhancement of speech corrupted by acoustic noise," in *ICASSP'79. IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 208–211, Washington, DC, USA, 1979.

[24] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals," *Soviet physics doklady*, vol. 10, no. 8, pp. 707–710, 1966.

[25] R. Ardila, M. Branson, K. Davis et al., "Common voice: a massively-multilingual speech corpus," http://arxiv.org/abs/1912.06670.

[26] Y. Qin, N. Carlini, G. Cottrell, I. Goodfellow, and C. Raffel, "Imperceptible, robust, and targeted adversarial examples for automatic speech recognition," in *International Conference on Machine Learning (ICML)*, pp. 5231–5240, PMLR, Long Beach, CA, USA, 2019.

[27] L. Schönherr, T. Eisenhofer, S. Zeiler, T. Holz, and D. Kolossa, "Imperio: robust over-the-air adversarial examples for automatic speech recognition systems," in *Annual Computer Security Applications Conference*, pp. 843–855, New York, 2020.

Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 8350006, 15 pages
https://doi.org/10.1155/2022/8350006

Research Article

Novel Searchable Attribute-Based Encryption for the Internet of Things

Zhenhua Lu,[1] **Yuyan Guo**,[1] **Jiguo Li**,[2] **Weina Jia**,[3] **Liping Lv**,[3] **and Jie Shen**[4]

[1]*College of Computer Science and Technology, Huaibei Normal University, Huaibei, Anhui 235000, China*
[2]*College of Computer and Cyber Security, Fujian Normal University, Fuzhou, Fujian 350117, China*
[3]*College of Information Engineering, Zhengzhou Shengda University, Xinzheng, Henan 451100, China*
[4]*School of Mechanical Engineering, North China University of Water Resources and Electric Power, Zhengzhou, Henan 450000, China*

Correspondence should be addressed to Yuyan Guo; guoyuyan428@163.com

Received 16 January 2022; Accepted 8 March 2022; Published 11 April 2022

Academic Editor: Qingqi Pei

As the product of the third information technology revolution, the Internet of Things (IoT) has greatly altered our way of lifetime. Cloud storage has gradually become the best choice for data processing due to its scalability and flexibility. However, the cloud is not a completely trusted entity, such as tampering with user data or leaking personal privacy. Therefore, cloud storage usually adopts attribute-based encryption schemes to accomplish data confidentiality and fine-grained access control. However, applying the ABE scheme to the Internet of Things still faces many challenges, such as dynamic user revocation, data sharing, and excessive computational burden. In this paper, we propose a novel searchable attribute encryption system that replaces the traditional key generation center with consortium blockchain to generate and manage partial keys. In addition, our scheme can perform predecryption operations in the cloud, and users only need to spend a small amount of computational cost to achieve decryption operations. Security analysis proves that our scheme achieves security under both the chosen keyword attack and the chosen plaintext attack. Compared with other schemes, this scheme is more economical in terms of computing and storage.

1. Introduction

The Internet of Things (IoT) originated in the media field, and it is an important part of the new generation of information technology. It connects all items to the Internet through information sensing devices to achieve positioning, tracking, supervision, and intelligent identification [1]. Simply put, the IoT is the Internet that connects everything. In recent years, the IoT has gradually become digitized in the real world, reduced the dispersion of information, and integrated the digital information between objects. The IoT is widely used in the fields of transportation and logistics, industrial manufacturing, medical care, and smart environments [2–7].

Sahai and Waters first proposed the concept of attribute-based encryption (ABE) in 2005 [8]. Because the access structure needs to be more flexible to adapt to more application scenarios, Goyal et al. [9] and Bethencourt et al. [10], respectively, proposed the concepts of key policy ABE (KP-ABE) and ciphertext policy ABE (CP-ABE). In KP-ABE, the ciphertext is associated with a set of attributes, and the access policy is embedded in the key. In contrast, in CP-ABE, the ciphertext is associated with the access policy, and a set of attributes is embedded in the key [11–14].

Although IoT has now blossomed in various fields, there are still many problems in applying ABE to the IoT. Compared with the traditional Internet, the IoT lacks standardization. In addition, the IoT itself is a complex network system involving many application domains, which is difficult to manage [15]. This makes it challenging to achieve its security and privacy. Therefore, blockchain technology is widely used in IoT for its persistence, anonymity, and auditability [16].

In this article, we will present a new blockchain-aided searchable attribute-based encryption (BC-SABE) scheme. This scheme replaces a traditional centralized server using

a distributed consortium blockchain consisting of a predefined set of trusted consensus nodes. Our main contributions can be summarized as follows:

(i) We present a novel BC-SABE scheme. Our scheme uses a distributed consortium blockchain containing a set of credible consensus nodes to achieve the function of key generation. Pedersen secret sharing protocol [17] and reciprocal protocol [18] are used to generate all secret parameters, which also means that the master key is not needed. Our scheme can also support keyword search under cloud assistance. Users only need to provide user identity information and partial token information to the blockchain, and the cloud server will receive the complete token from the blockchain and search for it. Moreover, the scheme can realize predecryption in the cloud, which can greatly reduce the burden of users

(ii) In our scheme, the Pedersen secret sharing protocol enables the sharing of subsecrets between consensus nodes, and each consensus node can combine the subsecrets as the master secret. And the reciprocal protocol ensures that the key information is shared without a trusted party

(iii) In addition, we use blockchain technology to realize the dynamic revocation of users. Because consensus nodes in the blockchain can use time period tags and status tags to update the user revocation list, this also means that we can use the blockchain to update the user revocation list to achieve user revocation. And our scheme does not require the user to re-encrypt the ciphertext for user revocation

Our scheme can also be applied in simple medical scenarios. For example, hospital can register admission information for patient and store the admission information in the blockchain. Registered patients can enter their data and information into the cloud server. When searching for relevant information, he/she only needs to submit a partial token to blockchain, and then, blockchain can produce the complete token for him/her and send it to the cloud server for search operations. In addition, patients can access data information from the cloud, which will first generate a predecryption key for them, and the patient can fully decrypt the data with a simple calculation.

The remainder of this paper is organized as follows. In Section 2, we reviewed some related work, and then, we gave some preliminaries in Section 3, including the complexity assumptions, binary trees, and blockchain. The system model, system procedure, and security model are given in Section 4, and the detailed structure of the system is given in Section 5. We give the security proof of the scheme in Section 6 and compare the performance of the scheme in Section 7, and finally, we give a brief summary in Section 8.

2. Related Work

Since the concepts of ABE were introduced, many improved ABE schemes have been proposed, such as traceable ABE [19], anonymous ABE [20, 21], and hierarchical ABE [22–25]. However, the application of ABE to IoT is still an issue that needs to be discussed. The resources of the Internet of Things devices are limited, and most of the ABE algorithm encryption and decryption calculation costs are relatively large, so the outsourcing ABE scheme has been proposed [26–29]. In addition, IoT systems should be able to revoke malicious users and update legitimate users with new attributes. How to implement dynamic user revocation is also an issue. Liu et al. [30] proposed a direct revocation scheme; however, in this scheme, all data owners are required to maintain the revocation list. Recently, Cui et al. [31] proposed a server-aided revocable ABE scheme. This scheme outsources all the workload to the server at the time of user revocation, and each user stores only a fixed size private key.

However, this single authorization ABE scheme has limited security and cannot carry a large number of IoT devices. Many multiauthorization-based ABE schemes have been proposed by scholars [32–35]. Chase [32] first presented the concept of multiauthority attribute-based encryption (MA-ABE) in 2007. Recently, Belguith et al. [34] presented a policy-hidden outsourced MA-ABE scheme, which hides the access structure to protect user privacy. In [35], a new MA-CP-ABE scheme was presented by Sethi et al. This system decentralizes authority and can support white box traceability along with outsourcing decryption.

Recently, the widespread application of data sharing has deepened the academic research on searchable encryption schemes, and many searchable attribute-based encryption (SABE) schemes have also been proposed [36–44]. In [36], Miao et al. presented a multikeyword SABE scheme, which supports comparable attributes by using 0-encoding and 1-encoding. Xu et al. [37] proposed for the first time a decentralized attribute-based keyword search (ABKS) scheme for multikeyword search in cloud storage. In this scheme, data sharing and data searching can be achieved without a fully trusted central authority. Recently, a seed string searchable ABE (SSS-ABE) solution for sharing and querying encrypted data was proposed by Sun et al. [38]; data users can query the entire ciphertext by substring without presetting keywords.

Blockchain technology originated from Bitcoin; the concept of blockchain was first proposed by Satoshi Nakamoto in [45]. Blockchain is a distributed shared ledger and database. Compared with the previous centralized accounting model, blockchain can achieve decentralization, which means removing the trust intermediary, which also makes the transactions on blockchain more open and transparent. Recently, Wu et al. [46] used blockchain technology to propose a privacy-protected and traceable ABE scheme, using blockchain to achieve data integrity and nonrepudiation. In [47], Pournaghi et al. proposed a scheme to share medical data based on blockchain technology and attribute-based encryption (MedSBA), which utilizes blockchain to share medical data. Recently, Liu et al. [48] presented a blockchain-aided attribute-based searchable encryption scheme, and Zheng et al. [49] proposed a fair outsourcing decryption ABE scheme utilizing blockchain and sampling technology. In [50], Guo et al. used blockchain technology to propose an efficient and traceable ABE scheme with

dynamic access control, which implements dynamic access control and can flexibly update the access structure.

As of late, blockchain technology has been universally exploited in ABE scheme, because compared to the traditional server structure, blockchain has no central node, which allows no single institution or member to achieve control over global data, while any node stops working without affecting the overall operation of the system. In addition, blockchain is also superior to traditional key management center in ensuring the confidentiality of data. Therefore, we adopt blockchain technology to assure the security and robustness of the system.

3. Preliminaries

3.1. Composite Order Bilinear Groups and Complexity Assumptions. First of all, the concept of bilinear group is reviewed as follows. Let G and G_1 be the multiplicative groups of order $N = p_1p_2p_3$, g is a generator of G, and p_1, p_2, p_3 are three different prime. Then, $e : G \times G \longrightarrow G_1$ is a bilinear map, and it has these properties as follows:

(1) Bilinearity: For $\forall x, y \in Z_N$, $e(g^x, g^y) = e(g, g)^{xy}$

(2) Nondegeneracy: $e(g, g) \neq 1_{G_1}$

(3) Computability: There is an algorithm to calculate e efficiently

Now, we show the definition of the composite order bilinear groups. It is similar to bilinear groups except the order of the group is the product of two or more distinct prime numbers. That is to say, G is a composite order group; $G_{p_1}, G_{p_2}, G_{p_3}$ are its three subgroups of order p_1, p_2, p_3. For $\forall x \in G_{p_i}$ and $\forall y \in G_{p_j}$, if $x \neq y$, then $e(x, y) = 1$.

Decisional linear assumption. Given $(g_1^{\alpha}, g_2^{\beta}, g_1, g_2, g_3)$, $\alpha, \beta \in Z_N$, any probabilistic polynomial time (PPT) algorithm is difficult to distinguish $(g_1^{\alpha}, g_2^{\beta}, g_3^{\alpha+\beta}, g_1, g_2, g_3)$ from $(g_1^{\alpha}, g_2^{\beta}, g_1, g_2, g_3, A)$, where $g_1, g_2, g_3, A \in G$ and $a, b \in Z_p$ are randomly selected.

Decisional bilinear Diffie–Hellman problem (DBDH). Given $g^{\alpha}, g^{\beta}, g^{\gamma} \in G$ and $P \in G_1$, any PPT algorithm is difficult to distinguish $P = e(g, g)^{\alpha\beta\gamma}$ from $P = e(g, g)^{p}$, where $\alpha, \beta, \gamma, p \in Z_p^*$.

3.2. Access Structure and Linear Secret Sharing Scheme (LSSS)

Definition 1. Assume $O = \{attr_1, \cdots, attr_n\}$ is a set of attributes, and $\mathscr{A} \subset 2^O$ is a nonempty subset of 2^O, where 2^O represents the set constituted by all subsets of O; that is, $\mathscr{A}$ is a nonempty set constituted by some subsets of O. We call $\mathscr{A}$ is an access structure on O. If for any P, Q satisfies the condition $P \in \mathscr{A}$ and $P \subseteq Q$, namely $Q \in \mathscr{A}$, and then set $\mathscr{A} \subseteq O$ is monotonic. Authorized set refers to the set in $\mathscr{A}$; on the contrary, the unauthorized set is not in $\mathscr{A}$.

Definition 2. In the linear secret sharing matrix formed by the access policy, each row corresponds to an attribute value, that is, row vector and attribute value form a one-to-one mapping relationship. If the following two properties are satisfied, and then, a secret sharing scheme Σ on a set of $O = \{attr_1, \cdots, attr_n\}$ is called linear.

(1) The shared secret key for each attribute is a vector formed on Z_p

(2) In scheme Σ, there is an $n \times m$ secret sharing matrix $\boldsymbol{A}$, whose row label is $b(i)$, $i \in \{1, 2, \cdots, n\}$. Given a secret sharing column vector $u = (\mu, u_2, \cdots, u_m)$, where $\mu \in Z_p$ is the secret key to be shared, $u_2, \cdots, u_m$ is selected at random, $\mathbf{A}u$ represents the vector of n shared secret keys according to Σ. Shared $\gamma_i = (\mathbf{A}u)_i$, that is the inner product $\mathbf{A}u$ belongs to the property $b(i)$, where b is a function that maps $i \in \{1, 2, \cdots, n\}$ to $b(i)$

The LSSS matrix has an important feature, that is, linear reconstruction. Suppose Σ is a LSSS scheme representing access structure $\mathscr{A}$, $Q \in \mathscr{A}$ is an authorized set, and then, we can define $T \subset [n]$ as $T = \{i : b(i) \in Q\}$. If there has constant $\{\beta_i \in Z_p\}_{i \in T}$ that can be discovered in polynomial time such that $\{\gamma_i\}$ are valid shares of the secret key μ, then $\sum_{i \in T} \beta_i \gamma_i = \mu$. There is no such constant for any unauthorized set.

3.3. Binary Tree. First, there is a brief review of the definition of binary tree. Suppose UT represents a binary tree, in which there are L leaves corresponding to L users, the root node of BT is Rt. $path(\varphi)$ represents the set of all nodes on the path of leaf node φ from the root to φ. Note that φ and the root node are included here. φ_l, φ_r represents the left and right children of nonleaf nodes. The algorithm KUNodes is used to calculate the minimum set of nodes that need to release key updates, and only unrevoked users can decrypt the ciphertext within a period of time. That is, nodes in rl corresponding to time periods before or t do not have any ancestors in the set, and all other leaf nodes have exactly one ancestor in the set. Figure 1 gives the working principle of the KUNodes algorithm, where it first marks all ancestors of the revoked nodes as revoked and then outputs all unrevoked children of the revoked nodes. The Algorithm 1 is the formal definition of the KUNodes algorithm.

3.4. Shamir Secret Sharing [51]. The Shamir secret sharing scheme is based on Lagrange interpolation polynomials, which are described as follows:

(1) A trusted dealer D first randomly chooses a polynomial $g(x) = a_0 + a_1x + \cdots + a_{l-1}x^{l-1}$ with order l-1 such that $a_0 = s$, where $a_1, a_2, \cdots, a_{l-1}$ are in finite filed $F_p = GF(p)$. Then, D computes $s_1 = g(1), \ldots, s_m = g(m)$ and sends s_i to each shareholder p_i secretly

(2) More than l shareholder $p_r \subset p$ $(|p_r| \leq l)$ work together can reconstruct the secret using the Lagrange interpolating formula

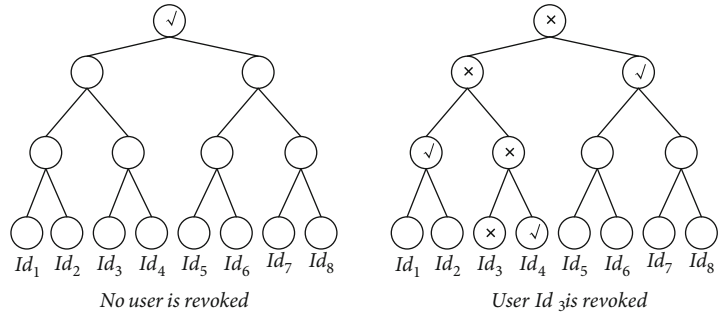


Figure 1: The KUNodes algorithm. This shows a pictorial depiction of the algorithm.

Inputs: a binary tree UT, a revocation list rl, a time period t, two empty sets P, Q.
$\forall(\varphi_i, t_i) \in rl$, if $t_i \le t$, then add $path(\varphi_i)$ to P.
$\forall p \in P$, if $p_l \notin P$, then add p_l to Q.
if $p_r \notin P$, add p_r to Q.
If $Q = \varnothing$, then add Rt to Q.
Return Q.

Algorithm 1: KUNodes.

3.5. Pedersen (l, m) Secret Sharing [17]. The Pedersen secret sharing scheme allows each dealer (shareholder) to randomly select a secret as a subsecret and can share the subsecret with other shareholders. Therefore, each shareholder can merger all the subsecrets as the master secret. The Pedersen secret sharing scheme is described as follows:

(1) Each shareholder $H_i (i \in [1, \cdots, m])$ randomly independently picks a subsecret S_i, and then, the master secret can be described as $S = \sum_{i=1}^{m} S_i$

(2) For each subsecret S_i, a l-1 polynomial $g(x)$ is randomly selected by shareholder H_i such that $S_i = g_i(0)$. After that, it calculates $s_{ij} = g_i(x_j)$,$(j = 1, 2, \cdots, m)$ for other shareholders by using Shamir's secret sharing. Finally, H_i sends each s_{ij} to other H_j secretly, and each shareholder H_i has m subshares s_{ij}

(3) Each shareholder H_i calculates its own master share $s_i = \sum_{j=1}^{m} s_{ji} = \sum_{j=1}^{m} g_j(x_i)$

(4) More than l shareholders $p_r \subset p$ $(|p_r| \le l)$ work together can reconstruct the secret by using the Lagrange interpolating formula

3.6. Reciprocal Protocol [18]. Suppose that shareholders H_i $(i \in [1, \cdots, m])$ share a secret μ using Pedersen (l, m) secret sharing protocol. The role of the reciprocal protocol is to get share μ^{-1} without disclosing relevant message about μ and μ^{-1}. The description of this protocol is as follows:

(1) Shareholders jointly run the Pedersen (l, m) secret sharing scheme to generate a (l, m) sharing of a random element $\alpha \in Z_q$. Denote all shares $\alpha_1, \alpha_2, \cdots, \alpha_m$ as $(\alpha_1, \alpha_2, \cdots, \alpha_m) \leftrightarrow^{(l,m)} \alpha$

(2) Shareholders jointly run the Pedersen (2l, m) secret sharing scheme to generate and retain a share of zero value β_i

(3) Shareholders need to pass the value $\mu_i\alpha_i + \beta_i$ and interpolating the corresponding $2l$ degree polynomial to reconstruct the value $\eta = \mu\alpha$

(4) Each shareholders sets $\delta_i = \eta^{-1}\alpha_i$ to calculate it share δ_i of μ^{-1}

3.7. Blockchain. On January 3, 2009, Satoshi Nakamoto generated the first Bitcoin block. A few days later, a second bitcoin block appeared to connect with the first block to form the chain, marking the birth of the blockchain. Because of its four main features, immutability, irreproducible uniqueness, smart contracts, and decentralized self-organization or community, blockchain is widely used in various fields.

In simple terms, a hash function (SHA-256) is used to form a blockchain. Each block contains a parent block hash, a timestamp, and a Merkle root. Where the parent block hash stores the hash value of the previous block header and is used to connect the previous block, the timestamp records the approximate time when the block was created, and the Merkle root is the Merkle tree root hash generated by the transaction list. There are three general types of blockchains: public blockchains, consortium blockchains, and private blockchains. Our system uses a consortium blockchain. Figure 2 illustrates the basic structure of a consortium blockchain.

In our consortium blockchain, consensus node nodes perform the consistency protocol to renovate the blockchain and reserve all nodes in the system with a consistent state. The consortium blockchain used in our system is similar to the scheme [48] in that firstly the consensus nodes can

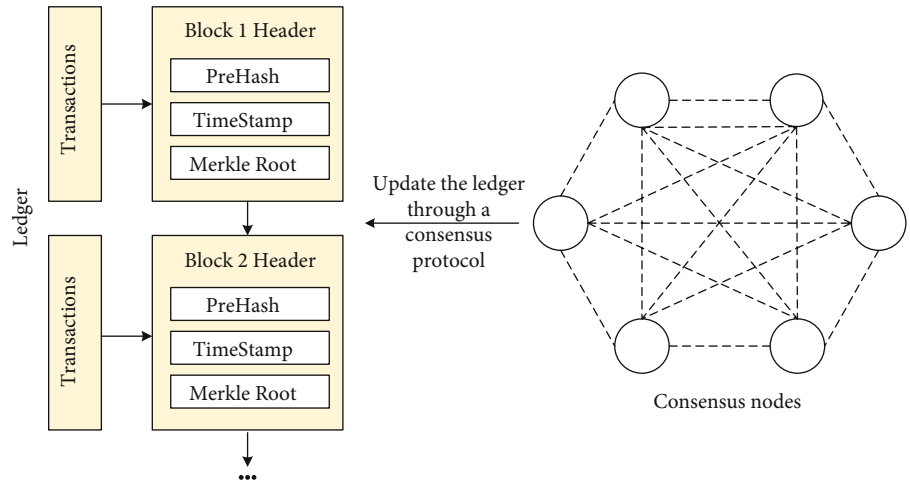


Figure 2: Consortium blockchain. Consensus node nodes perform the consistency protocol to renovate the blockchain.

initialize the system parameters using the Pedersen secret sharing scheme and the reciprocity protocol. Secondly, the consensus nodes manage the associated keys of the users. Updating the user revocation list requires the joint participation of all consensus nodes, which effectively improves the system security. Users can submit searches to the blockchain, and the cloud server is able to perform predecryption operations for the users.

4. System Definition

4.1. System Model. Our data management system includes the following four participants:

Data owner (DO): The data owner stores the generated index and encrypted data in the cloud, where the index is used for the cloud to perform search operations.

Data user (DU): The data user is able to store the generated partial token in the consortium blockchain and is able to get the predecrypted message from the cloud to fully decrypt it using their private key.

Blockchain (BC): The consortium blockchain in the system consists of a set of credible predefined consensus nodes, a data pool, and a distributed ledger. The blockchain is responsible for initializing the system, storing the users' public identity keys, and generating the users' public decryption keys, key update information, and predecryption keys. In addition, the blockchain is also responsible for generating the complete token and sending it to cloud.

Cloud server (CS): Cloud server can search and predecrypt for users, and putting predecryption operations in the cloud can effectively reduce the burden of users.

Figure 3 shows the system procedure.

4.2. System Procedure. Based on [48], the basic process of the scheme is defined as follows:

4.2.1. System Init. All consensus nodes run $Setup(1^k, \sigma) \longrightarrow GPK$ algorithm to get the GPK. Pedersen (l, m) secret sharing protocol and reciprocal protocol are used by all consensus nodes to jointly determine the master key, and the exact value of the master parameter is unknown.

4.2.2. User Registration and Revocation

(1) The data user runs the $IdKG(GPK, Id_U) \longrightarrow (sk_{Id_U}, pk_{Id_U})$ to get its identity key pair to join the system and sends pk_{Id_U} to BC. Then, the user public decryption key is generated by the consensus nodes using pk_{Id_U}. In the meantime, the user's predecryption key is also generated later using the public decryption key

(2) For user revocation, based on a time period t, a state mark sm, and the revocation list rl, consensus node runs the $Rv(Id_U, t, rl, sm) \longrightarrow rl$ to update rl whenever a user wishes to be revoked

4.2.3. Key Gen. In this step, consensus nodes generate three keys:

(1) Public decryption key generation: In this step, consensus nodes use the user identity Id_U, an access structure $(\mathscr{D}, b)$ to run $PubDecKG(GPK, Id_U, (\mathscr{D}, b), sm) \longrightarrow (PubDK_{Id_U}, sm)$ algorithm to generate the user public decryption key, which is used to verify whether the user has the attributes included in its attribute set

(2) Updated key generation: Consensus nodes run the $UpKG(GPK, t, rl, sm) \longrightarrow (U_t, sm)$ to get a key update message after rl is updated; it inputs a new time period t and a state mark sm, users who have not revoked will use it when generating a predecryption key

(3) Predecryption key generation: Consensus nodes use (U_t, sm) to run $PreDecKG(GPK, Id_U, (\mathscr{D}, b), pk_{Id_U},$

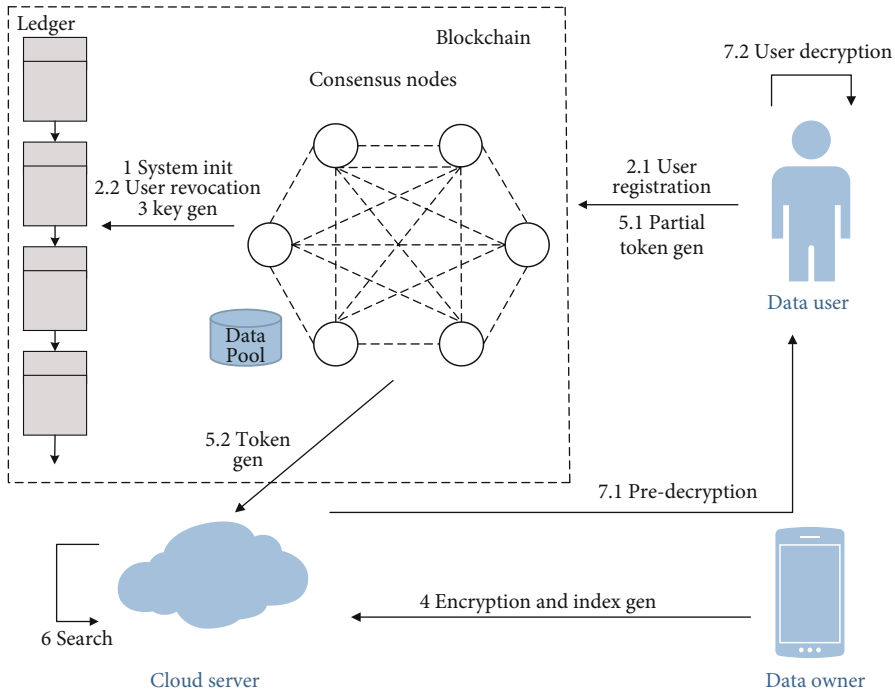


Figure 3: System procedures. The data management system contains four participants, and each participant operates as shown in the figure.

$U_t) \longrightarrow \mathrm{Pre}DK_{Id_U,t}$); it generates the predecryption key for user

4.2.4. Encryption. First, the data owner selects several keywords related to his/her data to generate a keyword set $\{w\}$, and then, he/she uses a symmetric algorithm with the key k_s to encrypt the data. Then, data owner runs the $Enc(GPK, \mathbf{S}, t, k_s) \longrightarrow CT$ to hide the symmetric encryption key and runs the $IdxG(GPK, \mathbf{S}, \{w\}) \longrightarrow Idx$ to generate an index set, where the same attribute set $\mathbf{S}$ is used. Finally, data owner sends CT and Idx to cloud.

4.2.5. Token Gen

(1) Patrial token generation: first, the data user runs the $PTokG(GPK, sk_{Id_U}, w) \longrightarrow \mathrm{Tok}'$ to generate the partial token. Then, data user sends $(\mathrm{Tok}', H(w))$ to BC

(2) Complete token generation: consensus nodes run $TokG(GPK, Id_U, (\mathscr{D}, b), \mathrm{Tok}') \longrightarrow Tok$ algorithm for data user after receiving Tok. Then, data user sends Tok to the cloud

4.2.6. Search. When data users need to search, the cloud runs the $Search(GPK, \mathbf{S}, Tok, Idx) \longrightarrow Addr/\perp$ algorithm to search.

4.2.7. Decryption

(1) Predecryption: The predecryption operation is done in the cloud, and the cloud takes as input the predecryption key sent to it by the blockchain, and it runs the $PreD(GPK, (\mathscr{D}, b), Id_U, \mathrm{Pre}DK_{Id_U,t}, CT, t) \longrightarrow CT'/\perp$ to convert the ciphertext of data users after receiving the requested key. In this phase, most of the calculation costs will be carried out in the cloud

(2) User decryption: Since a portion of the predecryption work has already been done in the cloud, data user runs the $Dec(GPK, sk_{Id_U}, CT') \longrightarrow k_s/\perp$ to generate the symmetric decryption key. After that, the complete decryption is done by running the symmetric algorithm

4.3. Security Model

4.3.1. Ciphertext Indistinguishability. Based on [48], we give a security definition of indistinguishability under chosen plaintext attacks (IND-CPA) for BC-SABE, defined by an indistinguishable game between adversary A and challenge B. Let U_m be the authority universe. Adversary A is defined as an (l,m) adversary that can compromise at most l-1 authority; Pedersen (l, m) secret sharing protocol and reciprocity protocol are applied in this security model.

Setup. A corrupted authority set U_ϕ is output by A, where $m_\phi \le l-1$. Then, B runs the $Setup(1^k, \sigma) \longrightarrow GPK$ to get the global public key GPK, a state mark sm and a revocation list rl. Note that rl is initially empty. It outputs $(GPK, rl, sm, \{a_i, r_i\}_{i \in U_\phi})$ to A.

Phase 1. First, B creates an empty list L, and A can perform the following queries adaptively:

(i) IdKey query: First of all, A issues a IdKey query on an identity Id_U, B returns (sk_{Id_U}, pk_{Id_U}) by running $IdKG(GPK, Id_U)$ algorithm, and then, it adds $(Id_U, sk_{Id_U}, pk_{Id_U})$ to list L. It returns sk_{Id_U} to A

(ii) PubDKey query: A issues a PubDKey Query on an identity Id_U and an access structure $(\mathscr{D}, b)$. If Id_U has not been issued to PubDKey query, then B returns (sk_{Id_U}, pk_{Id_U}) by running $IdKG(GPK, Id_U)$, runs $PubDecKG(GPK, Id_U, (\mathscr{D}, b), sm) \longrightarrow (PubDK_{Id_U}, sm)$ algorithm, and returns $(Id_U, PubDK_{Id_U}, sm)$ to A

(iii) Upkey query: Algorithm A submits a Upkey query on a time period t. B returns the U_t to A by running $UpKG(GPK, t, rl, sm)$

(iv) PreDKey query: Algorithm A submits a PreDKey query on $(Id_U, (\mathscr{D}, b), t)$. If Id_U has not been issued to PreDKey query, B runs the $IdKG(GPK, Id_U)$,$UpKG(GPK, t, rl, sm)$, $PubDecKG(GPK, Id_U, (\mathscr{D}, b), sm)$, and $PreDecKG(GPK, Id_U, (\mathscr{D}, b), pk_{Id_U}, U_t)$. Then, B returns the $PreDK_{Id_U,t}$ to A. Note that this oracle cannot be queried on a time period t before a Upkey query has been queried on t

(v) Revocation query: A submits a PreDKey Query on (Id_U, t), where t is not queried in UpKey query. B returns an updated revocation list rl to A by running $Rv(Id_U, t, rl, st)$

Challenge. A hands over two symmetric keys with same length SK_1^* and SK_2^*, an attribute set $\mathbf{S}^*$, and a time period t^* satisfying the following restrictions:

(i) If an identity Id_U^* has performed to the IdKey query, $\mathbf{S}^*$ of Id_U^* satisfies a query on $(Id_U^*, (\mathscr{D}^*, b^*))$ issued to the PubDKey query. Then, the revocation query must be queried on (Id_U^*, t^*) with $t = t^*$ or any t occurs before t^*, and the PreDKey query cannot be queried on (Id_U^*, t^*)

(ii) If Id_U^* with access structure $(\mathscr{D}^*, b^*)$ can be satisfied by $\mathbf{S}^*$ is not revoked before or at t^*, then Id_U^* has never been queried by the IdKey query

B picks random $\rho \in \{0, 1\}$ and runs $Enc(GPK, \mathbf{S}^*, t^*, SK_\rho^*) \longrightarrow CT^*$ to encrypt SK_ρ^*, and then, it returns CT^* to A.

Phase 2. A can adaptively perform the same five queries to B as in phase 1; the queries sent by A must also meet the above conditional restrictions.

Guess. A makes a guess ρ' for ρ; if $\rho' = \rho$, it wins.

The advantage of the adversary A in this game is described as $\Pr[\rho = \rho'] - 1/2$.

If the advantages of any (l, m) PPT adversary defined above are negligible, then a BC-SABE scheme is IND-CPA secure.

4.3.2. Index Indistinguishability. Based on [48], we give a security definition of indistinguishability under selective access structure and chosen keyword attacks (IND-sCKA) for BC-SABE, defined by an indistinguishable game between adversary A and challenge B.

Init. Let $\mathbf{S}^*$ be the challenge access structure defined by A.

Setup. B returns the global public parameter GPK to A by running $Setup(1^k, \sigma)$ algorithm.

Phase 1. A can adaptively submit the following query:

(i) IdKey query: A issues a IdKey query on an identity Id_U, B returns (sk_{Id_U}, pk_{Id_U}) by running $IdKG(GPK, Id_U)$, and then, it adds $(Id_U, sk_{Id_U}, pk_{Id_U})$ to list L and returns sk_{Id_U} to A

(ii) Token query: A issues a token query on Id_U, an access structure $\mathbf{S}$, and a keyword w. B runs $PTokG(GPK, sk_{Id_U}, w) \longrightarrow \mathrm{To}k'$ by using sk_{Id_U}. Then, B returns Tok to A by running the $TokG(GPK, Id_U, (\mathscr{D}, \rho), \mathrm{To}k')$ algorithm. Note that the challenge access structure does not contain the attribute set $\mathbf{S}$

(iii) Index query: A issues an index query on $\mathbf{S}$ and a keyword set $\{w\}$. Then, B returns an index set by running $IdxG(GPK, \mathbf{S}, \{w\})$

Challenge. A submits two keywords K_1^* and K_2^* of the same size. B picks random $\rho \in \{0, 1\}$, and then it returns the Ind^* to A by running the $IdxG(GPK, \mathbf{S}^*, w_\rho)$ algorithm with the challenge access structure $\mathbf{S}^*$.

Phase 2. A can perform IdKey query, token query, and index query to B; the queries sent by A must also meet the above conditional restrictions.

Guess. A makes a guess ρ' for ρ; if $\rho' = \rho$, it wins.

The advantage of the adversary A in this game is described as $\Pr[\rho = \rho'] - 1/2$.

If the advantages of any PPT adversary defined above are negligible, then a BC-SABE scheme is IND-sCKA secure.

5. Construction

5.1. System Init. $Cu = (Cu_1, Cu_2, \cdots, Cu_m)$ is a consensus node set which has m consensus nodes. First, consensus nodes exploit the Pedersen (l, m) secret sharing protocol [17] and the reciprocal protocol [18] to run the $Setup(1^k, \sigma) \longrightarrow GPK$ to generate the global public key GPK, the user revocation list rl, and the user tree UT, where $\sigma \in \{0, 1\}^{poly(k)}$ is a randomly public string. Let G be groups of a prime order p, g is a generator of G, and $e : G \times G \longrightarrow G_1$ is a bilinear map. Then, it chooses random $u_0, \cdots, u_\eta, h_0, \cdots, h_\tau \in G$. Let $P(x) = \prod_{j=0}^{\eta} u_j^{x_j}$ and $F(x) = \prod_{j=0}^{\tau} h_j^{x_j}$, $H : \{0, 1\}^* \longrightarrow Z_p^*$ be the collision-resisted hash function, and $D = (e, G, G_1, p, g, H)$ be the admissible bilinear group parameters. Then, Cu_i share two secret parameters $a, r \in G$ and using Pedersen (l, m) secret sharing scheme and reciprocal protocol to compute shares of r^{-1}. Based on its shares a_i, r_i, and j_i, each consensus node Cu_i computes and broadcasts g^{a_i}, g^{r_i}, and g^{j_i}; excess l consensus nodes cooperate to recreate $g^{r^{-1}} = \prod_{i=1}^{l} (g^{j_i})^{L(i)} = g^{\sum_{i=1}^{l} L(i) \cdot j_i}$; and each nodes spreads $g^{a_i/r}$. The public parameters of the system are also generated in this way: $g^r = \prod_{i=1}^{l} (g^{r_i})^{L(i)} = g^{\sum_{i=1}^{l} L(i) \cdot r_i}$, $e(g, g)^a = \prod_{i=1}^{l} (e(g, g)^{a})^{L(i)} = e(g, g)^{\sum_{i=1}^{l} L(i) \cdot a_i}$, $g^{\alpha/\gamma} = \prod_{i=1}^{l} (g^{a_i/r})^{L(i)} = g^{\sum_{i=1}^{l} L(i) \cdot a_i/r}$.Finally, it outputs global public key:

$$GPK = \left(D, P(x), F(x), u_0, \cdots, u_\eta, h_0, \cdots, h_\tau, e(g,g)^a, g^r, g^{a/r}\right). \tag{1}$$

5.2. User Registration and Revocation

5.2.1. User Registration. The IdKG algorithm inputs the *GPK* , the user identity Id_U, and UT; it picks random $g_3 \in G$ and $b_1, b_2 \in Z_p$; and then, it calculates $g_1 = g_3^{1/b_1}$, $g_2 = g_3^{1/b_2}$. The user runs this algorithm to generate the user public and private key pair $(pk_{Id_U}, sk_{Id_U}) = ((g_1, g_2, g_2), (b_1, b_2))$, and then the user sends pk_{Id_U} to BC. An undefined leaf node φ is selected by BC from the UT to storage Id_U and pk_{Id_U}.

5.2.2. User Revocation. The revoke algorithm is run by the consensus node to update the user revocation list *rl* whenever user want to be revoked. It inputs the user identity Id_U, a state mark *st*, a time period *t*, and *rl*; then, it will find all nodes y associated with the identity Id_U and put (y, t) into the *rl* list and outputs the revised *rl*.

5.3. Key Generation

5.3.1. Public Decryption Key Gen. Consensus nodes run the PubDecKG algorithm to generate the public decryption key $PubDK_{Id_U}$; it takes *GPK*, user identity Id_U, and a state mark *sm* and an access structure (D, b) as the input, where matrix D is an $n_D \times m_D$ with row label $b(i)$, $i \in \{1, 2, \cdots, l\}$. Then, it selects random the leaf node φ from the UT based on Id_U, and compute $path(\varphi)$. It selects random $v_{y,2}, \cdots, v_{y,m_D} \in Z_p$ and let $\vec{v} = (a, v_{y,2}, \cdots, v_{y,m_D})$, and then, it computes $v_{y,i} = D_i \cdot \vec{v}_y$ for $i \in [n_D]$, where D_i denotes the vector of the i-th row of matrix D. For each $y \in path(\varphi)$, it takes g_y and computes $g'_{y,i} = g^{v_{y,i}}/g_y$ and stores g_y in the node y. Then, share γ_1, γ_2, $\{\gamma_{y,j}\}_{j \in Id_U}$ among consensus nodes by exploiting the Pedersen (l, m) secret sharing scheme, and each consensus nodes computes and spreads $\{g^{\gamma_{y,j,i}}, F(b(i))^{\gamma_{y,j,i}}\}_{j \in Id_U}$ based on share $\{\gamma_{y,j,i}\}_{j \in Id_U}$. Then, excess l consensus nodes cooperate to compute the public decryption key as follows:

$$\begin{aligned} D_{y,1,j} &= g_3^{\gamma_1+\gamma_2} \cdot g'_{y,i} \cdot \prod_{i=1}^{l} \left(F(b(i))^{\gamma_{y,j,i}}\right)^{L(i)} \\ &= g_3^{\gamma_1+\gamma_2} \cdot g^{v_{y,j}}/g_y \cdot \prod_{i=1}^{l} \left(F(b(i))^{\gamma_{y,j,i}}\right)^{L(i)} \\ &= g_3^{\gamma_1+\gamma_2} \cdot g^{v_{y,j}}/g_y \cdot F(b(i))^{\sum_{i=1}^{l} \gamma_{y,j,i} \cdot L(i)}, \\ &= g_3^{\gamma_1+\gamma_2} \cdot g^{v_{y,j}}/g_y \cdot F(b(i))^{\gamma_{y,j}} \end{aligned} \tag{2}$$

$D_{y,2,j} = \prod_{i=1}^{l} (g^{\gamma_{y,j,i}})^{L(i)} = g^{\sum_{i=1}^{l} \gamma_{y,j,i} \cdot L(i)} = g^{r_{g,i}}$, $D_3 = g_1^{\gamma_1}$, $D_4 = g_2^{\gamma_2}$. Finally, BC stores PDK_{Id} in the ledger:

$$PDK_{Id} = \left\{ \left\{ y, \{D_{y,1,j}, D_{y,2,j}\}_{i \in [l]} \right\}_{y \in path(\varphi)}, D_3, D_4 \right\}. \tag{3}$$

5.3.2. Key Update Messages Gen. Consensus nodes run the UpKG algorithm to generate the update information U_t; it inputs the *GPK*, the revocation list *rl*, a time period *t*, the user tree UT, and a state mark *sm*. Share s_y among consensus nodes by exploiting the Pedersen (l, m) secret sharing scheme, and each consensus nodes computes and spreads $g^{s_{y,i}}, P(t)^{s_{y,i}}$ based on share $s_{y,i}$.

For all $y \in KUNodes(UT, rl, t)$, it takes g_y from the node y. Then, more than l consensus nodes cooperate to compute the update information as follows: $U_{y,1} = g_y \cdot \prod_{i=1}^{l} (P(t)^{s_{y,i}})^{L(i)} = g_y \cdot P(t)^{\sum_{i=1}^{l} s_{y,i} \cdot L(i)} = g_y \cdot P(t)^{s_y}$, $U_{y,2} = \prod_{i=1}^{l} (g^{s_{y,i}})^{L(i)} = g^{\sum_{i=1}^{l} s_{y,i} \cdot L(i)} = g^{s_y}$. BC stores U_t in the ledger:

$$U_t = \left\{ \left\{ y, U_{y,1}, U_{y,2} \right\}_{y \in KUNodes(UT, rl, t)} \right\}. \tag{4}$$

5.3.3. Predecryption Key Gen. Consensus nodes run the PreDecKG algorithm to generate the update information $PreDK_t$; it inputs the *GPK*, a time period *t*, user identity Id_U, an access structure (D, b), the user tree UT, a state mark *sm*, the revocation list *rl*, user public decryption key $PubDK_{Id}$, and a update information U_t. Then, let $I = Path(\theta)$ and $J = KUNodes(UT, rl, t)$, so we have $U_t = (\{y, U_{y,1}, U_{y,2}\}_{y \in J})$ and $PDK_{Id} = (\{y, \{D_{y,1,j}, D_{y,2,j}\}_{i \in [l]}\}_{y \in path(\varphi)}, D_3, D_4)$. If $I \cap J = \varnothing$, it returns $\perp$. Then, share s_y and $\{\gamma_{y,j}\}_{j \in Id_U}$ among consensus nodes by exploiting the Pedersen (l, m) secret sharing protocol. Based on share $s'_{y,i}$ and $\{\gamma'_{y,j,i}\}_{j \in Id_U}$, each consensus nodes computes and spreads $\{g^{\gamma'_{y,j,i}}, F(b(i))^{\gamma'_{y,j,i}}\}_{j \in Id_U}$, $g^{s'_{y,i}}, P(t)^{s'_{y,i}}$. Then, more than l consensus nodes cooperate to compute the predecryption key.

$$\begin{aligned} Tk_{1,j} &= D_{y,1,j} \cdot U_{y,1} \cdot \prod_{i=1}^{l} \left(F(b(i))^{\gamma'_{y,j,i}} \cdot P(t)^{s_{y,i}}\right)^{L(i)} \\ &= g_3^{\gamma_1+\gamma_2} \cdot g^{v_{y,j}}/g_y \cdot g_y \cdot P(t)^{s_x} \cdot \prod_{i=1}^{l} \left(F(b(i))^{\gamma'_{y,j,i}} \cdot P(t)^{s'_{y,i}}\right)^{L(i)} \\ &= g_3^{\gamma_1+\gamma_2} \cdot g^{v_{y,j}} \cdot P(t)^{s_y} \cdot F(b(i))^{\sum_{i=1}^{l} \gamma'_{y,j,i} \cdot L(i)} \cdot P(t)^{\sum_{i=1}^{l} s'_{y,i} \cdot L(i)} \\ &= g_3^{\gamma_1+\gamma_2} \cdot g^{v_{y,j}} \cdot F(b(i))^{\gamma_{y,j}+\gamma'_{y,j}} \cdot P(t)^{s_y+s_y'}, \end{aligned} \tag{5}$$

$Tk_{2,j} = D_{y,2,j} \cdot \prod_{i=1}^{l} (g^{\gamma'_{y,i}})^{L(i)} = g^{\gamma_y+\gamma_y'}$,
$Tk_3 = U_{y,2} \cdot \prod_{i=1}^{l} (g^{s'_{y,i}})^{L(i)} = U_{y,2} \cdot g^{\sum_{i=1}^{l} s'_{y,i} \cdot L(i)} = g^{s_y+s_y'}$. BC stores $PreDK_{Id_U,t}$ in the ledger.

$$\text{Pre}DK_{Id_U,t} = \left\{ \left\{ Tk_{1,j}, Tk_{2,j} \right\}_{j\in[l]}, Tk_3 \right\}. \quad (6)$$

5.4. Encryption

5.4.1. Data Encryption. Date owner runs the Enc algorithm to get the ciphertext CT. It inputs the GPK, an attribute set $\mathbf{S}$, and a symmetric key k_s and a time period t. Let $\mathbf{S} = \{\mathbf{S}_1, \cdots, \mathbf{S}_l\}$ and choose $\mu \in Z_p$. It calculates $C_0 = e(g,g)^{a\mu} \cdot k_s$, $C_1 = g^{\mu}$, $C_{2,i} = F(\mathbf{S}_i)^{\mu}$, and $C_3 = P(t)^{\mu}$ and outputs CT_{sym} and $CT = (\mathbf{S}, t, C_0, C_1, \{C_{2,i}\}_{i\in[l]}, C_3)$.

5.4.2. Index Generation. Date owners run the IdxG algorithm to generate the Idx. It inputs the GPK, the same attribute set S, and a keyword set $\{w_l\}_{l\in U_w}$, and then, it computes $Idx_{1,l} = e(g,g)^{a\mu\cdot H(w_l)}$, $Idx_2 = g^{r\mu}$, $Idx_{3,i} = g^{rv_i}$, and $Idx_{4,i} = F(b(i))^{v_i}$. Finally, it sent index set Idx along with CT_{sym} and CT to the cloud.

$$Ind = \left\{ \left\{ Idx_{1,l} \right\}_{l\in U_w}, Idx_2, \left\{ Idx_{3,i}, Idx_{4,i} \right\}_{i\in[n_D]}, \mathbf{S}^* \right\}. \quad (7)$$

5.5. Token Gen

5.5.1. Partial Token Gen. In this phase, data users run the PTokG algorithm to get the partial token. It inputs the GPK, user private key sk_{Id_U}, and a keyword w. Then, it randomly selects $q \in Z_p$; it computes the partial token $\text{To}k' = (g^{a/r})^{(b_1+b_2+q)}$ and the hash value $H(w)$. Finally, it sends $\text{To}k'$ and $H(w)$ to BC.

5.5.2. Complete Token Gen. Blockchain runs the TokG algorithm to get the complete token. It inputs the GPK, the user identity Id_U, the same access structure (D, b), and the partial token $\text{To}k'$. Share $\{\alpha_j\}_{j\in Id_U}$ among consensus nodes by exploiting Pedersen (l, m) secret sharing protocol. Based on share $\{\alpha_{j,i}\}_{j\in Id_U}$, each consensus nodes computes and spreads $\{g^{r\cdot\alpha_{j,i}}, F(b(i))^{a_{j,i}}\}_{j\in Id_U}$. Then, more than l consensus nodes cooperate to compute the complete token as follows:

$$\begin{aligned} Tok_{1,j} &= \text{To}k' \cdot \prod_{i=1}^{l} \left(F(b(i))^{\alpha_{j,i}}\right)^{L(i)} \\ &= \left(g^{a/r}\right)^{(b_1+b_2+r)} \cdot F(b(i))^{\sum_{i=1}^{l} \alpha_{j,i}\cdot L(i)} \\ &= \left(g^{a/r}\right)^{(b_1+b_2+r)} \cdot F(b(i))^{\alpha_j}, \end{aligned} \quad (8)$$

$Tok_{2,j} = \prod_{i=1}^{l} \left(g^{r\cdot\alpha_{j,i}}\right)^{L(i)} = g^{r\cdot\sum_{i=1}^{l}\alpha_{j,i}\cdot L(i)} = g^{r\cdot\alpha_j}$, $Tok_3 = \text{To}k' \cdot \left(g^{a/r}\right)^{H(w)} = \left(g^{a/r}\right)^{(b_1+b_2+H(w))}$. Finally, it sends the token Tok to the cloud:

$$Tok = \left\{ \left\{ Tok_{1,j}, Tok_{2,j} \right\}_{j\in Id_U}, Tok_3 \right\}. \quad (9)$$

5.6. Search. Cloud runs search algorithms to perform search operations; it takes the GPK, the same access structure (D, b), the complete token Tok, and the Ind as inputs. It yields the capacity address $Addr$ of related ciphertext if and only if the algorithm runs successfully; otherwise, the algorithm stops. First, it verifies whether the user's attribute set satisfies the access structure, and if not, it outputs a stop character $\perp$. If it is satisfied, let $T = \{i \in [n_D] \mid b(i) \in \mathbf{S}\}$, and then the algorithm can calculate a set $\{d_i \in Z_p\}_{i\in T}$ which makes $\sum_{i\in T} d_i D_i = (1, 0, \cdots, 0)^{m_D}$. Then, it verifies whether the equation described is valid.

$$Idx_{1,l} = \frac{e(Idx_2, Tok_3)}{\prod_{i\in T}\left(e\left(Tok_{1,b(i)}, Idx_{3,i}\right)/e\left(Tok_{3,b(i)}, Idx_{4,i}\right)\right)^{d_i}}. \quad (10)$$

If the formula holds, the stored address $Addr$ is the output.

5.7. Decryption

5.7.1. Predecryption. The predecryption operation is performed by the cloud, which runs the PreD algorithm to complete. It inputs the GPK, user identity Id_U, the predecryption key $\text{Pre}DK_{Id_U,t}$, a time period t, the same set $T = \{i \in [n_D] \mid b(i) \in \mathbf{S}\}$, and the ciphertext CT. If $\{v_i\}$ are valid shares of any secret μ from D, then the algorithm can compute a set $\{d_i \in Z_p\}_{i\in T}$ which makes $\sum_{i\in T} d_i v_i = \mu$. It computes

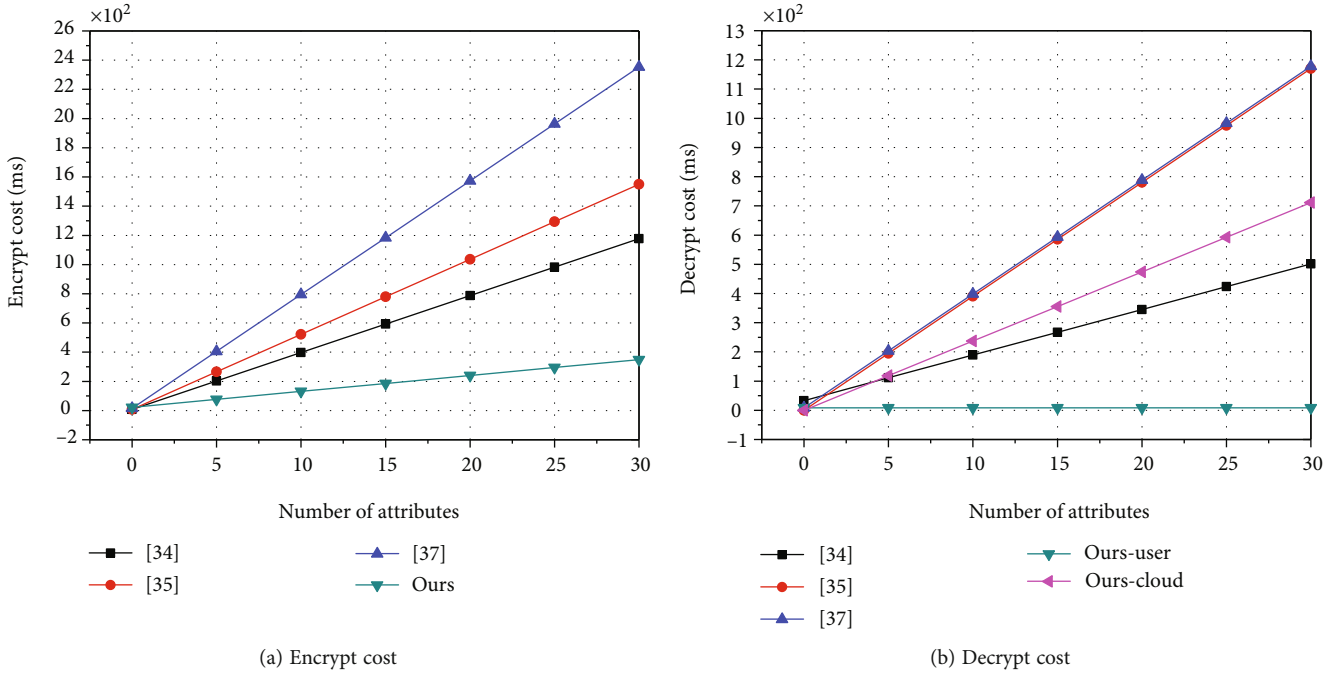
$$\begin{aligned} CT' &= \prod_{i\in T}\left(\frac{e\left(C_{2,i}, Tk_{2,b(i)}\right)e(C_3, Tk_3)}{e\left(Tk_{1,b(i)}, C_1\right)}\right)^{d_i} \\ &= \frac{1}{e\left(g_3^{\gamma_1+\gamma_2}, C_1\right)e(g, C_1)^a}. \end{aligned} \quad (11)$$

5.7.2. Decryption. The Dec algorithm inputs the GPK, private user key sk_{Id_U}, the predecryption ciphertext CT', and the symmetric decryption algorithm to generate the symmetric decryption key. This algorithm is run by the DU, and it can finish full decryption. The decryption is as follows. Using k_s to run symmetric algorithms can enable users to get plaintext, and users will not consume a lot of costs because the previous operations are already performed in the cloud.

$$\begin{aligned} k_s &= CT' \cdot e\left(D_3^{b_1} D_4^{b_2}, C_1\right) \cdot C_0 \\ &= \frac{e\left(g_1^{b_1\gamma_1} g_2^{b_2\gamma_2}, g^{\mu}\right) \cdot e(g,g)^{a\mu} \cdot k_s}{e\left(g_3^{\gamma_1+\gamma_2}, g^{\mu}\right)e(g, g^{\mu})^a} \\ &= \frac{e\left(g_3^{\gamma_1+\gamma_2}, g^{\mu}\right) \cdot e(g,g)^{a\mu} \cdot k_s}{e\left(g_3^{\gamma_1+\gamma_2}, g^{\mu}\right)e(g, g^{\mu})^a}. \end{aligned} \quad (12)$$

Table 1: Performance comparison. Five aspects were compared with the other four schemes.

Schemes	Encrypt cost	Index gen	Tok gen	Search	Decrypt cost
[34]	$Ep_T + (5N_e + 1)Ep$	No	No	No	$2N_T Pa + 3(EP_T + Ep)$
[35]	$(N_e + 1)Pa + 4N_e Ep$	No	No	No	$5N_T Pa + Ep_T$
[36]	$SymEnc$	$(2N_e + N_w + 3)Ep$	$(2N_t + 3)Ep$	$Ep_T + (2N_s + 4)Pa$	$SymDec$
[37]	$(3N_e + 2)Pa + 5N_e Ep$	No	No	No	$(5N_T + 1)Pa + 2Ep_T$
Ours	$Sym + (N_e + 2)Ep + Ep_T$	$(2N_e + 1)Ep + N_w Ep_T$	User:Ep BC:$2N_s Ep + Ep_T$	$N_T Ep_T + (2N_T + 1)Pa$	User:$2Ep_T + Sym + Pa$ Cloud:$2N_T Ep_T + (3N_T)Pa$.



(a) Encrypt cost

(b) Decrypt cost

Figure 4: Enc/Dec time cost comparison. It can be seen that our scheme is not very expensive in terms of encryption and decryption.

6. Security Proof

Theorem 4. *If Pedersen (l, m) secret sharing protocol is IND-CPA secure and the SR-ABE scheme presented by Cui et al. [31] is IND-CPA secure, then our BC-SABE scheme is IND-CPA secure.*

Proof: In contrast to the scheme in [31], distributed consortium blockchain is used in our scheme, replacing the centralized server in [31]. This allows the security of the whole system to be improved. Supposing that the adversary in our scheme can compromise at most l-1 authority, the reasons are as follows: Our PubDKey Oracle, the UpKey Oracle, and the PreDKey Oracle have excellent performance because it needs more than l authorities that are required to execute together. In addition, during the challenge phase, we defined related restrictions. Therefore, the proof of this scheme can be deduced from the security proof of the scheme [31] under the security of Pedersen (l, m) secret sharing protocol, and reciprocity protocol.

Theorem 5. *Under the DBDH assumption, The BC-SABE scheme is IND-sCKA secure.*

Proof: Assuming that there is a PPT adversary A who can win the exponential indistinguishability game with an advantage ε that cannot be ignored, then the challenge B is constructed to resolve the DBDH problem with an advantage $\varepsilon/2$ that cannot be ignored.

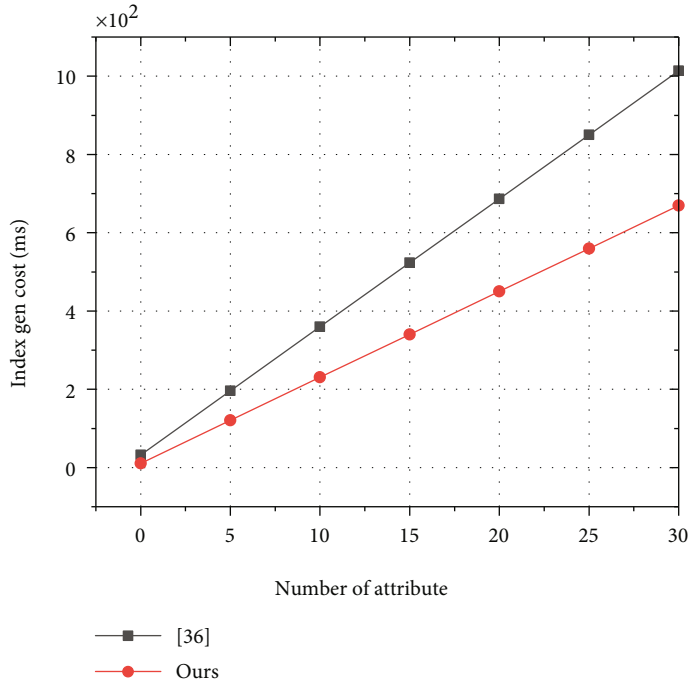
Init. Let $\mathbf{S}^*$ be the challenge access structure defined by A.

Setup. B returns the GPK to A by running the $Setup(1^k, \sigma)$ algorithm, the difference in GPK is $e(g, g)^a = e(g^\beta, g^\gamma) = e(g, g)^{\beta\gamma}$ and $r \in Z_p^*$, and other parameters are ignored here.

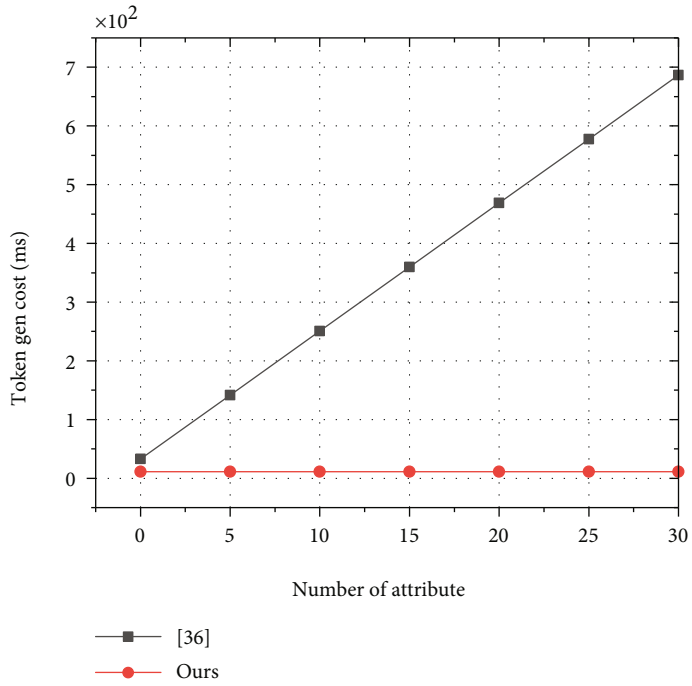
Phase 1. A can adaptively execute the following queries:

(i) IdKey query: A issues a IdKey query on an identity Id_U, B returns (sk_{Id_U}, pk_{Id_U}) by running $IdKG(GPK, Id_U)$, and then, it adds $(Id_U, sk_{Id_U}, pk_{Id_U})$ to list L and returns sk_{Id_U} to A

(ii) Token query: A issues a token query on Id_U, an access structure $\mathbf{S}$, and a keyword w. B runs $PTokG(GPK, sk_{Id_U}, w) \longrightarrow \mathrm{Tok}'$ by using sk_{Id_U}. Then, B

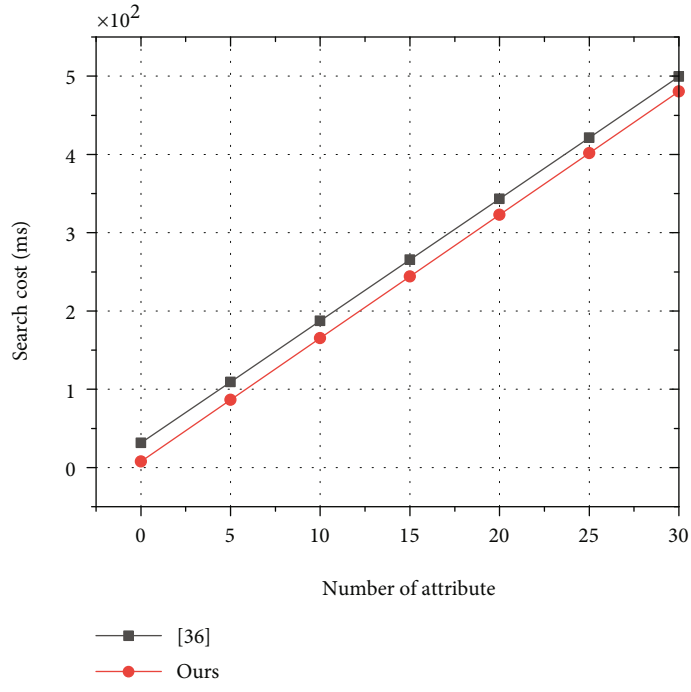


(a) Index gen cost



(b) Tok gen cost

Figure 5: Continued.



(c) Search cost

FIGURE 5: Other time cost comparison. It can be seen that our scheme is better in terms of index generation, token generation, and search compared to scheme [36].

returns Tok to A by running the $TokG(GPK, Id_U, (\mathscr{D}, \rho), \mathrm{Tok}')$ algorithm. The attribute set $\mathbf{S}$ does not meet challenge access structure

(iii) Index query: A issues an index query on a keyword set $\{w\}$ and $\mathbf{S}$. Then, B returns the index set by running $IdxG(GPK, \mathbf{S}, \{w\})$

Challenge. A submits two keywords w_1 and w_2 of the same size. B picks random $\rho \in \{0, 1\}$, and then, it returns the $Ind^* = (\{Idx^*_{1,l}\}_{l \in U_w}, Idx^*_2, \{Idx^*_{3,i}, Idx^*_{4,i}\}_{i \in [n_D]}, \mathbf{S}^*)$ to A by running the $IdxG(GPK, \mathbf{S}^*, w_\rho)$ algorithm with the challenge access structure $\mathbf{S}^*$, where $Idx^*_{1,w_\rho} = Z^{H(w_\rho)}$, $Idx^*_2 = g^{\alpha r}$, $Idx^*_{3,i} = g^{rv_i}$, and $Idx^*_{4,i} = F(b(i))^{v_i}$.

Two different situations require attention as follows:

(1) $P = e(g, g)^{\alpha\beta\gamma}$. Let $\mu = \alpha$ and $a = \beta\gamma$, and then, the index set obtained in this case is the real index. $Idx^*_{1,w_\rho} = e(g, g)^{a\mu \cdot H(w_\rho)}$, $Idx^*_2 = g^{\mu r}$, $Idx^*_{3,i} = g^{rv_i}$,$Idx^*_{4,i} = F(b(i))^{v_i}$

(2) $P = e(g, g)^p$. In this case, A cannot get information about ρ because of the randomness of p

Phase 2. A can perform IdKey query, token query, and index query to B. The queries sent by A must also meet the above conditional restrictions.

Guess. A performs a guess ρ' for ρ; if $\rho' = \rho$, it wins. If it is case one $P = e(g, g)^{\alpha\beta\gamma}$, then $\Pr[\rho' = \rho] = \varepsilon + 1/2$; if it is case two $P = e(g, g)^p$, then $\Pr[\rho' = \rho] = 1/2$. Finally, we can get that the probability that B can resolve DBDH assumption is

$$\left|\frac{1}{2}\Pr\left[\rho' = \rho\right] \mid P = e(g, g)^{\alpha\beta\gamma}\right| + \left|\frac{1}{2}\Pr\left[\rho' = \rho\right] \mid P = e(g, g)^{z}\right| = \left|\left[\frac{1}{2}\left(\frac{1}{2} + \varepsilon\right) + \frac{1}{2} \cdot \frac{1}{2}\right] - \frac{1}{2}\right| = \frac{\varepsilon}{2}. \tag{13}$$

$\varepsilon/2$ is negligible due to the difficulty of the BDDH problem; that is, it is negligible that A can break the advantage of our scheme; that is to say, our scheme is security.

7. Performance Comparison

The performance of other related scheme [34–37] is compared with this scheme in this section. Let Ep and Ep_T denote exponential operation in G and G_1; Pa denotes pairing operation. For convenience, N_T indicates the number of attributes in the decryption operation and search operation in the system, N_e indicates the number of attributes in the encryption operation in the system, N_s denotes the number of attributes in the token generation in the system, and N_w indicates the number of attributes in the index generation in the system. Let symmetric encryption and decryption operations expressed as Sym.

We use JPBC library version 2.0.0 for related experiments. The experiment was simulated on Windows system with an Intel(R) Core (TM) i5 CPU 3.20GHz and 8.00GB

RAM to approximate the actual operation. We have obtained the measured values of exponentiation and pairing operations. The operating times of Ep, Pa, and Ep_T are 10.9 ms, 7.8 ms, and 0.15 ms, respectively. Table 1 shows the comparison of our scheme with other schemes in terms of encryption cost, decryption cost, and other aspects.

Figure 4 shows the comparison of the cost of encryption and decryption between our scheme and the other three multiauthority attribute-based encryption schemes. It is not difficult to see that the encryption and decryption time has a linear relationship with the number of attributes. Our scheme shifts the decryption process to operate on the cloud server, which makes the user's computational cost effectively reduced.

Figure 5 compares our scheme with the scheme [36]. It is not difficult to see from the figure that our scheme is highly efficient in index generation, search, and token generation stages. Among them, in the token generation stage, our scheme transfers the work of token generation to the blockchain node, and users only need to generate part of the token.

Obviously, because most of the calculation and storage work in the scheme is handed over to cloud servers and blockchain nodes, this makes our scheme more efficient in all aspects, especially in user decryption and token generation. Although the performance of some algorithms will be affected by the throughput of the blockchain and other factors, the security of the scheme will not be affected.

8. Conclusion

In this essay, we have presented a new BC-SABE scheme that replaces the centralized key management server in [31] using a consortium blockchain. The consortium blockchain consist of a trusted set of consensus nodes and is responsible for jointly generating the relevant partial parameters. We can guarantee the confidentiality of data transmission using the Pedersen secret sharing protocol, which enables sharing of subsecrets among consensus nodes, and the reciprocity protocol ensures that key information is shared without a trusted party. The update of the user revocation list is also performed entirely by the blockchain without re-encrypting the ciphertext. In addition, we move the predecryption operations to be performed in the cloud, and users are able to fully decrypt them with only a small amount of computation. Performance analysis shows that this scheme is more efficient compared to other schemes.

Data Availability

We guarantee the confidentiality of data transmission.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

This research was funded by the National Natural Science Foundation of China under Grant No. 61902140, No. 61972095, No. 62072104, and No. U21A20465; the Anhui Provincial Natural Science Foundation under Grant No. 1908085QF288; the Nature Science Foundation of Anhui Higher Education Institutions under Grant No. KJ2021A0527, No. KJ2019A0605, No. KJ2020A0034, and No. KJ2020A0032; the Natural Science Foundation of the Fujian Province under Grant No. 2020 J01159; and the Key Projects of Science and Technology of Henan Province under Grant No. 222102210043, No. 222102210173, and No. 222102210209.

References

[1] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): a vision, architectural elements, and future directions," *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645–1660, 2013.

[2] J. Feng, L. Liu, and Q. Pei, "Min-max cost optimization for efficient hierarchical federated learning in wireless edge networks," *IEEE Transactions on Parallel and Distributed Systems*, 2021.

[3] Y. Chen, J. Li, C. Liu, J. Han, Y. Zhang, and P. Yi, "Efficient attribute based server-aided verification signature," *IEEE Transactions on Service Computing*, 2021.

[4] J. Li, Y. Chen, J. Han, C. Liu, Y. Zhang, and H. Wang, "Decentralized attribute-based server-aid signature in the internet of things," *IEEE Internet of Things Journal*, 2021.

[5] L. Liu, M. Zhao, and M. Yu, "Mobility-aware multi-hop task offloading for autonomous driving in vehicular edge computing and networks," *IEEE Transactions on Intelligent Transportation Systems*, 2022.

[6] L. Liu, C. Chen, Q. Pei, S. Maharjan, and Y. Zhang, "Vehicular edge computing and networking: a survey," *Mobile Networks and Applications*, vol. 26, no. 3, pp. 1145–1168, 2021.

[7] P. Shi, H. Wang, S. Yang, C. Chen, and W. Yang, "Blockchain-based trusted data sharing among trusted stakeholders in IoT," *Software: Practice and Experience*, vol. 51, no. 10, pp. 2051–2064, 2021.

[8] A. Sahai and B. R. Waters, "Fuzzy identity-based encryption," *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, vol. 3494, pp. 457–473, 2005.

[9] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proceedings of the 13th ACM Conference on Computer and Communications Security*, pp. 89–98, Alexandria, VA, USA, 2006.

[10] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *2017 IEEE Symposium on Security and Privacy*, pp. 321–334, Berkeley, CA, USA, 2007.

[11] J. Li, W. Yao, Y. Zhang, H. Qian, and J. Han, "Flexible and fine-grained attribute-based data storage in cloud computing," *IEEE Transactions on Services Computing*, vol. 10, no. 5, pp. 785–796, 2017.

[12] J. Li, W. Yao, J. Han, Y. Zhang, and J. Shen, "User collusion avoidance CP-ABE with efficient attribute revocation for cloud storage," *IEEE Systems Journal*, vol. 12, no. 2, pp. 1767–1777, 2018.

[13] J. Li, Q. Yu, Y. Zhang, and J. Shen, "Key-policy attribute-based encryption against continual auxiliary input leakage," *Information Sciences*, vol. 470, pp. 175–188, 2019.

[14] Q. Yu, J. Li, and S. Ji, "Fully secure ID-based signature scheme with continuous leakage-resilience," *Security and Communication Networks*, vol. 2022, 12 pages, 2022.

[15] P. K. Malik, R. Sharma, R. Singh et al., "Industrial internet of things and its applications in industry 4.0: state of the art," *Computer Communications*, vol. 166, pp. 125–139, 2021.

[16] L. Liu, J. Feng, Q. Pei et al., "Blockchain-enabled secure data sharing scheme in mobile-edge computing: an asynchronous advantage actor–critic learning approach," *IEEE Internet of Things Journal*, vol. 8, no. 4, pp. 2342–2353, 2020.

[17] T. P. Pedersen, "A threshold cryptosystem without a trusted party," *Workshop on the Theory and Application of Cryptographic Techniques*, vol. 547, pp. 522–526, 1991.

[18] R. Gennaro, S. Jarecki, H. Krawczyk, and T. Rabin, "Robust threshold DSS signatures," *International Conference on the Theory and Applications of Cryptographic Techniques*, vol. 1070, pp. 354–371, 1996.

[19] K. Zhang, Y. Li, Y. Song, L. Lu, T. Zhang, and Q. Jiang, "A traceable and revocable multiauthority attribute-based encryption scheme with fast access," *Security and Communication Networks*, vol. 2020, 14 pages, 2020.

[20] X. Gao and L. Zhang, "Efficient anonymous ciphertext-policy attribute-based encryption for general structures supporting leakage-resilience," *International Journal of Network Security*, vol. 22, no. 5, pp. 763–774, 2020.

[21] H. Nasiraee and M. Ashouri-Talouki, "Anonymous decentralized attribute-based access control for cloud-assisted IoT," *Future Generation Computer Systems*, vol. 110, pp. 45–56, 2020.

[22] M. Ali, J. Mohajeri, M. R. Sadeghi, and X. Liu, "A fully distributed hierarchical attribute-based encryption scheme," *Theoretical Computer Science*, vol. 815, pp. 25–46, 2020.

[23] J. Li, Q. Yu, and Y. Zhang, "Hierarchical attribute based encryption with continuous leakage-resilience," *Information Sciences*, vol. 484, pp. 113–134, 2019.

[24] J. Li, N. Chen, and Y. Zhang, "Extended file hierarchy access control scheme with attribute based encryption in cloud computing," *IEEE Transactions on Emerging Topics in Computing*, vol. 9, no. 2, pp. 983–993, 2021.

[25] N. Chen, J. Li, Y. Zhang, and Y. Guo, "Efficient CP-ABE scheme with shared decryption in cloud storage," *IEEE Transactions on Computers*, vol. 71, no. 1, pp. 175–184, 2022.

[26] M. Green, S. Hohenberger, and B. Waters, "Outsourcing the decryption of abe ciphertexts," in *Proceedings of the 20th USENIX conference on Security*, pp. 1–16, San Francisco, CA, 2011.

[27] H. El Gafif and A. Toumanari, "Efficient ciphertext-policy attribute-based encryption constructions with outsourced encryption and decryption," *Security and Communication Networks*, vol. 2021, 17 pages, 2021.

[28] C. Feng, K. Yu, M. Aloqaily, M. Alazab, Z. Lv, and S. Mumtaz, "Attribute-based encryption with parallel outsourced decryption for edge intelligent IoV," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 11, pp. 13784–13795, 2020.

[29] J. Li, Y. Wang, Y. Zhang, and J. Han, "Full verifiability for outsourced decryption in attribute based encryption," *IEEE Transactions on Services Computing*, vol. 13, no. 3, pp. 478–487, 2020.

[30] J. K. Liu, T. H. Yuen, P. Zhang, and K. Liang, "Time-based direct revocable ciphertext-policy attribute-based encryption with short revocation list," *International Conference on Applied Cryptography and Network Security*, vol. 10892, pp. 516–534, 2018.

[31] H. Cui, T. Hon Yuen, R. H. Deng, and G. Wang, "Server-aided revocable attribute-based encryption for cloud computing services," *Concurrency and Computation: Practice and Experience*, vol. 32, no. 14, article e5680, 2020.

[32] M. Chase, "Multi-authority attribute-based encryption," in *Theory of cryptography conference*, pp. 515–534, Springer, Berlin, Heidelberg, 2007.

[33] H. Qian, J. Li, Y. Zhang, and J. Han, "Privacy preserving personal health record using multi-authority attribute-based encryption with revocation," *International Journal of Information Security*, vol. 14, no. 6, pp. 487–497, 2015.

[34] S. Belguith, N. Kaaniche, M. Laurent, A. Jemai, and R. Attia, "Phoabe: securely outsourcing multi-authority attribute based encryption with policy hidden for cloud assisted ioT," *Computer Networks*, vol. 133, pp. 141–156, 2018.

[35] K. Sethi, A. Pradhan, and P. Bera, "Practical traceable multi-authority CP-ABE with outsourcing decryption and access policy updation," *Journal of Information Security and Applications*, vol. 51, article 102435, 2020.

[36] Y. Miao, J. Ma, and X. Liu, "Practical attribute-based multi-keyword search scheme in mobile crowdsourcing," *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 3008–3018, 2017.

[37] Q. Xu, C. Tan, W. Zhu, Y. Xiao, Z. Fan, and F. Cheng, "Decentralized attribute-based conjunctive keyword search scheme with online/offline encryption and outsource decryption for cloud computing," *Future Generation Computer Systems*, vol. 97, pp. 306–326, 2019.

[38] X. Sun, H. Wang, X. Fu et al., "Substring-searchable attribute-based encryption and its application for IoT devices," *Digital Communications and Networks*, vol. 7, no. 2, pp. 277–283, 2021.

[39] J. Li, X. Lin, Y. Zhang, and J. Han, "KSF-OABE: outsourced attribute-based encryption with keyword search function for cloud storage," *IEEE Transactions on Services Computing*, vol. 10, no. 5, pp. 715–725, 2017.

[40] J. Li, Y. Zhang, J. Ning, X. Huang, G. S. Poh, and D. Wang, "Attribute based encryption with privacy protection and accountability for CloudIoT," in *IEEE Transactions on Cloud Computing*, 2020.

[41] Y. Lu, J. Li, and Y. Zhang, "Secure channel free certificate-based searchable encryption withstanding outside and inside keyword guessing attacks," *IEEE Transactions on Services Computing*, vol. 14, no. 6, pp. 2041–2054, 2021.

[42] Y. Lu, J. Li, and F. Wang, "Pairing-free certificate-based searchable encryption supporting privacy-preserving keyword search function for IIoTs," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 4, pp. 2696–2706, 2021.

[43] Y. Lu, J. Li, and Y. Zhang, "Privacy-preserving and pairing-free multi-recipient certificateless encryption with keyword search for cloud-assisted IIoTs," *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 2553–2562, 2020.

[44] Y. Lu, J. Li, and F. Wang, "Lightweight public key authenticated encryption with keyword search against adaptively-chosen-targets adversaries," *IEEE Transactions on Mobile Computing*, 2021.

[45] S. Nakamoto, *Bitcoin: A peer-to-peer electronic cash system*, Decentralized Business Review, 2008.

[46] A. Wu, Y. Zhang, X. Zheng, R. Guo, Q. Zhao, and D. Zheng, "Efficient and privacy-preserving traceable attribute-based encryption in blockchain," *Annals of Telecommunications*, vol. 74, no. 7-8, pp. 401–411, 2019.

[47] S. M. Pournaghi, M. Bayat, and Y. Farjami, "MedSBA: a novel and secure scheme to share medical data based on blockchain technology and attribute-based encryption," *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, no. 11, pp. 4613–4641, 2020.

[48] S. Liu, J. Yu, Y. Xiao, Z. Wan, S. Wang, and B. Yan, "bc-sabe: blockchain-aided searchable attribute-based encryption for cloud-IoT," *IEEE Internet of Things Journal*, vol. 7, no. 9, pp. 7851–7867, 2020.

[49] H. Zheng, J. Shao, and G. Wei, "Attribute-based encryption with outsourced decryption in blockchain," *Peer-to-Peer Networking and Applications*, vol. 13, no. 5, pp. 1643–1655, 2020.

[50] L. Guo, X. Yang, and W. C. Yau, "TABE-DAC: efficient traceable attribute-based encryption scheme with dynamic access control based on blockchain," *IEEE Access*, vol. 9, pp. 8479–8490, 2021.

[51] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612-613, 1979.

Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 2032165, 19 pages
https://doi.org/10.1155/2022/2032165

Research Article

A Transaction Traffic Control Approach Based on Fuzzy Logic to Improve Hyperledger Fabric Performance

Lei Hang,[1] **BumHwi Kim**,[2] **and DoHyeun Kim**[3]

[1]*Shanghai Normal University Tianhua College, Shanghai 201815, China*
[2]*Daegu-Gyeongbuk Research Center, Electronics and Telecommunications Research Institute, Daegu 2994, Republic of Korea*
[3]*Department of Computer Engineering, Jeju National University, Jeju 63243, Republic of Korea*

Correspondence should be addressed to DoHyeun Kim; kimdh@jejunu.ac.kr

Received 21 October 2021; Revised 9 February 2022; Accepted 1 March 2022; Published 24 March 2022

Academic Editor: Qingqi Pei

Blockchain is a potential technology for migrating the central server's processing burden into a decentralized, secure, and transparent manner. This technology has had significant influence and revolution in various industries. However, the limited transaction processing power is a stumbling block, especially compared to proven alternatives like distributed database systems. This paper proposes a transaction traffic control approach based on fuzzy logic to enhance the blockchain network's transaction processing capacity. The proposed fuzzy controller is implemented in the smart contract to adjust the transaction traffic flow in real-time network conditions automatically. An experiment environment is built on Hyperledger Fabric to demonstrate the design approach's significance. According to the experiment results, the blockchain performance is significantly improved compared to the baseline. Furthermore, the proposed approach is integrated with an existing blockchain performance-enhancing tool, and the results indicate that the proposed approach is flexible enough to integrate with other existing approaches.

1. Introduction

Distributed ledger systems, such as blockchain, allow for secure and verifiable transactions without a trusted third party [1]. Blockchain technology is progressively turning into style, with applications in varied domains like finance, supply chains, healthcare, the Internet of Things (IoT), sharing economy, and vehicular edge computing [2–8]. A blockchain is a distributed ledger of transactions maintained by all of the blockchain network's participating nodes. The transactions are a series of linked blocks that reflect the business logic. Every node in the network maintains a duplicate copy of the ledger and updates it with new blocks when all nodes agree.

As a result, it is widely expected that blockchain will substantially impact various industries, including finance and real estate, government administration, energy, and transportation [9]. However, to be practical, blockchain must handle transaction rates equivalent to those provided by traditional database management systems and provide some of the same transactional guarantees. Performance is one of the biggest problems in implementing blockchain solutions to replace present centralized servers [10]. Limited scalability, throughput bottleneck, transaction latency, and storage limits may hinder blockchain adoption due to inefficient transaction processing capability and a lack of standards [11]. Bitcoin, for example, has a 1 MB block size restriction and creates a new block every 10 minutes. As a result, the Bitcoin network is limited to 7 transactions per second, rendering it unsuitable for trading with a high rate of change [12]. Ethereum transactions take about 15 seconds to complete [13], though the average time would increase exponentially as network conditions change. Anyone can join a permissionless blockchain like Bitcoin or Ethereum, and each member is anonymous. This means that neither the agreements themselves nor the information exchanged can be kept private as a result. Permissionless platforms typically offer tokens to incentivize excessive mining or accelerate transaction processing to compensate for the lack of privacy. A negative link with the adoption of digital

currencies can significantly impact transaction costs and speed. It also makes it challenging to collaborate with other decentralized platforms because the tokens used on each platform must be consistent [14]. Since permissionless blockchains have poor performance and scalability, side chains are used to offload transaction processing from the main chain.

We concentrate on permissioned blockchains, which have known identities of all participating nodes instead of permissionless blockchains that do not restrict network membership. A permissioned blockchain is a technique to safeguard interactions among a set of entities that have a common aim but may not fully trust one another [15]. The resultant throughput is substantially higher since traditional consensus techniques such as crash fault-tolerant (CFT) or byzantine fault-tolerant (BFT) consensus protocols may be utilized, requiring expensive mining to commit transactions to the ledger.

Table 1 depicts the distinctions between permissionless and permissioned blockchains to overview the two briefly. In comparison to permissionless blockchains, permissioned blockchains are faster, more energy-efficient, and easier to implement. However, there is a lack of depth and complexity in the study on analyzing and improving permissioned blockchain performance. Permissioned blockchain's performance can start to compete with traditional databases for small systems, according to the results in [16]. Differences in consistency models and setup, on the other hand, can have a significant effect on the overall performance.

Hyperledger Fabric [17], of the most well-known permissioned blockchain frameworks for enterprise use cases, implements a distributed ledger platform for running smart contracts, leveraging familiar and proven technologies, with a modular architecture and pluggable components aimed at private enterprises. Many studies have investigated the performance modeling of Hyperledger Fabric. In [18], the authors evaluated multiple Hyperledger Fabric versions (v0.6 and v1.0). In terms of throughput and latency, the results show that Fabric v1.0 outperforms Fabric v0.6. To see how the Fabric network behaved, the authors altered block sizes, resource allocation, state database (DB), and endorsement policies in [19]. The authors explored the impact of peer central processing unit (CPU) and disk type on blockchain latency and throughput in [20]. The authors in [21] evaluated the performance of Hyperledger Fabric depending on the indexes such as throughput, latency, number of transactions, and scalability. The results showed that increasing the transaction send rate significantly impacts network performance, particularly latency. However, the throughput approaches zero when the network reaches its maximum capacity. In [22], the authors used seven distinct scenarios to test the performance of Hyperledger Fabric in terms of transaction throughput and network latency. The impact of various parameters such as batch-timeout, batch size, and the number of peers was investigated in these scenarios. Performance bottlenecks differ slightly at each stage depending on the network setup environment. In [23], the authors looked into the performance characteristics of each phase and assessed ordering services like Solo, Kafka, and Raft. The validation stage has been identified as one of the primary bottlenecks affecting transaction processing capability in this research. Because of the continual expansion in network utilization, the performance of Hyperledger Fabric is a significant problem for businesses. Permissioned blockchain platforms, in their current state, cannot meet the performance requirements of massive provenance use cases in the finance industry, such as stock exchanges, credit card companies, and mobile payment platforms. As a result, it is critical to enhance Fabric's performance to keep pace with its rapid expansion. Even with Fabric's present performance, nodes are overprovisioned to meet peak loads because there is no way to scale up or down a network, and unnecessarily high production costs arise.

Recent optimizations include parallel transaction validation, block and identity certificate caching, and bulk reading of slow CouchDB for the validation phase. However, most of these studies modify the Fabric network's original architecture by updating some stages of the transaction process or adding external user-specific modules. Hyperledger Fabric's architecture is still in the works, receiving various changes and bug fixes during development. There are times when new versions outside of the user-specified release cycles may lead to instability and compatibility issues. Besides, these studies present complex configurations which require deep knowledge of blockchain infrastructure, making it difficult to use for ordinary users. These are the main issues that we need to direct our attention to when thinking about improving the performance of the blockchain.

This paper presents a novel transaction traffic control approach based on fuzzy logic to improve blockchain performance. The fuzzy-based transaction traffic controller in the smart contract can automatically control the transaction traffic flow according to network conditions monitored without third-party intervention. Developers can choose the appropriate language to depend on to implement the smart contract without concerning the blockchain's infrastructure. The performance of the designed approach is evaluated in a clinical trial testbed built on Hyperledger Fabric. The evaluation results indicate that the proposed approach can significantly improve the transaction throughput while reducing the transaction latency.

The contributions of this paper are summarized as follows:

(i) Novelty: this paper presents a novel transaction traffic approach based on fuzzy logic to improve blockchain performance. The fuzzy controller resides in the smart contract so as to automatically perform different operations on received transactions according to real-time network conditions.

(ii) Usability: the usability of the proposed approach has been demonstrated in a clinical trial testbed built on Hyperledger Fabric from our previous work. The experiment results indicate that the blockchain performance is improved significantly in terms of transaction throughput and latency compared to the baseline.

Table 1: Comparison of permissionless and permissioned blockchain platforms.

Property	Permissionless blockchain	Permissioned blockchain
Consensus participant	Public ownership	The nodes that have been chosen
Permission	Public	Either public or private
Efficiency	Low	High
Decentralization	Fully decentralized	Partially decentralized
Cost	Not cost-effective	Cost-effective
Consensus mechanism	Proof of stake, proof of work	BFT, raft, CFT
Use case	Bitcoin, Ethereum	Hyperledger Fabric

(iii) Scalability: the scalability of the proposed approach has been validated by integrating with an existing blockchain performance-enhancing tool, and the results show that the network's performance can be improved further with the proposed approach.

The remainder of this paper is organized as follows: Section 2 examines existing research on improving blockchain performance. The designed transaction traffic control approach based on fuzzy logic is presented in Section 3. The execution of the proposed approach is detailed in Section 4. Section 5 presents the experimental setup and evaluates the performance of the designed algorithm in the clinical trial testbed. The security of the proposed approach is discussed in Section 6. Finally, Section 7 concludes the work by outlining some future research directions.

2. Related Work

Because of the technological improvements in the last few years, blockchain technology has shown to be a robust way of ensuring data integrity, particularly in trustless networks [24–27]. Even though blockchain has the property of data integrity, using it to improve distributed systems entails accepting obligatory performance disadvantages such as high latency and low throughput [28]. Blockchain has received much attention as a crucial technology that enables decentralized and incredibly reliable database management. High latency and low throughput in highly concurrent situations, on the other hand, are regarded as the primary performance bottlenecks of blockchain technology and have an impact on its adoption. Due to the blockchain's nature, the consensus process is complex. The operation and maintenance costs are high because the block data is redundantly stored in the nodes constituting the blockchain. Most of the researchers have tried to build a blockchain network based on high-performance hardware to improve throughput. However, it is challenging to use blockchain in small and medium-sized enterprises with insufficient funds due to the high cost.

Since the initial version of Hyperledger Fabric was launched in 2017, the architecture of Hyperledger Fabric has undergone various changes and bug fixes in development. The Fabric comprises miscellaneous components such as peers, membership service providers (MSPs), clients, and ordering services. The basic transaction workflow goes through the following stages: the endorsement stage, the ordering stage, and the validation stage. Each stage runs independently and does not affect the other stages. The business logic of Fabric is provided by the smart contracts that serve as a trustworthy decentralized program, gaining its trust and security from the blockchain and conjointly the entire agreement across the entire network. Fabric introduces a brand-new approach known as an execute-order-validate to perform transactions in three stages. In simple terms, a submitted transaction will be executed, thus being endorsed, ordered in a block, and before appending to the ledger, these endorsed transactions must be validated against the predefined endorsement policy. The flow of transaction execution across the network is illustrated in Figure 1. The client application should have credentials issued by the certificate authority (CA) to induce approval for submitting dealings proposals. The CA issues credentials to clients who want to submit transaction proposals and authenticates the identities of these clients before they are allowed to participate in the network. Client applications generate transactions, and the application software development kit (SDK) creates connections between the client application and network peers. These peers are the basic units to form the network, and they can be separated into endorser peers or committer peers depending on the type of task. Endorser peers perform on proposals, sign them with their signatures, and then respond with approvals or rejections. Each endorsed transaction is validated against the endorsement policy by committer peers, who then add the block of transactions to the ledger. Endorser peers in a simulated environment handle the received transaction proposal by invoking the smart contract. At this time, the results of transaction execution will not be replicated in the ledger.

Every endorser peer signs the read and write (RW) set and returns proposal responses to the client application for inspection. The client verifies the endorsing signatures to check if the required endorsement policy (e.g., the required number of endorser peers that have to endorse the transaction execution results) has been consummated. These signed transactions are packaged and submitted along with RW sets to the orderer by the client. The batched data is ordered into a block by the orderer and delivered to any or all committer peers. Every committer peer validates the transaction by checking whether the RW sets match the current state or not. Once the committer peer validates the transaction, it writes it to the ledger and updates the state using the Write

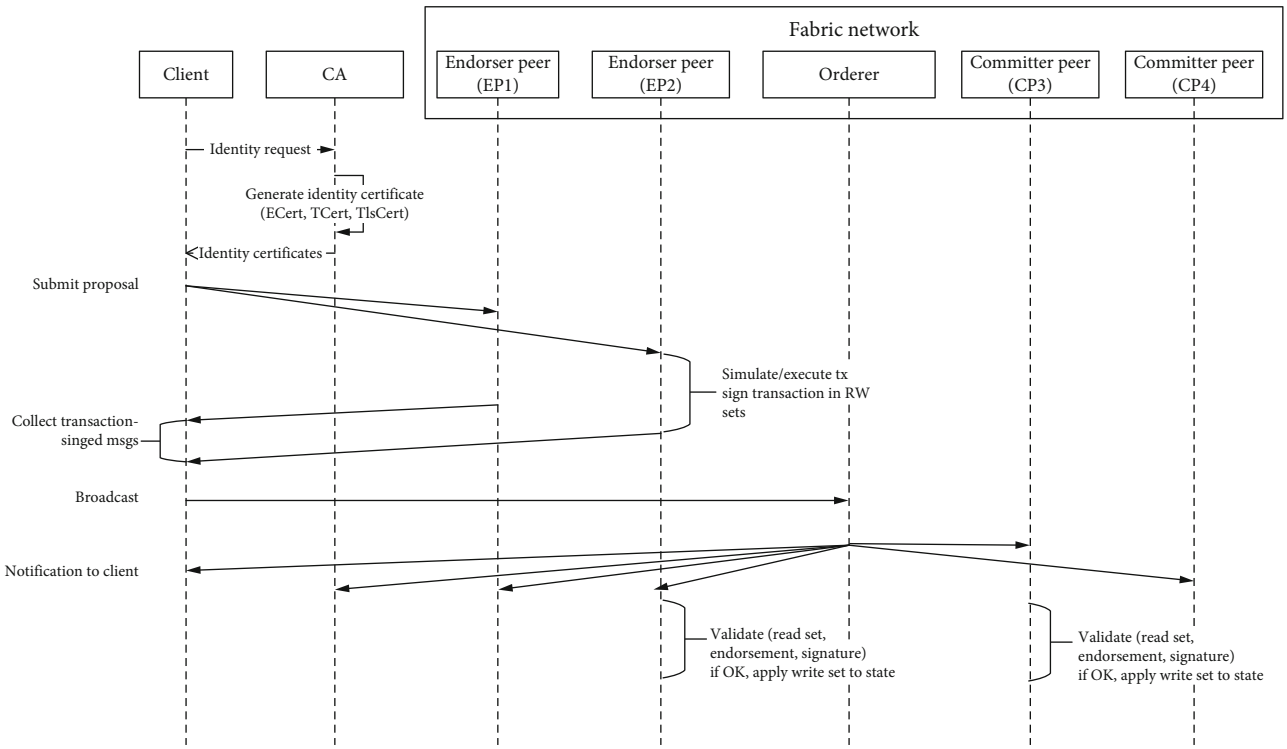


Figure 1: Transaction processing workflow in Hyperledger Fabric.

data from the RW set. Finally, the committer peers send events to the client application to inform whether the submitted transaction succeeded or not. Besides, client applications can subscribe to events to be notified by every committer peer once an event happens.

There are few studies on improving Fabric performance or optimizing its architecture. The authors in [29] propose a configurable blockchain system with a new consensus algorithm to adjust the verification process. The network scale-up/out, space efficiency, and security level are all factors in determining transaction processing capability. As a constraint, the basic authentication procedure is performed within the Hyperledger Fabric network, and general network security is in place. Besides, all peer-to-peer (P2P) data transmission utilizes the public key infrastructure (PKI) encryption module.

Nexledger Accelerator [30] is a novel transaction processing engine with an independent and modular structure that acts as an intermediate between application clients and the blockchain network. Such a feature is appropriate for IoT applications since IoT devices might not have enough computing power to run a novel approach to improve blockchain performance. Nexledger Accelerator provides a straightforward, however robust, transaction processing algorithm using batch scheduling. It classifies the incoming transactions into batched transactions. To this end, Nexledger Accelerator is fastidiously designed to settle on the self-adaptive batch size, counting on the characteristic of requested transactions and the remaining computing resource of the blockchain network. A similar approach is also presented in [31] to reduce the network latency and the number of transfers by batching chaincode execution. Batch processing refers to transferring/handling transactions in a batch by grouping transactions into a set of data rather than processing them sequentially.

Though multiple studies are dedicated to addressing low throughput, these studies only focus on the use cases with giant transaction volumes, neglecting conflicting transactions. Conflicting transactions are those transactions initiated by the client with the constant primary key. The system performance of Hyperledger Fabric may be severely degraded if various conflicting transactions exist. The authors [32] suggest utilizing a cache-enabled endorser to discover conflict transactions before executing the smart contract. When the system is integrated with the cache, it takes fewer steps than the original system to reject conflict transactions. The cache is enforced on the local endorser. The account code is retrieved from the incoming data and saved in the cache as necessary information before each transaction is completed. If the account code is located in the cache, the endorser can refuse to execute such transactions and alert clients, shortening the time between transactions.

Hyperledger Fabric is designed with flexibility and generality in mind, allowing for a wide range of nondeterministic smart contracts and pluggable services to be implemented. Version 1.0, on the other hand, does not include a BFT ordering service implementation, instead offering only a crash fault-tolerant ordering service. The authors in [33] present the design, implementation, and evaluation of a new BFT ordering service, similar to the Practical Byzantine Fault Tolerance (PBFT) protocol. Even

with ordering nodes spread across regions, the BFT-SMART ordering service can achieve up to 10 k representative transactions per second and write a transaction irreversibly within the blockchain in half a second, according to the evaluation conducted on a local cluster and in a geodistributed setting.

In Hyperledger Fabric, every peer node is linked to a distributed repository, so-called state database, to store the latest values of ledger data. LevelDB and CouchDB are the two-state databases that Hyperledger Fabric currently supports. LevelDB is a peer-to-peer database that allows for comparatively speedy access. CouchDB is a client-server database that can be accessed over an HTTP-based representational state transfer (REST) application programming interface (API). The authors in [34] redesign the transaction validation phase of Hyperledger Fabric based on the analysis from a fine-grained breakdown of the validation latency. An optimization approach using chaincode cache is proposed to enable ledger reading and writing operations in parallel. The experiment results reveal performance improvements of 2x for CouchDB and 1.3x for LevelDB.

Similarly, the authors in [35] rearchitect Hyperledger Fabric to increase transaction throughput. They focus on performance bottlenecks beyond the consensus algorithm and present architectural changes that reduce computation and input or output (I/O) overhead during transaction ordering and validation phases. The authors in [36] develop a Hyperledger Fabric GoLevelDB benchmark to characterize database access performance by simulating the transaction behaviors. The characterization results reveal several performance bottlenecks and identify some optimization opportunities to achieve better performance. The authors in [37] propose two other competent APIs between the chaincode and the peer. The first API is called the Differential Update State (DUS), which can reduce reading the state of the key before writing the updated value. As the name implies, the DUS API provides a specified set of operations to compute the updated values via different operations and writes the ledger's commutated value. The second API is called the Compound Request (CR), which supports read, write, and combined functions. It executes all the requests in a specified order and removes the number of requests compared to the DUS API. This feature makes it suitable for use cases requiring frequent parameter read and initialization operations. QiQi, a component-level performance isolation approach for Hyperledger Fabric, is presented by the authors in [38]. By monitoring Fabric's performance, this technique may dynamically adjust the CPU scheduling of Fabric components. The experiment results show that QiQi can support a variety of Fabric ordering services and chaincodes. By rearchitecting Hyperledger Fabric, the authors provide a pipeline execution of validation and commit phases in [39]. They also introduce the sparse peer, a new type of node that may selectively commit transactions to avoid CPU and I/O intensive tasks.

Table 2 describes the comparison between the proposed approaches with some existing studies discussed above. It is evident from the table that most of the existing studies modify the original architecture or process of Hyperledger Fabric. This may result in incompatibility issues, especially when a new version is released. This paper's proposed approach does not change the original system as the fuzzy controller is directly deployed into the smart contract that is flexible enough to be extended. Nexledger Accelerator is a recent blockchain performance-enhancing tool with similar features to the proposed approach. Although it is also built on an independent and modular architecture, however, it is only specified to the Fabric network. Besides, Nexledger Accelerator's configuration is complicated, making it difficult to use, especially for people who know little about blockchain. The proposed approach can be applied to any other blockchain platforms that support smart contracts. Developers can choose the appropriate language to depend on to implement the smart contract without concerning the blockchain's infrastructure.

3. Designed Architecture of the Transaction Traffic Control Based on Fuzzy Logic in Smart Contract

3.1. System Architecture. Figure 2 illustrates the proposed system's architecture, which comprises the admin, transaction traffic measurement analyzer, blockchain adaptor, benchmark DB, and the Hyperledger Fabric network. The Fabric network consists of various peers who copy the distributed ledger and a smart contract. The admin can configure the benchmark and network files for the performance evaluation. A network configuration file describes the system under test and specifies the network's connection requirements. The performance benchmark workload and user-specified test files are specified in a benchmark configuration file. The blockchain adaptor receives the transactions from the client where the workload happens and sends commands to initialize the blockchain network. Multiple clients can submit transactions to the blockchain network and return the transaction responses by invoking the functions specified in the smart contract. The transaction traffic measurement analyzer reads predefined performance statistics and stores benchmark results into the benchmark DB. The fuzzy controller adjusts the transaction acceptance rate by comparing transaction throughput, transaction latency with the acceptance rate. Transaction throughput and transaction latency are input parameters of the fuzzifier. Rules are evaluated in the inference engine. The defuzzifier converts output data (acceptance rate) into nonfuzzy values. The transaction control module obtains the output value to adjust the transaction acceptance rate. The whole process is repeated, and the transaction processing capability of the Fabric network can be dynamically maintained at a suitable level.

Figure 3 details the block diagram of the network configuration. The admin creates crypto certificates for each network entity and updates the network configuration, which specifies the network's topology. The blockchain adaptor consists of a config validator, Fabric SDK, and network configuration module. The config validator validates each network configuration object. The Fabric SDK provides the interface to connect with the Fabric network. The network

Table 2: Comparison between the proposed approaches with existing studies from the literature.

Name	Change of architecture	Use of smart contract	Interoperability	Compatibility	Configuration difficulty
[29]	Yes	No	No	No	High
[30]	No	No	No	Yes	High
[31]	No	No	No	Yes	High
[32]	Yes	No	No	No	Low
[33]	Yes	No	No	No	High
[34]	Yes	No	No	No	Low
[35]	Yes	No	No	No	High
[36]	No	No	No	Yes	Low
[37]	Yes	No	No	No	High
[38]	No	No	No	Yes	High
[39]	Yes	No	No	No	High
Proposed approach	No	Yes	Yes	Yes	Low

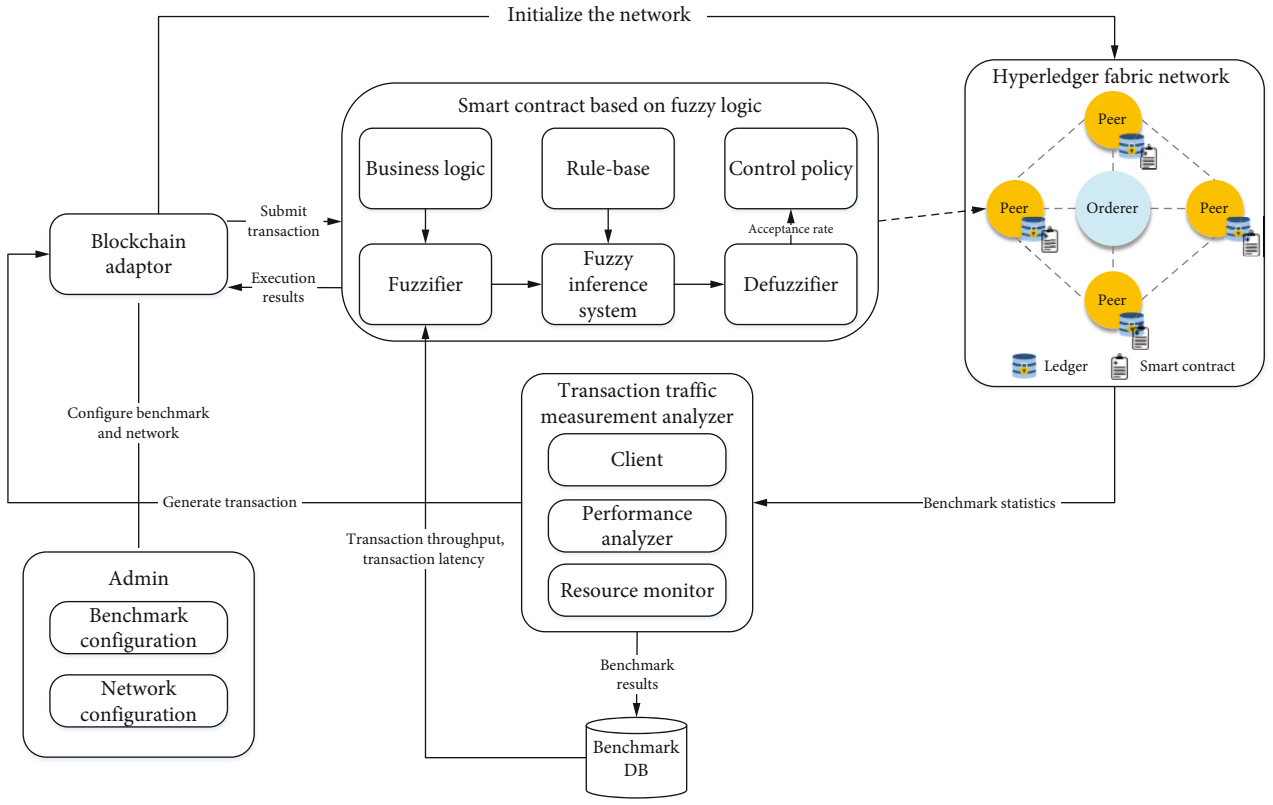


Figure 2: The system architecture of the transaction traffic control based on fuzzy logic.

configuration is used to access information in the connection profile configuration. The blockchain adaptor can send commands to initialize the network (channel, peer) and install the smart contract to the Fabric network.

Figure 4 details the structure of the transaction traffic measurement analyzer. The workload module acts as the brain of the analyzer. When the analyzer schedules transactions for a given round, it is the task of the workload control module to generate the transactions' content and submit it to the adaptor. Multiple local clients are generated according to the benchmark configuration, and each of them is connected with a rate control module. The rate control module regulates the rate of transactions at a fixed rate or follows a specific profile. The resource monitor collects statistics on resource utilization during benchmarking, while the performance analyzer calculates benchmark results according to performance statistics. Benchmark results are collated into a test report by the report generator, and a copy of the report is stored in the benchmark DB.

3.2. Fuzzy Based Transaction Traffic Control. The smart contract based on fuzzy logic is the core component of the proposed transaction traffic control approach that makes decisions based on network conditions from the blockchain.

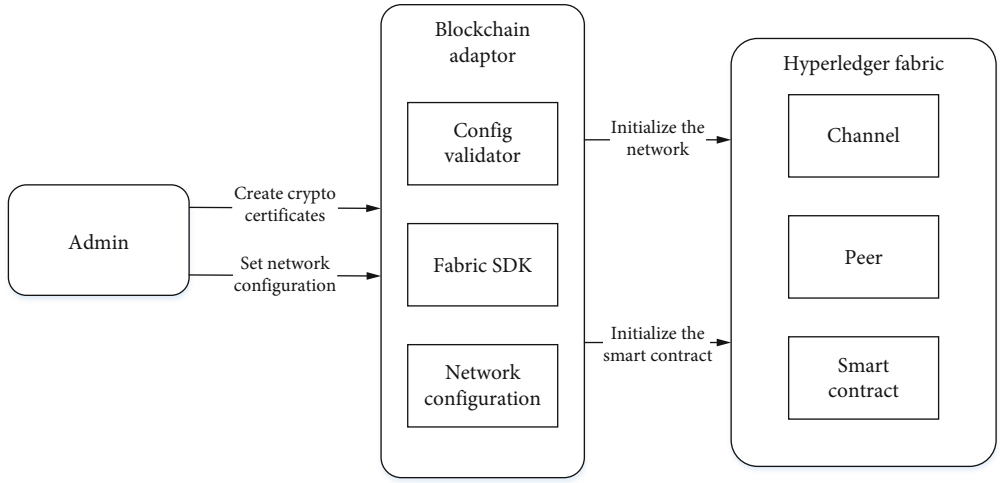


Figure 3: Blockchain adaptor block diagram.

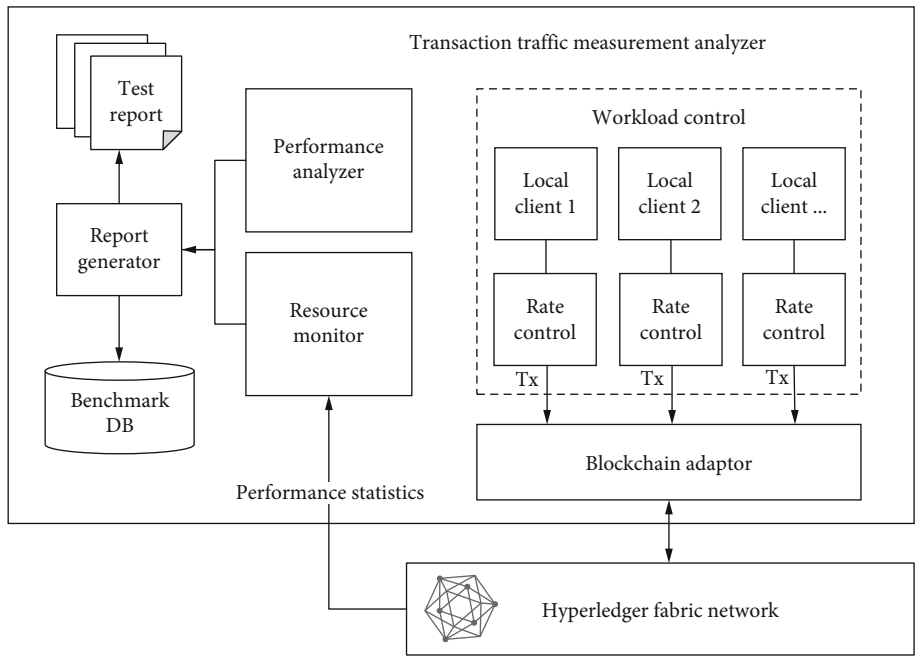


Figure 4: The detailed structure of the transaction traffic measurement analyzer.

The used fuzzy logic module keeps a favorable position by utilizing the general knowledge or experience to take a network-specific decision. In this paper, the Mamdani fuzzy system is used, one of the most well-known theories in the field of fuzzy logic control (FLC) [40]. The linguistic control strategy is born-again into an automatic control strategy supported by up-to-date data. Linguistic expression labeling information granular, like temperature for the weather or age for persons, is expressed as a linguistic variable. It is familiar and comfortable to convert linguistic values using adverbs or adjectives since natural languages do not continuously contain enough worth to define a fuzzy variable scale. This paper utilizes the Mamdani rule structure to set up linguistic modeling to regulate transaction traffic control. Figure 2 shows that the fuzzy logic module mainly consists of a fuzzification module, fuzzy rules, an inference system, and a defuzzification module. The steps of a fuzzy logic implementation can be summarized as follows:

Step 1. Define a set of fuzzy variables (see Table 3).

Step 2. Define a set of membership functions.

Step 3. Build a set of fuzzy rules for each variable (see Table 4).

Step 4. Get outcomes of the fuzzy rules by combing rule strength and output membership functions.

Table 3: Fuzzy set definition for input and output parameters.

(a)

Input variables	Linguistic terms	Fuzzy sets (a, b, c, d)
Transaction throughput	Very low	0, 0, 20, 60
	Low	20, 60, 100
	Acceptable	60, 100, 140
	High	100, 140, 180
	Very high	140, 180, 200, 200
Transaction latency	Very low	0, 0, 0.15, 0.45
	Low	0.15, 0.45, 0.75
	Acceptable	0.45, 0.75, 1.05
	High	0.75, 1.05, 1.35
	Very high	1.05, 1.35, 1.5, 1.5

(b)

Output variables	Linguistic terms	Fuzzy sets (a, b, c, d)
Acceptance rate	Very low	0, 0, 10, 30
	Low	10, 30, 50
	Medium	30, 50, 70
	High	50, 70, 90
	Very high	70, 90, 100, 100

Table 4: Sample fuzzy rule definition.

Transaction throughput	Transaction latency	Acceptance rate
Very low	Very low	Very high
Low	Low	Very high
Acceptable	Acceptable	Medium
High	High	Low
Very high	Very high	Very low

Step 5. Obtain the output by combing the outcomes.

Step 6. Defuzzify the output membership function.

Rule definition is one of the most complex and essential tasks in fuzzy inference systems. It requires domain expert knowledge; for rule generation, we have used the model evaluated from the previous study [41]. The linguistic terms of the input and output variables and their corresponding fuzzy sets are given in Table 3. In the proposed approach, the fuzzy variables for membership functions are defined as transaction throughput and transaction latency, and acceptance rate is valued as "high," "acceptable," and "low." For every acceptance rate factor, the output values are scaled in [0, 100], where the minimum value 0 expresses the lowest acceptance rate and the value 100 represents the highest acceptance rate.

We use both triangular and trapezoidal membership functions to outline the fuzzy variables within the fuzzy system. The trapezoidal fuzzy set A performs $\mu_A(x)$, assigned by four quantified variables (a, b, c, d). The mathematical illustration of the fuzzy membership function is interpreted, as shown in

$$\mu_A(x)\begin{cases} 0, & x < a, \\ \dfrac{x-a}{b-a}, & a < x < b, \\ 1, & b < x < c\,(a < b \le c \le d), \\ \dfrac{d-x}{d-c}, & c < x < d, \\ 0, & x > d. \end{cases} \tag{1}$$

It is worth noting that the trapezoidal function is triangular when b equals c. Equation (2) defines the fuzzy intersection operation between two fuzzy sets A and B, where $A, B \in U$ and x is an element in the U universe:

$$\mu_{A\cap B}(x) = \min\{\mu_A(x), \mu_B(x)\}, \quad \forall x \in U. \tag{2}$$

Moreover, their union is defined by

$$\mu_{A\cup B}(x) = \max\{\mu_A(x), \mu_B(x)\}, \quad \forall x \in U. \tag{3}$$

The proposed fuzzy controller considers two input variables: transaction throughput and transaction latency. The output variable generated by the fuzzy controller is the acceptance rate of the transaction. For combining, these two fuzzy input parameters are used to obtain rule strength by using fuzzy operators. Afterward, the fuzzy inference system evaluates each rule via the membership functions and concludes with the rule's conditions. The quantitative results of the given fuzzy sets and corresponding membership degrees are calculated by

$$\text{mCoA} = \frac{\int f(x) \bullet x dx}{\int f(x) dx}. \tag{4}$$

The defuzzification approach used is the Modified Center of Area (mCoA). It considers all areas covered by the scaled membership functions, even if they extend beyond the range of the output variable, where (x) is the aggregated membership function and x represents the output value. The integration interval is the distance between the minimum and maximum membership function values.

For rule definition, three different approaches are adopted as follows:

(1) Minimum based: in this scheme, the output value is set to the minimum value of given input, e.g.,

$$\text{If } M_1 \text{ is high and } M_2 \text{ is high, then } y \text{ is minimum.} \tag{5}$$

(2) Average based: in this scheme, the output value is set to the average value of given input, e.g.,

$$\text{If } M_1 \text{ is high and } M_2 \text{ is low, then } y \text{ is medium.} \quad (6)$$

(3) Maximum based: in this scheme, the output value is set to the maximum value of given input, e.g.,

$$\text{If } M_1 \text{ is low and } M_2 \text{ is low, then } y \text{ is maximum.} \quad (7)$$

The proposed fuzzy controller is aimed at holding the acceptance rate of the transaction at an optimum level. For example, the acceptance rate is meager if the transaction throughput is exceedingly high and has incredibly low latency. In a word, the fuzzy controller serves as a regulator of transaction traffic in line with transaction throughput and transaction latency. Table 4 presents a sample of specified fuzzy rules, and in total, twenty-five rules are defined. The total number of rules for the fuzzy controller is counted by considering all possible combinations of input variables.

3.3. Transaction Traffic Control Execution Process. Figure 5 describes the execution process of the transaction traffic control approach based on fuzzy logic. At the beginning of each test, the admin should configure the network and benchmark profiles to fulfill the test scenario's requirements. The benchmark file describes how the evaluation test should be executed, including the number of rounds, send rate of the transaction, and settings about monitoring the test network. The network configuration file describes the topology of the test network, such as the configuration of nodes, number of clients, and smart contracts deployed to the test network. Once the network is set up, the admin can start the script to start the benchmark test. One or more clients generate transactions to the adaptor; in turn, the adaptor submits transactions to the Fabric network.

Meanwhile, the transaction traffic measurement analyzer observes and collects the benchmark results. The analyzer calculates the benchmark statistics and stores the results in the benchmark DB. The fuzzifier retrieves the transaction throughput and network latency as the input parameters of the fuzzy inference system. The inference engine evaluates the input parameters according to the fuzzy rules. The defuzzifier produces the acceptance rate as the output value and sends this value to the transaction control module. The transaction module performs transaction traffic control operations concerning the acceptance rate. The transaction execution response is generated and returned to the client. This process is repeated across the entire benchmark experiment until the user stops the test. Finally, all network entities and the smart contract will be removed.

4. Implementation of the Transaction Traffic Control Based on Fuzzy Logic

4.1. Development Environment. Table 5 presents the technology stack used to implement transaction traffic control based on fuzzy logic. The Hyperledger Fabric (v1.4.1) is used as the blockchain infrastructure deployed in the Ubuntu Linux (18.04 LTS) operating system. All the network elements of Hyperledger Fabric are encapsulated as Docker images in Docker containers, which are running in the virtual machine. The Node SDK enables interactions between external applications and the Fabric blockchain network via APIs to submit transactions to the ledger or query content data. Hyperledger Caliper (v2.0.0) is an open-source blockchain benchmark tool that allows users or developers to measure different performance indexes of blockchain implementation [42]. FuzzyIS is a JavaScript library for building a fuzzy inference system in smart contracts that utilize Node.js. MongoDB is a NoSQL database used to store the benchmark results in the JSON-like document with a schema. Express.js is a Node. Js-based web server framework to build web applications provides various REST APIs to manipulate MongoDB.

4.2. Smart Contract Implementation. Algorithm 1 illustrates the process to initialize the fuzzy inference system in the smart contract. The fuzzy inference system in the smart contract contains three core objects. The linguistic variable initializes and adds input and output linguistic variables into the system. In the proposed fuzzy inference system, the input linguistic values are *transaction throughput* and *transaction latency*, while the output linguistic value is the acceptance rate. The variable term describes fuzzy terms for each variable like high/low and very high/very low. The rule describes the connection between input and output linguistic variables. These are conditions like: "if *transaction throughput* is very low and *transaction latency* is very low, then *acceptance rate* should be very high," which describes how the system works. The fuzzy inference system is created with input and output linguistic variables along with described rules. It calculates precise values for output variables referring to the rules given. We can express the algorithmic complexity by using the Big-O asymptotic notation. Algorithm 1 is a constant-time function that can be expressed as $O(1)$.

Algorithm 2 describes the process of invoking the fuzzy inference system in the smart contract. When the smart contract is invoked, it initializes the *Benchmark Url* and connects with the database. A new database instance is created, and the smart contract retrieves the latest record. *Transaction throughput* and *transaction latency* values are extracted from the result. These two values are used as the fuzzy inference system's input variables to compute the output variable *acceptance rate*. Afterward, the smart contract performs different operations on the transaction according to the *acceptance rate's* value, as described in Algorithm 3. Three operations (drop, delay, and accept) are defined in the smart contract along with three thresholds. The algorithmic complexity of Algorithms 2 and 3 is also a constant-time function that can be expressed as $O(1)$.

5. Performance Evaluation

5.1. Clinical Trial Testbed. In this section, we verify the efficiency and usability of the proposed approach by applying it in the clinical trial testbed from our previous work [43]. The workflow of the proposed system is illustrated in Figure 6.

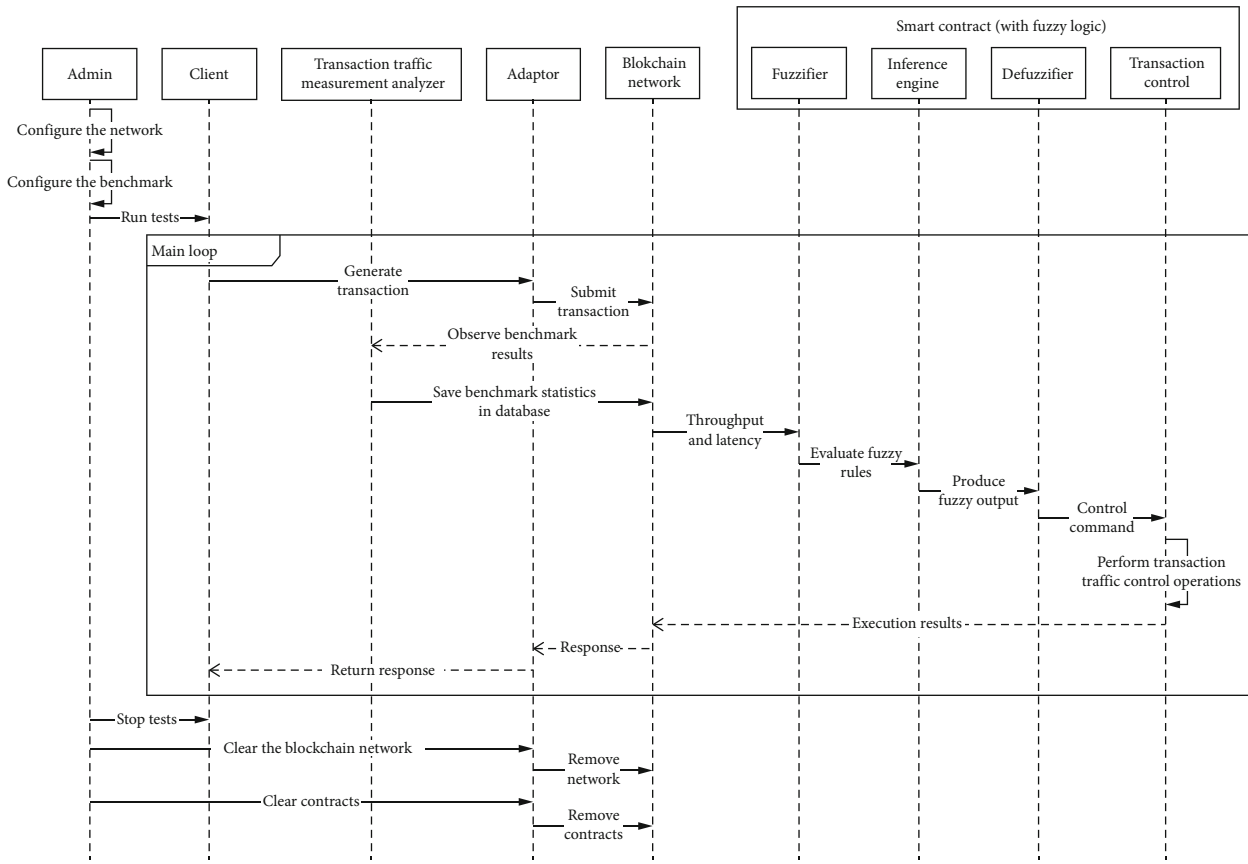


Figure 5: Sequence diagram of the transaction traffic control approach based on fuzzy logic.

Table 5: Development environment of the transaction traffic control based on fuzzy logic.

Component	Description
CPU	Intel Core i5-8500 @ 3.00 GHz
Memory	12 GB
OS	Ubuntu Linux 18.04 LTS
Docker engine	v19.03.8
Docker-composer	v1.24.0
SDK	Node.js v8.17.0
Blockchain infrastructure	Hyperledger fabric v1.4.1
Transaction traffic measurement tool	Hyperledger caliper V2.0.0
FIS library	FuzzyIS
DBMS	MongoDB
Web server	Express.js
Programming language	JavaScript
IDE	VSCode

Each participant must have credentials to get the authorized permission for submitting a transaction to the blockchain network. The principal investigator (PI), clinical research coordinator (CRC), and clinical research associate (CRA) can only read and update their profiles. The PI and CRC can create profiles for new subjects participating in the clinical trial. They can also set profiles for devices (pillbox, bgm), which will update the settings accordingly. The devices collect biomedical data from subjects and generate electronic case report form (eCRF) pillbox/blood glucose meter (bgm) data in the blockchain. The eCRF PI consult data and lab test data are created by the CRC when the subject visits the clinical site. After confirmation by the PI, these data cannot be modified. The CRA can review the data

```
Input (Linguistic Variables, Variable Terms, Rules)
Begin
    Describe a new fuzzy inference system
    Initialize and add Linguistic Variables into the system
    If the input variable is null, then
        Initialize and add transaction throughput variable
        Initialize and add transaction latency variable
    Else
        Throw an error
    If the output variable is null, then
        Initialize acceptance rate variable
    Else
        Throw an error
    Describe Variable Terms for each variable
    IfVariable Term is null
    Describe term for transaction throughput variable
    Describe term for transaction latency variable
    Describe term for acceptance rate variable
    Else
        Throw an error
    Describe Rules for each variable
    IfRule is null
    Describe each rule in the same order as listed in the term description
    Else
        Throw an error
End
```

Algorithm 1: Initialize fuzzy inference system method.

```
Input (Transaction Throughput, Transaction Latency, BenchmarkDB Url)
Output (Acceptance Rate)
Begin
    Initialize the BenchmarkDB Url
    Connect to the database with the BenchmarkDB Url
    If an error occurs, then
        Throw an error
    Else
        Initialize the database instance
        Find the latest record from the collection
    If an error occurs, then
        Throw an error
    Else
        Extract the Transaction Throughput value from the result
        Extract the Transaction Latency value from the result
        Compute the Acceptance Rate value
        Close the database connection
End
```

Algorithm 2: Invoke fuzzy inference system method.

and generate audit queries if there exist errors in data. Afterward, the PI and CRC can access the audit and correct the data accordingly.

The smart contract for the clinical trial testbed contains seven participants, five assets, and nine transactions, as shown in Table 6. The participants are CRC, PI, CRA, subject, pillbox, bgm, and last but not least, the admin of the network. Table 6 gives a list of transactions and describes the transaction structure, which comprises the participant, operation, and resource. Participants are users who can submit the transaction to the business network. The operation specifies the action (e.g., create and read) that the transaction can perform on the resource. ALL represents that the transaction can support all kinds of actions. Resources represent either participant (e.g., CRC and CRA) or assets such as eCRF pillbox data and eCRF bgm data. Transactions submitted by a participant are to perform the specified operation against the resource.

Input (*Acceptance Rate, Drop Tx Threshold, Delay Tx Threshold, Accept Tx Threshold*)
Begin
Set the *Acceptance Rate*
Set the *Drop Tx Threshold*
Set the *Wait Tx Threshold*
Set the *Accept Tx Threshold*
If*Acceptance Rate* is less than the *Drop Tx Threshold* then
Drop the transaction
Else If*Acceptance Rate* is greater than the *Drop Tx Threshold* and less than the *Wait Tx Threshold* then
Delay the transaction
Else If*Acceptance Rate* is greater than the *Acceptance Tx Threshold then*
Accept the transaction
Else
Throw an error
End

Algorithm 3: Transaction control method.

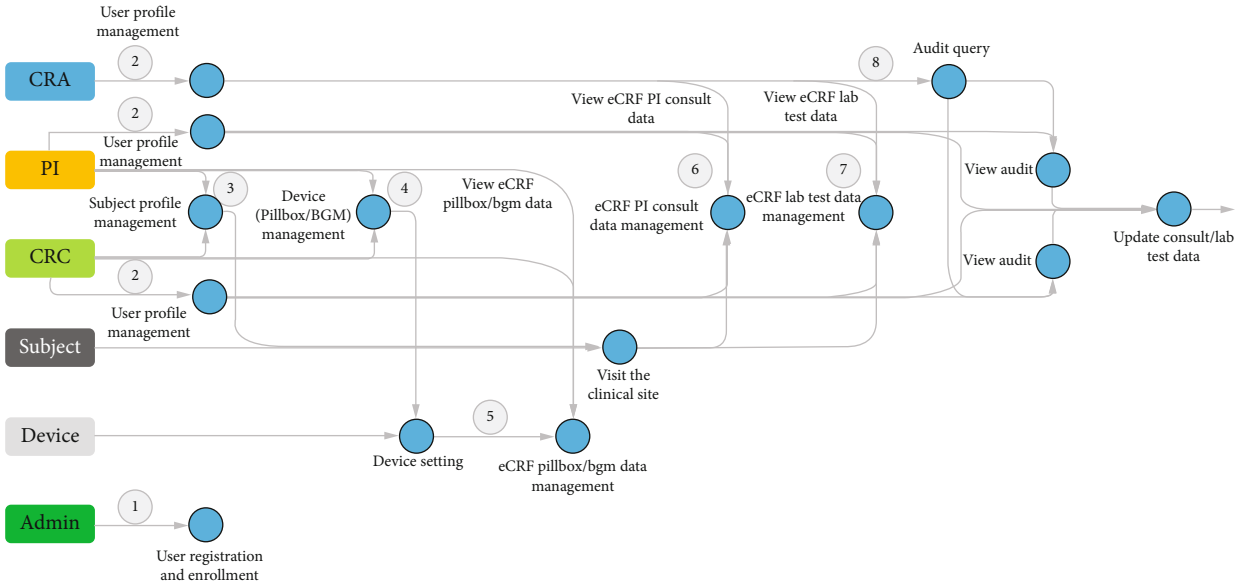


Figure 6: The workflow of the blockchain-based clinical trial service platform.

Table 6: Defined transactions in the smart contract.

Transaction	Participant	Operation	Resource (participant, asset)
User profile management	Admin	ALL	CRC, CRA, PI
Subject management	CRC, PI	ALL	Subject
Device pillbox profile management	CRC, PI	ALL	Pillbox
Device BGM profile management	CRC, PI	ALL	BGM
eCRF pillbox data management	CRC, PI, pillbox	READ, CREATE	eCRF pillbox data
eCRF BGM data management	CRC, PI, BGM	READ, CREATE	eCRF BGM data
eCRF PI consult data management	CRC, PI, CRA	ALL	eCRF PI consult data
eCRF LAB data management	CRC, PI, CRA	ALL	eCRF lab data
eCRF CRA audit	CRC, PI, CRA	ALL	eCRF audit

5.2. Setup for Experiment. This experiment was performed in a single channel of the clinical trial testbed network, consisting of 4 organizations with 6 endorser peer nodes in total. A new block is generated every 250 milliseconds, with a default block size of 10 transactions per block. The default ordering service is solo, with only one ordering node. The default

Table 7: Default experiment setup unless otherwise stated.

Parameters	Values
Number of orgs	4
Number of endorser peers	6
Endorsement policy	AND (a, b, c)
Ordering service	Solo
Block size	10 transactions per block
Block frequency (maximum timeout to create a block)	250 ms
State database	LevelDB
Programming language	Node.js
Use of TLS	No
Number of clients	5
Smart contract	Clinical trial network

state database in this experiment was LevelDB. Table 7 lists the remaining experiment parameters. The experiment's scripts were specified to target one function of our prototype: eCRF lab data generation, since the user most frequently invokes this transaction. The evaluation tests in this section were averaged over numerous rounds to reduce errors caused by system overload and network congestion. Figure 7 shows the snapshot of the command in the console to install the smart contract onto all of the peer nodes for each organization.

5.3. Performance Metrics. The two main performance metrics used to assess the blockchain network's performance are throughput and latency. The throughput can be further divided into two divisions in terms of the processes to be handled. Read throughput is a metric that counts how many read operations are accomplished in a given amount of time, expressed in reads per second (rps). Most of the systems are often deployed adjacent to the blockchain to obtain significant reading query efficiency. As a result, the read throughput of the blockchain is not utilized as a core performance metric. The rate at which the blockchain commits valid transactions in a given period, represented in transactions per second (tps), is known as transaction throughput. Transaction throughput is measured across all nodes in a network, not only at a single node.

$$\begin{aligned}\text{Read throughput} &= \frac{\text{Total read operations}}{\text{Total time in seconds}},\\ \text{Transaction throughput} &= \frac{\text{Total valid transactions}}{\text{Total time in seconds}}.\end{aligned} \tag{8}$$

Regarding the types of operations, latency can be divided into two sections. The whole time it takes to send a read request and obtain a response is read latency. Transaction latency costs the entire network to validate a transaction, including broadcasting time and consensus algorithm allocation time.

$$\begin{aligned}&\text{Read latency} = \text{response received time} - \text{submission time},\\ &\text{Transaction latency} = \text{confirmation time} - \text{submission time}.\end{aligned} \tag{9}$$

5.4. Throughput and Network Latency Evaluation. This section evaluates the performance of the proposed approach using the clinical trial testbed in terms of transaction throughput and transaction latency. This experiment is performed by scaling the transaction send rate and the number of clients to obtain a comprehensive result.

Figure 8 plots the experimental results of the baseline and the fuzzy logic scheme in terms of average transaction throughput with 1 client. The transaction throughput increased linearly with the increase in send rate until it reached around 100 tps. The transaction throughput growth slowed drastically and eventually came to a halt when the send rate exceeded this point. When the send rate was 125 tps, the transaction throughput was 95 tps, and 106.4 tps, respectively, resulting in a 12% improvement in transaction throughput. Figure 9 plots the experimental results of the baseline and the fuzzy logic scheme in terms of average transaction latency with 1 client. It is observed that the baseline network generated more transaction latency than the fuzzy logic scheme when the send rate was above the saturation point. When the send rate was 200 tps, there is a 57.3% reduction in transaction latency.

Figure 10 plots the experimental results of the baseline and the fuzzy logic scheme in terms of average transaction throughput with 5 clients. The transaction throughput increased linearly with the increase in send rate until it reached around 150 tps. The growth of transaction throughput decreased significantly and approached to a flat when the send rate was above this point. When the send rate was 150 tps, the transaction throughput was 108.6 tps, and 125.4 tps, respectively. When compared to the baseline, the fuzzy-based technique can increase transaction throughput by 15.5% at this point. Figure 11 plots the experimental

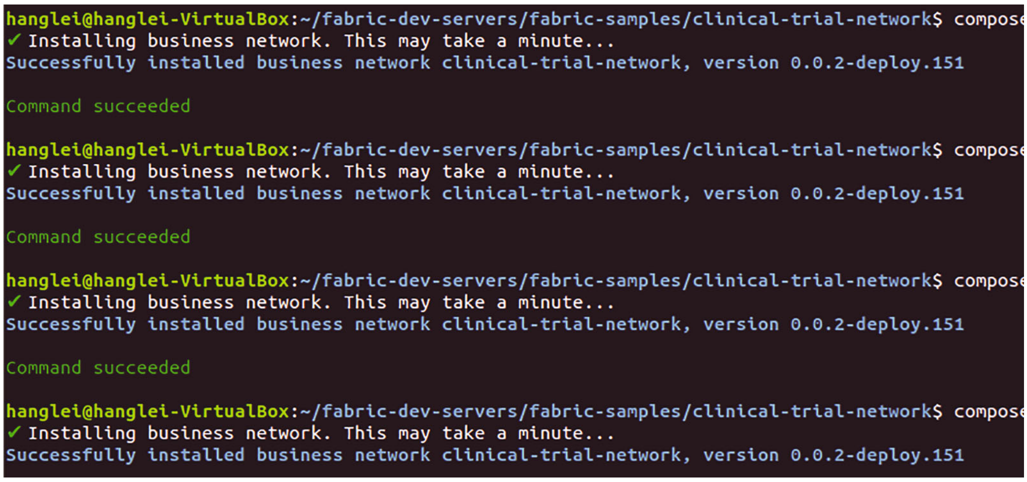


FIGURE 7: Snapshot of smart contract installation in the network.

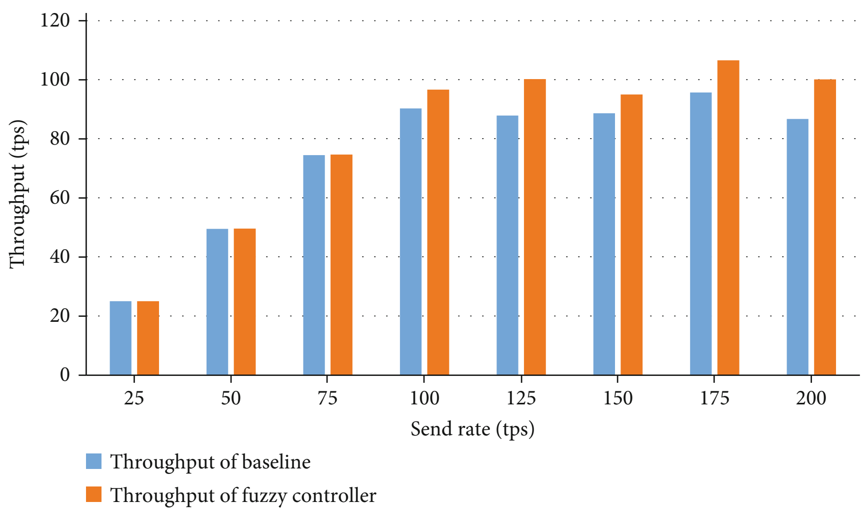


FIGURE 8: Evaluation of transaction throughput with one client.

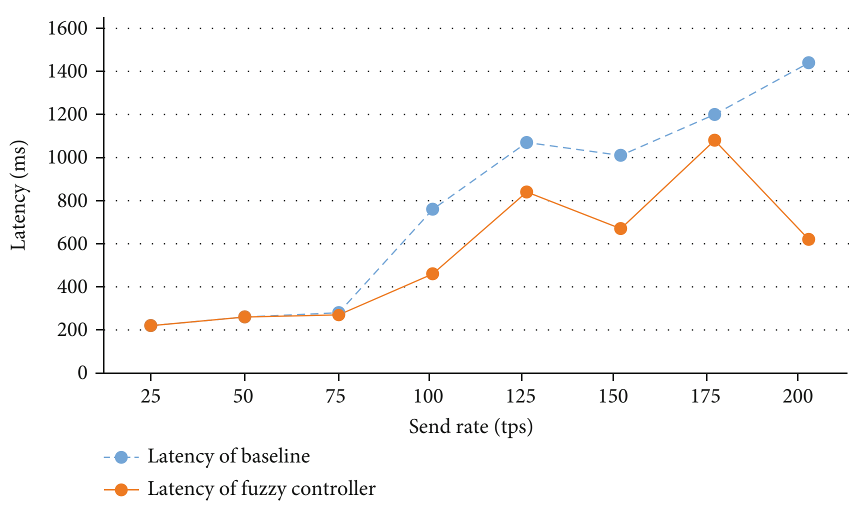


FIGURE 9: Evaluation of transaction latency with one client.

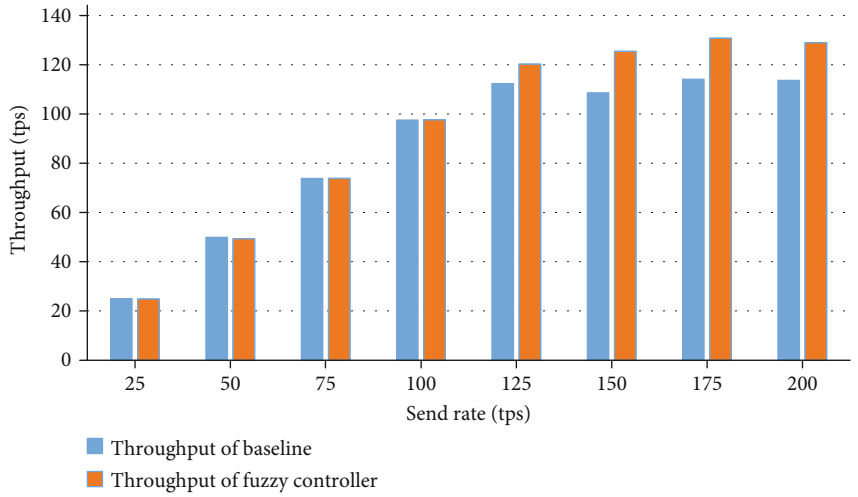


Figure 10: Evaluation of transaction throughput with five clients.

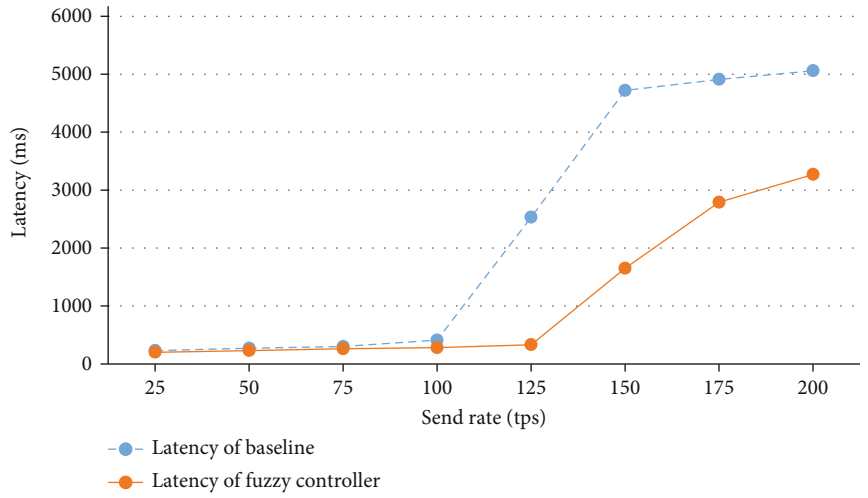


Figure 11: Evaluation of transaction latency with five clients.

results of the baseline and the fuzzy logic scheme in terms of average transaction latency with 5 clients. It is observed that the baseline network generated more transaction latency than the fuzzy logic scheme when the send rate was above the saturation point. This experiment results indicate that the fuzzy logic scheme performs better than the baseline concerning transaction latency and throughput. When the transmit rate was 200 tps, there is a 37% reduction in transaction latency.

The proposed approach is effective, especially when the send rate reaches 150 tps and more. According to the experiment results in the previous work [41], 150 tps is a saturation point of the network, and higher arrival rates can cause lower transaction throughput. The reality is that scaling the rate of transactions sent to the network will likely reach a point where transaction latency becomes untenable because the peers become saturated, consuming all the available CPU and memory resources allocated to the host server. In this case, the fuzzy controller regulates the network's acceptance rate to maintain the transaction processing capability just as we expected.

Furthermore, we perform a collaboration experiment by integrating the proposed approach with one of the existing performance-enhancing tools called Accelerator, overviewed in Related Work. Accelerator retrieves transactions from clients on behalf of blockchain nodes and routes the transactions to the blockchain network. The proposed fuzzy logic controller is applied to the nodes connected with Accelerator nodes. The proposed approach was evaluated thoroughly by comparing it to the baseline network and the baseline network with Accelerator. The experiment was carried out by altering the number of clients while maintaining a fixed send rate of 200 tps, as shown in Table 8. Compared to the baseline, the network with Accelerator boosts transaction throughput by 67.2% and lowers transaction latency by 35.4% for one client. With Accelerator and the proposed technique, transaction throughput is enhanced by 77.7%, and transaction latency is reduced by 52.3%. Compared to

Table 8: Comparison analysis of the performance evaluation with Accelerator.

Number of clients	Send rate (tps)	Performance indexes	Baseline	Accelerator	Accelerator (with proposed approach)
1	200	Transaction throughput (tps)	98.7	165	175.4
	200	Transaction latency	650	420	280
5	200	Transaction throughput (tps)	135.5	246.7	266.8
	200	Transaction latency (ms)	2350	1880	1460

Table 9: Resource utilization of the baseline network.

Name	Memory (max)	Memory (avg)	CPU (max)	CPU (avg)	Traffic in	Traffic out
local-client.js	103.2 MB	88.5 MB	12.64%	9.76%	—	—
http://peer1.company.com	115.0 MB	106.2 MB	12.36%	6.73%	5.6 MB	5.2 MB
http://peer2.home.com	114.2 MB	106.0 MB	16.92%	7.75%	5.7 MB	13.6 MB
http://peer3.home.com	94.6 MB	86.7 MB	12.95%	7.75%	5.2 MB	5.4 MB
http://peer4.hospital.com	133.4 MB	115.5 MB	11.20%	4.74%	3.9 MB	29.0 KB
http://peer5.hospital.com	109.0 MB	96.2 MB	11.46%	5.53%	4.6 MB	4.7 MB
http://peer6.cro.com	112.3 MB	107.0 MB	13.92%	5.24%	5.4 MB	11.1 MB
http://orderer.com	40.6 MB	25.4 MB	6.16%	2.85%	4.3 MB	1472 MB
ca_nodeCompany	6.5 MB	6.5 MB	0.00%	0.00%	536 B	0 B
ca_nodeHome	6.5 MB	6.5 MB	0.00%	0.00%	486 B	0 B
ca_nodeHospital	6.5 MB	6.5 MB	0.00%	0.00%	516 B	0 B
ca_nodeCRO	6.5 MB	6.5 MB	0.00%	0.00%	426 B	0 B

Table 10: Resource utilization of the prototype with the designed approach.

Name	Memory (max)	Memory (avg)	CPU (max)	CPU (avg)	Traffic in	Traffic out
local-client.js	103.7 MB	88.7 MB	12.84%	9.96%	—	—
http://peer1.company.com	115.4 MB	106.7 MB	12.46%	6.93%	5.8 MB	5.4 MB
http://peer2.home.com	114.7 MB	106.5 MB	17.22%	8.05%	5.8 MB	13.8 MB
http://peer3.home.com	95.2 MB	87.1 MB	13.05%	8.15%	5.3 MB	5.6 MB
http://peer4.hospital.com	133.9 MB	115.8 MB	11.35%	4.94%	4.1 MB	31.0 KB
http://peer5.hospital.com	112.0 MB	96.6 MB	11.67%	5.73%	4.7 MB	4.9 MB
http://peer6.cro.com	112.7 MB	107.3 MB	14.02%	5.54%	5.6 MB	11.5 MB
http://orderer.com	41.6 MB	25.6 MB	6.46%	3.05%	4.6 MB	1502 MB
ca_nodeCompany	6.8 MB	6.8 MB	0.00%	0.00%	566 B	0 B
ca_nodeHome	6.8 MB	6.8 MB	0.00%	0.00%	496 B	0 B
ca_nodeHospital	6.8 MB	6.8 MB	0.00%	0.00%	536 B	0 B
ca_nodeCRO	6.8 MB	6.8 MB	0.00%	0.00%	446 B	0 B

the baseline, the network with Accelerator increases transaction throughput by 82.1 percent and reduces transaction latency by 20% for five clients. With Accelerator, the transaction throughput is enhanced by 96.9%, and the latency is reduced by 37.9% using the fuzzy-based approach. The results show that the proposed method is interoperable enough to collaborate with other methodologies.

The performance of the proposed approach is also evaluated in terms of resource utilization, such as memory usage, CPU utilization rate, and traffic in and traffic out. Table 9 and Table 10 represents the results of the baseline and the prototype with the designed approach, respectively. From these two tables, it is observed that there is no significant burden on the network by utilizing the fuzzy-based approach. The results indicate that the proposed algorithm can efficiently utilize system resources.

This paper presents a transaction traffic control approach based on fuzzy logic to improve blockchain platforms' performance that supports smart contracts. A clinical trial testbed implemented on Hyperledger Fabric is used as part of the experimental test to demonstrate the proposed approach's usability. This technique is based on smart contracts rather than the underlying blockchain architecture, accommodating various blockchain implementations easily. Furthermore, it can work with some of the existing blockchain performance enhancement tools to get even more

significant benefits; for example, the proposed approach improves Accelerator's performance.

The experimental results in this section indicate that the proposed approach can be used in business scenarios that require high throughput and concurrency. Some other business disciplines, including supply chains and energy trading, can also benefit from the significance of this work. Due to the absence of transaction processing capabilities caused by a time-consuming consensus procedure that remains a restriction in deploying blockchain on IoT applications, the proposed approach can be expanded into current decentralized food supply chain systems to increase efficiency. IoT sensors can be attached to any product, such as fish entrusted for transportation, and transmit temperature, humidity, and location data. The fuzzy controller is placed on a smart contract that each supply chain business network party can access. The transaction traffic flow can be controlled based on network conditions in real time.

6. Security Verification

This paper introduces an adaptor that acts as a link between the user and the blockchain network. This adaptor is used to set up the network and benchmark profiles that the user has created. It can also receive transactions from numerous clients and distribute them to each network endorser peer. The CA of Hyperledger Fabric secures data transfer between the adapter and the blockchain network. The CA is in charge of digital certificate registration and issuing. By the name subject of the certificate, the digital certificate identifies the owner of a public key. The adapter can consume services using the issued certificate in X.509 to certify itself in network messages. This method allows all participants to trust the digital signature associated with the issued certificate's private key. Recipients of digitally signed messages can check if the signature is valid with the sender's public key to verify the message's origin and integrity.

7. Conclusion and Future Work

Blockchain networks provide a decentralized mechanism for peers to collaborate and create trust through business networks. Each peer node must perform operations and communicate with peers to confirm transactions, reach consensus, and update the shared ledger's status. Many well-known blockchain platforms such as Bitcoin and Ethereum have been widely adopted into different application domains. So far, there has been much confusion about whether or not the blockchain can scale, as well as a paucity of information regarding best practices for improving performance and scaling. Besides, more analysis and evaluation of the performance of these platforms are urgent.

This paper proposes a novel transaction traffic control approach using fuzzy logic to improve blockchain performance. Real-time network feedback is used as input parameters, and the fuzzy controller adjusts the transaction traffic across the whole network accordingly. A clinical trial testbed built on the Hyperledger Fabric is used as the experiment environment to evaluate the proposed approach's performance. The experiment results indicate that the designed approach can significantly improve the transaction throughput while reducing transaction latency. Furthermore, the proposed approach is applied with an existing blockchain performance-enhancing tool called Nexledger Accelerator. The results indicate that the proposed approaches integrate with the existing performance-enhancing approach and improve blockchain performance.

One of this study's limitations is that all benchmark trials are done on a single-host virtual system. The blockchain network is also running on a local network that is inappropriate for production. A future work will refine the prototype system. To test the impact of the proposed approach in the production environment, we will duplicate the results using a cloud service such as Amazon Web Services (AWS) or IBM Blockchain. Furthermore, we will verify the applicability of the proposed technique by deploying it on current smart contract-enabled blockchain platforms such as Ethereum and Corda.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this article.

Acknowledgments

This research was supported by Energy Cloud R&D Program through the Ministry of Science ICT (2019M3F2A1073387), and this work was also supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea Government (MSIT) (2021-0-00188, Open Source Development and Standardization for AI Enabled IoT Platforms and Interworking).

References

[1] R. Maull, P. Godsiff, C. Mulligan, A. Brown, and B. Kewell, "Distributed ledger technology: applications and implications," *Strategic Change*, vol. 26, no. 5, pp. 481–489, 2017.

[2] L. Hang and D.-H. Kim, "SLA-based sharing economy service with smart contract for resource integrity in the Internet of Things," *Applied Sciences*, vol. 9, no. 17, p. 3602, 2019.

[3] F. Jamil, L. Hang, K. Kim, and D. Kim, "A novel medical blockchain model for drug supply chain integrity management in a smart hospital," *Electronics*, vol. 8, no. 5, p. 505, 2019.

[4] L. Hang, E. Choi, and D.-H. Kim, "A novel EMR integrity management based on a medical blockchain platform in hospital," *Electronics*, vol. 8, no. 4, p. 467, 2019.

[5] L. Hang and D.-H. Kim, "Design and implementation of an integrated IoT blockchain platform for sensing data integrity," *Sensors*, vol. 19, no. 10, p. 2228, 2019.

[6] Y. Lu, Y. Qi, S. Qi, Y. Li, H. Song, and Y. Liu, "Say no to price discrimination: decentralized and automated incentives for

price auditing in ride-hailing services," *IEEE transactions on Mobile computing*, vol. 21, no. 2, 2022.

[7] Y. Lu, J. Zhang, Y. Qi et al., "Accelerating at the edge: a storage-elastic blockchain for latency-sensitive vehicular edge computing," *IEEE transactions on intelligent transportation systems*, pp. 1–15, 2021.

[8] Y. Lu, J. Zhang, Y. Qi et al., "Safety warning! Decentralised and automated incentives for disqualified drivers auditing in ride-hailing services," *IEEE Transactions on Mobile Computing*, vol. 1233, no. c, p. 1, 2021.

[9] U. Bodkhe, S. Tanwar, K. Parekh et al., "Blockchain for Industry 4.0: a comprehensive review," *IEEE Access*, vol. 8, pp. 79764–79800, 2020.

[10] W. Gao, W. G. Hatcher, and W. Yu, "A survey of blockchain: techniques, applications, and challenges," in *2018 27th International Conference on Computer Communication and Networks (ICCCN)*, pp. 1–11, Hangzhou, China, 2018.

[11] S. Wang, L. Ouyang, Y. Yuan, X. Ni, X. Han, and F. Wang, "Blockchain-enabled smart contracts: architecture, applications, and future trends," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, no. 11, pp. 2266–2277, 2019.

[12] S. Nakamoto and A. Bitcoin, *A Peer-To-Peer Electronic Cash System*, Bitcoin, 2008, https://bitcoin.org/bitcoin.pdf.

[13] S. Rouhani and R. Deters, "Performance analysis of Ethereum transactions in private blockchain," in *2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS)*, pp. 70–74, Beijing, China, 2017.

[14] R. Henry, A. Herzberg, and A. Kate, "Blockchain access privacy: challenges and directions," *IEEE Security & Privacy*, vol. 16, no. 4, pp. 38–45, 2018.

[15] C. V. Helliar, L. Crawford, L. Rocca, C. Teodori, and M. Veneziani, "Permissionless and permissioned blockchain diffusion," *International Journal of Information Management*, vol. 54, article 102136, 2020.

[16] S. Bergman, M. Asplund, and S. Nadjm-Tehrani, "Permissioned blockchains and distributed databases: a performance study," *Concurrency and Computation: Practice and Experience*, vol. 32, no. 12, article e5227, 2020.

[17] C. Cachin, "Architecture of the hyperledger blockchain fabric," *Workshop on distributed cryptocurrencies and consensus ledgers*, vol. 310, no. 4, pp. 1–4, 2016, https://allquantor.at/blockchainbib/pdf/cachin2016architecture.pdf.

[18] Q. Nasir, I. A. Qasse, M. Abu Talib, and A. B. Nassif, "Performance analysis of hyperledger fabric platforms," *Networks*, vol. 2018, pp. 1–14, 2018.

[19] E. Androulaki, A. Barger, V. Bortnikov et al., "Hyperledger fabric: a distributed operating system for permissioned blockchains," in *Proceedings of the thirteenth EuroSys conference*, Porto, Portugal, 2018.

[20] J. Garay, A. Kiayias, and N. Leonardos, "The bitcoin backbone protocol: analysis and applications," in *Advances in Cryptology-Eurocrypt 2015. Eurocrypt 2015*, E. Oswald and M. Fischlin, Eds., Springer, Berlin, Heidelberg, 2015.

[21] M. Kuzlu, M. Pipattanasomporn, L. Gurses, and S. Rahman, "Performance analysis of a hyperledger fabric blockchain framework: throughput, latency and scalability," in *2019 IEEE International Conference on Blockchain (Blockchain)*, pp. 536–540, Atlanta, GA, USA, 2019.

[22] S. Shalaby, A. A. Abdellatif, A. Al-Ali, A. Mohamed, A. Erbad, and M. Guizani, "Performance evaluation of hyperledger fabric," in *2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies (ICIoT)*, pp. 608–613, Doha, Qatar, 2020.

[23] C. Wang and X. Chu, "Performance characterization and bottleneck analysis of hyperledger fabric," in *2020 IEEE 40th International Conference on Distributed Computing Systems (ICDCS)*, pp. 1281–1286, Singapore, Singapore, 2020.

[24] L. Liu, J. Feng, Q. Pei et al., "Blockchain-enabled secure data sharing scheme in mobile-edge computing: an asynchronous advantage actor–critic learning approach," *IEEE Internet of Things Journal*, vol. 8, no. 4, pp. 2342–2353, 2021.

[25] J. Yin, Y. Xiao, Q. Pei et al., "SmartDID: a novel privacy-preserving identity based on blockchain for IoT," *IEEE Internet of Things Journal*, 2022.

[26] J. Feng, F. R. Yu, Q. Pei, J. Du, and L. Zhu, "Joint optimization of radio and computational resources allocation in blockchain-enabled mobile edge computing systems," *IEEE Transactions on Wireless Communications*, vol. 19, no. 6, pp. 4321–4334, 2020.

[27] C. Chen, C. Wang, T. Qiu, N. Lv, and Q. Pei, "A secure content Sharing scheme based on blockchain in vehicular named data networks," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 5, pp. 3278–3289, 2020.

[28] C. Fan, S. Ghaemi, H. Khazaei, and P. Musilek, "Performance evaluation of blockchain systems: a systematic survey," *IEEE Access*, vol. 8, pp. 126927–126950, 2020.

[29] P. J. Hee and C. S. Hong, *Blockchain Architecture Using Consensus Algorithm with Adjustable Validation and Its Performance Improvement*, The Korean Institute of Information Scientists and Engineers, 2019.

[30] H. Lee, C. Yoon, S. Bae et al., "Multi-batch scheduling for improving performance of hyperledger fabric based IoT applications," in *2019 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, Waikoloa, HI, USA, 2019.

[31] J.-W. Lee and S. Park, "A study on performance improvement of hyperledger fabric through batched chaincode message," in *2020 21st Asia-Pacific Network Operations and Management Symposium (APNOMS)*, pp. 259–262, Daegu, Korea (South), 2020.

[32] F. Lu, L. Gan, Z. Dong, W. Li, H. Jin, and A. Y. Zomaya, "A cache enhanced endorser design for mitigating performance degradation in hyperledger fabric," in *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, pp. 1001–1006, Halifax, NS, Canada, 2018.

[33] J. Sousa, A. Bessani, and M. Vukolic, "A byzantine fault-tolerant ordering service for the hyperledger fabric blockchain platform," in *2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, pp. 51–58, Luxembourg, Luxembourg, 2018.

[34] H. Javaid, C. Hu, and G. Brebner, "Optimizing validation phase of hyperledger fabric," in *2019 IEEE 27th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS)*, pp. 269–275, Rennes, France, 2019.

[35] C. Gorenflo, S. Lee, L. Golab, and S. Keshav, "FastFabric: scaling hyperledger fabric to 20 000 transactions per second," *International Journal of Network Management*, vol. 30, no. 5, article e2099, 2020.

[36] T. Nakaike, Q. Zhang, Y. Ueda, T. Inagaki, and M. Ohara, "Hyperledger fabric performance characterization and optimization using GoLevelDB benchmark," in *2020 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, pp. 1–9, Toronto, ON, Canada, 2020.

[37] T. Miyamae, T. Honda, M. Tamura, and M. Kawaba, "Performance improvement of the consortium blockchain for financial business applications," *Journal of Digital Banking*, vol. 2, no. 4, pp. 369–378, 2018.

[38] J. Kim, K. Lee, G. Yang, K. Lee, J. Im, and C. Yoo, "QiOi: performance isolation for hyperledger fabric," *Applied Sciences*, vol. 11, no. 9, p. 3870, 2021.

[39] P. Thakkar and S. Natarajan, "Scaling hyperledger fabric using pipelined execution and sparse peers," 2020, https://arxiv.org/abs/2003.05113.

[40] E. H. Mamdani and S. Assilian, "An experiment in linguistic synthesis with a fuzzy logic controller," *International Journal of Man-Machine Studies*, vol. 7, no. 1, pp. 1–13, 1975.

[41] L. Hang and D.-H. Kim, "Optimal blockchain network construction methodology based on analysis of configurable components for enhancing hyperledger fabric performance," *Blockchain: Research and Applications*, vol. 2, no. 1, p. 100009, 2021.

[42] Hyperldger Caliper Documentationhttps://github.com/hyperledger/caliper.

[43] L. Hang, B. H. Kim, K. H. Kim, and D. H. Kim, "A permissioned blockchain-based clinical trial service platform to improve trial data transparency," *BioMed Research International*, vol. 2021, Article ID 5554487, 22 pages, 2021.

Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 4009078, 13 pages
https://doi.org/10.1155/2022/4009078

Research Article

Blockchain and UAV-Enabled Signal Source Identification with Edge Computing and Wireless Signal-Aerial Image Fusion

Jian Xiao,[1,2] **Peng Liu**,[1,2] **Huaming Lin**,[3] **Hangxiang Fang**,[4] **Jiayi Xu**,[1] **Hangguan Shan**,[4] **Haoji Hu**,[4] **Yi Huang**,[3] **and Huijuan Lu**[2]

[1]*School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou, China 310018*
[2]*Key Laboratory of Electromagnetic Wave Information Technology and Metrology of Zhejiang Province, College of Information Engineering, China Jiliang University, Hangzhou 310018, China*
[3]*Hangzhou Security and Technology Evaluation Center, Hangzhou, China 310020*
[4]*College of Information Science and Electronic Engineering, Zhejiang University, Hangzhou, China 310027*

Correspondence should be addressed to Huaming Lin; lhm18@live.cn

Received 10 January 2022; Accepted 8 March 2022; Published 24 March 2022

Academic Editor: Celimuge Wu

This paper proposes a novel signal source identification system composed of unmanned aerial vehicles (UAVs) and a blockchain, in which the identification method makes full use of binocular camera data and received signal strength. The UAV tasks are organized by using blockchain technology and smart contracts. To tackle the challenge that the transmit power of the object and the channel path loss coefficient are unknown to the UAV, a maximum likelihood estimation method is developed to estimate the parameters in the path loss log-normal shadowing model. Then, the mean squared error is used as the metric to distinguish the signalling object. The simulation results show that the proposed method can effectively complete the task. Also, a mobile edge computing- (MEC-) enabled UAV testbed system is designed and implemented in real environment. The system works accurately where the number of candidate objects is 3.

1. Introduction

Radio signal source identification provides essential support in many applications [1–3]. However, except the basic information of the object (such as shape and outline), the parameters of the signal are usually unknown. Therefore, in a complex electromagnetic environment, it needs to first screen out the potential targets and then identify the one which is sending signals. The traditional ways of object detection and radio direction finding are easily constrained by their own mobility and rely on manual operation and control. Therefore, to improve the efficiency, accuracy, and reliability of the identification, as well as reduce the cost, an integrated and shareable aerial platform is required.

Technology advances in unmanned aerial vehicles (UAVs, also known as drones) have enabled them to carry multiple types of onboard sensors. Cameras are one kind of sensors that have been commonly used. A UAV carrying cameras can provide a clearer view from the air than ground while allowing flexible movement. Cameras carried on UAVs can be classified into many types, such as single camera [4], multiple cameras [5], and spectral cameras [6]. A binocular camera is a single camera with two lenses and is capable of outputting both graphs and depth information, thus arousing interest in UAV applications, such as UAV navigation [7], obstacle avoidance [8], and localization [9]. Furthermore, road detection [10] and photogrammetric measurements [11] can be also solved with UAV binocular vision.

Meanwhile, the reduced cost and weight of Software-Defined Radio (SDR) equipment make it feasible for UAVs to carry an SDR onboard. The application fields of UAVs carrying SDRs can then be extended to wildlife tracking, search, rescue, etc. For example, in [12], two SDR-based UAV-assisted wildlife tracking methods are presented: one is based on four single-stage antennas using the Doppler effect and the other one using Yagi omnidirectional antennas for

signal reach angle. Ref. [13] presents a UAV system carrying VHF telemetry equipment that includes a hexacopter UAV, an onboard computer, an SDR receiver, a directional antenna, and a control laptop on the ground. In paper [14], to search for survivors after a disaster, the authors use the UAV carrying an SDR as a GSM base station to receive signals sent by the target's GSM device and locate the target by the received signal strength and UAV coordinates.

As UAVs are getting more powerful and smarter nowadays [15], automatic UAV-assisted radio source detection becomes a potential solution and a hot research topic. An automatic UAV trajectory planning method for ground object localization is proposed in [16]. The solution locates ground objects by received signal strength and automatically finds paths with high localization accuracy and low energy consumption by reinforcement learning. A method for locating illegal base stations via received signal power in the case of unknown channel model and unknown noise model is proposed in [17]. It takes advantage of directional antennas and controls the UAV by Q-learning algorithms. However, all the above methods have only been validated in numerical simulations. In reality, the real-time performance of the system, due to heavy data transmission stress and computation demands, can be a major challenge to be settled in field experiments.

Considering a practical commercial application scenario, a user may not afford such a UAV fleet and it is also not necessary. Therefore, UAVs may be from a heterogeneous origin and form a computing paradigm of multiple autonomous agents. In the meantime, the air-ground integrated edge computing system has been put into practice [18]. In such an architecture, how to ensure secure communication and cooperation between multiple parties becomes a key problem. Blockchain [19, 20] has been proven to be very useful in various IoT applications. In literature [21], the authors describe a method of organizing the communication protocol, which allows agents of the multiagent system (MAS) to make decisions about their actions. The paper shows how to implement an autonomous economic system with UAVs and organize a communication system among agents in a peer-to-peer network using the decentralized Ethereum blockchain technology and smart contracts. "BUS" proposed in [22] is a UAV swarm-assisted data acquisition scheme in which data is collected from IoT devices via a UAV swarm and then stored in the nearest server with the assistance of blockchain. A smart contract is employed to handle the IoT devices and missions in BUS. In another work, the selection of the UAV for the desired quality of network coverage and the development of a distributed and autonomous real-time monitoring framework for the enforcement of service-level agreement (SLA) are introduced [23]. It builds a novel blockchain architecture that relies on machine learning techniques to monitor and penalize UAVs that violate SLA. In [24], the authors propose a new type of blockchain to resolve critical message dissemination issues in a vehicular ad hoc network (VANET), which can be used as a reference in our work.

In this paper, partly motivated by [25], we design and implement a mobile edge computing- (MEC-) enabled UAV system equipped with onboard SDR and binocular camera, one for each. A signal source identification algorithm that fuses the visual depth information and received power strength is proposed. The organization of tasks of UAVs is enabled by blockchain. Our method can work in situations where the objects' transmit power and channel parameters are unknown.

The rest of this paper is organized as follows. Section 2 briefly introduces the overall architecture of the system. Blockchain preliminaries are presented in Section 3. In Section 4, the system model, the four major parts, the formulation of the problem, and the design of the algorithm are described in detail. Section 5 gives the experimental results, including the results of the semiphysical simulation and the real-world experiment. Finally, Section 6 concludes the paper.

2. System Architecture

The proposed scenario consists of a target area, a hiding signal source, various UAVs, some users, a ground access point, edge computing servers, network services, and a blockchain, as shown in Figure 1. A user would like to make an order for signal source identification. With the help of the servers, a smart contract is generated with the order data (the purpose of the order, attribute data, participators, etc.) and then transferred to the blockchain. Any unoccupied UAVs can accept the contract which contains information about the task.

Then, the UAV makes a scheduled flight and informs the user about the result of the task. During the flight, the data collected by the UAV will be sent to the edge servers for analysis. After returning to the base, the UAV notifies the servers that the order is completed. This ensures the security and reliability of the proposed system.

3. Blockchain Preliminaries

The blockchain is essentially a distributed public ledger, where each transaction is recorded in a block. Each block is identified by its hash. Each block not only contains the content of the transaction and timestamp but also references the hash of its previous block. These blocks are linked by reference hash and then superimposed into a "chain" in chronological order to create a blockchain. A blockchain network is a peer-to-peer network composed of a group of nodes. Each node operates the same blockchain through its own copy. The blockchain is a distributed data structure that is copied and shared among network members. We assume that each user conducts transactions on the network through miners (i.e., nodes). The literature [26] summarizes that the operations of nodes in a blockchain network follow the below steps:

(1) The user interacts with the blockchain through a pair of private/public keys. When initiating a transaction, the transaction initiator needs to sign the transaction with a private key, and those transactions are addressable on the network through the initiator's public key. When a new transaction is generated, it

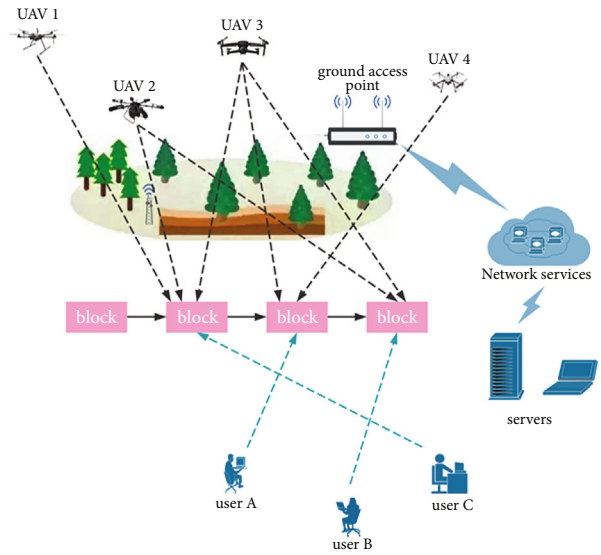


Figure 1: System architecture.

will be broadcast to other participating nodes in the blockchain network

(2) After the transaction is broadcast to the entire network, within an agreed time interval, each node will collect several unverified transactions and hash them into a time-stamped block. Each block can contain hundreds or thousands of transactions

(3) Each node performs a Proof-of-Work (PoW) calculation equivalent to solving a mathematical problem to determine who can verify the transaction. Letting the fastest node in calculation do the job is the way to achieve consensus. The node with the fastest PoW calculation will propagate its own block to other nodes. This calculation process is a "mining" process to derive the effective hash of the new block

(4) The node that obtained the verification right broadcasts the block to all nodes, and other nodes will confirm whether the transactions contained in this block are valid. If valid, add this block to the blockchain and apply the transactions it contains to update the world view of all nodes. Otherwise, this block will be discarded. Once the block is accepted by the blockchain, the transactions contained in the block are part of the blockchain and cannot be changed in any way

(5) Once all nodes accept the block, the blocks that have not previously completed the PoW work will be invalidated, and each node will reestablish a block to continue the next PoW calculation work

In essence, a smart contract refers to a piece of conditional statement code that runs on the blockchain. When two parties of a smart contract generate an asset transaction on the blockchain, this piece of code is triggered to automatically complete the specific transaction process. Smart contracts are special accounts on the blockchain, which contain addresses, balances, status, and codes. The address is a unique identifier for the account, just like a regular user account. The smart contract works as follows:

(1) Construction of smart contracts: smart contracts are jointly formulated by multiple users in the blockchain and can be used for any transaction behavior between any users. A contract clearly stipulates the rights and obligations of both parties to the transaction electronically. The code contains conditions that will trigger the automatic execution of the contract. All possible results of the contract should be described in the smart contract

(2) Storage smart contracts: once the coding is completed, the smart contract is uploaded to the blockchain network; that is, all nodes on the entire network can receive the contract

(3) Executing smart contracts: the smart contract will periodically check whether there are related events and trigger conditions and push the events that meet the conditions to the queue to be verified. The verification node on the blockchain first performs signature verification on the event. After most nodes reach a consensus on the event, the smart contract will be successfully executed and the users will be notified. The execution result of a smart contract must be deterministic; that is, the same input always produces the same output. Because all interactions with the contract are performed through signed messages on the blockchain, all network participants can obtain the cryptographically verifiable trace of the contract's operations

4. System Model

The architecture of the proposed blockchain-based signal source identification system is shown in Figure 2. There are three roles: miners, initiators, and participants. In the system, participants have backbone ground edge units to support UAV manipulating, data processing, negotiating with miners, and transmitting data to initiators.

Miner: the *miner* is a trusted and authenticated node that has multiple roles. First, it acts as a broker between initiators and participants. It accepts requests from initiators and matches them with participants. Second, it is a maintainer of the stable operations in the blockchain system which packages all kinds of data and smart contracts onto the blockchain. There could be a few miners, such as servers of the different public third-party platforms, in a real scenario.

Initiator: the *initiator* refers to the users in Section 2. The object detection or signal source identification task is launched by the initiator. Then, the initiator sends this task to the miner and waits for the assignment and draft smart contract. Smart contract and data related to the transaction will be handed to the miner to write onto the blockchain. Detection data will be returned by participants directly.

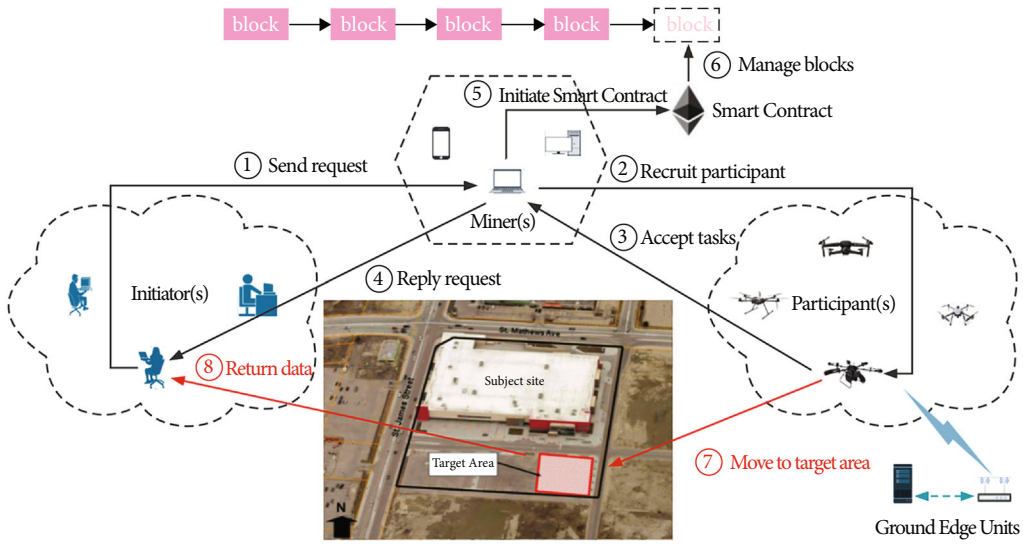


Figure 2: The proposed system model.

Finally, the initiators make payment through the blockchain by using digital currency.

Participant: the *participant* refers to the UAV. It is controlled by a ground edge unit. The participant can evaluate the recruitment from the miner and accept the task. It reads the parameters related to the object identification task and the smart contract from the blockchain. Then, it flies to the target area and sends the computation results to the initiator.

After matching and agreement on the smart contract, the object detection and signal source identification task is technically carried out in three major steps, i.e., visual-based object detection and tracking, binocular depth estimation, and wireless signal-aerial image fusion-based signal source identification. The flow chart of the method is shown in Figure 3.

4.1. Blockchain and Smart Contract. The design goals of introducing blockchain and smart contract lie in two aspects:

4.1.1. Reliable Collaboration. We need to ensure the authenticity and reliability of initiators and participants. To this end, the blockchain guarantees that the synchronization and consistency of quality of service (QoS) of each UAV, and the payment is correctly made by the user.

4.1.2. Security and Privacy. We assume that the initiator and participants are all semihonest nodes who obey the agreement and honestly execute the tasks. However, they may want to probe into others' data, either individually or collusively. On the other hand, both sides do not want to expose their identities. Therefore, the proposed blockchain-based architecture combines secure schemes such as asymmetric key cryptography, ring signature, and consensus mechanism. All the operations are performed in a privacy-preserving manner.

According to Figure 2, there are totally eight steps for the whole process.

Step 1. The initiator sends its task requirement and remuneration offering to the miner which is encrypted by its private key and with ring signature.

Step 2. The miner decrypts the requirement and recruits the participant with an optimal matching algorithm according to predefined conditions. The offer is sent with the encryption of the miner's private key.

Step 3. The participant decrypts the offer and chooses to accept or deny it in his own interest. If accepted, its parameters and the public key will be encrypted with the miner's public key and then sent back to the miner.

Step 4. The miner then answers the initiator with the information of the participant in an encryption manner.

Step 5. Then, a smart contract between the initiator and the participant is created in both parties' confirmation.

Step 6. The smart contract is deployed in the blockchain framework. The miner also builds a secure channel and provides a pair of keys for data exchange of both parties.

Step 7. After receiving the parameters of the task from the initiator (a user) via the secure channel, the participant (a UAV) flies to the target area and performs the identification task.

Step 8. When the task is completed, the result will be returned to and validated by the initiator via the secure channel. Finally, the payment is made using digital currency according to the smart contract.

4.2. Object Detection and Object Tracking. The visual target detection and tracking part mainly complete the subtask of tracking and identification of specific types of targets within the UAV field of view. Target detection is one of the key technologies to improve the perception capability of UAVs and is of great significance. Compared with the traditional methods based on manual features, the deep learning methods based on convolutional neural networks have powerful feature learning and expression capabilities and become the mainstream algorithms for target detection tasks at present [27].

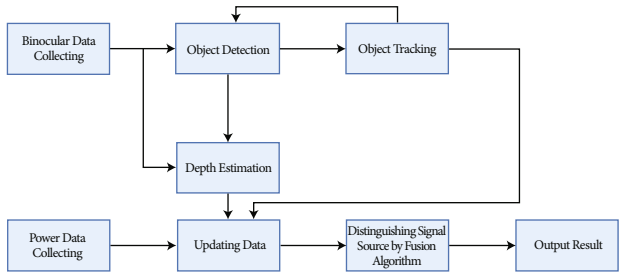


FIGURE 3: The flow chart of the wireless signal-aerial image fusion-based signal source identification.

Aerial photography generally has the following characteristics because the imaging perspective is different from natural scene images.

(i) Complex background

(ii) Small targets

(iii) Large field of view

(iv) Rotation

Inspired by [27], the visual target detection part adopts the YOLO-based target detection framework [28], and on this basis, the algorithm is optimized and adjusted based on the characteristics of UAV data collection and the difficulties of recognition, via the drone vehicle datasets [29] and DOTA datasets [30], as shown in Figure 4. For the target tracking part, the Deep SORT algorithm [31] is used to predict the movement probability of the object and calculate the difference between the front and back frame features of the object and then complete the continuous tracking of the target.

We perform data augmentation for both datasets of images. Data augmentation is a technique that artificially extends the training dataset by allowing limited data to produce more equivalent data. It is an effective means to overcome the shortage of training data. We mainly use random rotation, color transformation, blurring, noise injection, and hybrid image processing to enhance the diversity of the input data. Then, the YOLO pretraining network is trained using the enhanced data to fine-tune the weights of the convolutional layer parameters and migrate the object recognition task from the ground view to the object recognition task in the UAV view.

In the overall system, the signal source identification algorithm needs to take the visual detection result and the corresponding object ID as input; thus, we add a tracking algorithm in addition to YOLO's object recognition to ensure that each detected object corresponds to its ID. As an algorithm commonly used in Multiobject Tracking (MOT), Deep SORT is a detection-based tracking method of good performance and high industrial interest. The main process of the MOT algorithm is as follows:

(1) Given the original video frame, run a target detector such as YOLO for detection and obtain target detection frames

(2) Take out the interested targets in all target frames and perform feature extraction (including apparent features and motion features)

(3) Perform similarity calculation to calculate the matching degree between the targets in the front and back frames (the distance between the features belonging to the same target before and after is relatively small, and the distance between different targets is large)

(4) Associate the data, and assign the ID of the target to each object

4.3. Binocular Depth Estimate. The binocular depth estimate part is to complete the subtask of estimating the depth to the identified object, i.e., the distance of a UAV equipped with a binocular camera to the identified target object.

In this work, we choose the SGBM algorithm to estimate the depth between objects and the camera [32]. Firstly, the binocular disparity d measures the horizontal disparity of the surface point on the target object between the video frame taken by the left camera x_L and the right camera x_R is estimated by energy optimization [33] as shown in

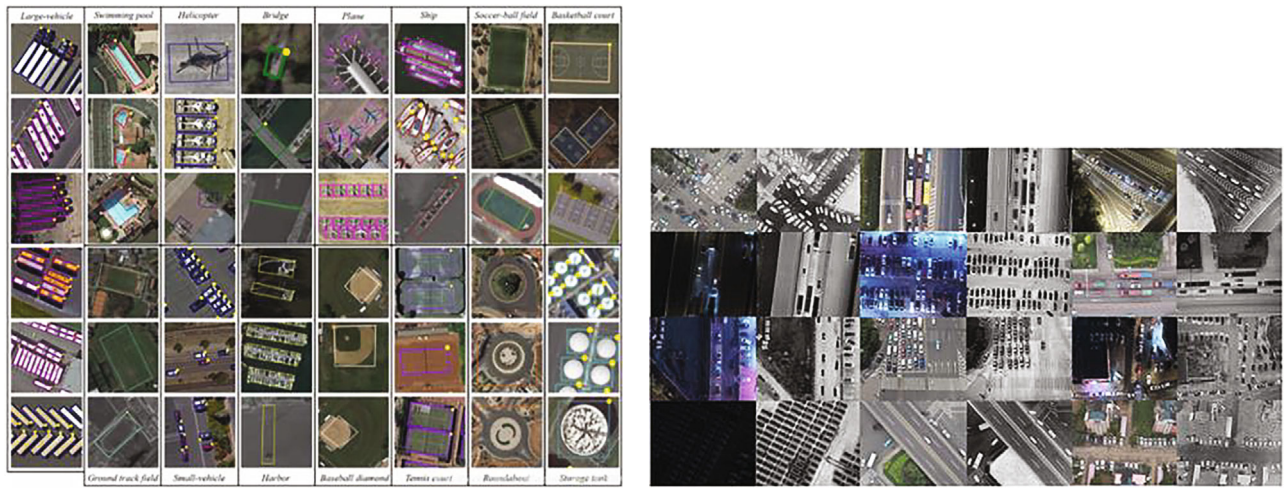
$$d = |x_L - x_R|. \tag{1}$$

Then, the obtained disparity map is converted into a depth map based on the relationship between disparity and depth. Finally, the distance from the target to the camera is extracted from the depth map. More specifically, the disparity value of each pixel is calculated with respect to the right eye view using the left eye view as the reference. We construct an energy function $E(D)$ to estimate the optimal disparity image D by minimizing the energy value using the following equation:

$$E(D) = \sum_p C(p, D_p) + \sum_{q \in N_p} P_1 I\left[\left|D_p - D_q\right| = 1\right] + \sum_{q \in N_p} P_2 I\left[\left|D_p - D_q\right| > 1\right]. \tag{2}$$

The first term is the sum of matching cost for every pixel in the disparity image D corresponding to the captured left video frame. $C(p, D_p)$ is the pixel-wise matching cost function with disparity D_p corresponding to the minimum cost for pixel p. In the current implementation, the matching cost value is calculated based on Mutual Information (MI) [34]. The second term adds a constant penalty P_1 for all pixels q in the neighborhood N_p of p, if the disparity between p and q is 1, while the third term adds a larger constant penalty P_2 if the disparity between p and q is larger than 1. Operator $I[X]$ equals 1 if event X occurs; otherwise, it equals 0. Constant P_2 is chosen larger than P_1.

Based on the mapping between disparity and depth value, the depth image according to the left-view video frame can be evaluated. The depth information of the surface point on the target object using disparity value is computed as follows:



(a) (b)

Figure 4: Two datasets for UAV object detection: (a) from drone vehicle datasets; (b) from DOTA datasets.

$$Z = \frac{b \cdot f}{d}, \tag{3}$$

where b is the baseline of the binocular camera and f is the camera focal length.

To obtain the distance between the target object and the camera, the bounding box of the object in the video frame needs to be computed based on the object detection and tracking algorithm. The pixels within the bounding box are sampled at an interval of 2 to 5 pixels, and the depth value is indexed from the corresponding depth map. The average depth value of the sampled pixels is then used to represent the actual distance from the object to the camera.

4.4. Fusion Object Detection. So far, we have found a number of suspicious objects distributed in the area, which are similar in appearance. Suppose that only one of them is the real object which is communicating with the outside world at a fixed wireless frequency band, and we need to identify it. Some existing works utilize a wireless network between UAVs and objects to help localization [35, 36]. However, in both cases, the information of the target is known beforehand. The wireless connection is only used to transmit data, power, and arrival of time for distance calculation. To this end, the proposed method utilizes a UAV equipped with a binocular camera to obtain the distances between the object and the UAV at some measurement points of the trajectory, by applying binocular depth estimation after using deep learning-based target detection. At the same time, the received power is obtained utilizing the UAV onboard SDR. As shown in Figure 5, the UAV moves simultaneously along the path until the desired data has been collected.

Let S denote the set of objects that the UAV detected in the flight area. Arbitrarily select an object $s \in S$ and take it as the source target sending signals. Then, let the received power p_n measured by the UAV at position n be as same as p_{sn} (assume that only s is the transmitting signal and note that s is anonymous to the UAV) and the distance estimated

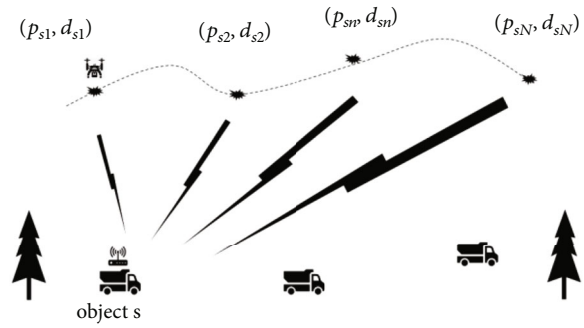


Figure 5: Problem formulation of signal source identification.

by binocular depth estimation at position n be d_{sn}. The two types of data collected during the flight have a correspondence such that the received power from the object s measured at N positions constitutes a vector $\mathbf{p}_s$, and the binocular depth estimated distance of the object s measured at N positions constitutes a vector $\mathbf{d}_s$, as shown in

$$\mathbf{d}_s = [d_{s1}, d_{s2}, \cdots, d_{si}, \cdots, d_{sN}]^T, \tag{4}$$

$$\mathbf{p}_s = [p_{s1}, p_{s2}, \cdots, p_{si}, \cdots, p_{sN}]^T. \tag{5}$$

Consider that the received wireless signal power satisfies the path loss log-normal shadowing model [37].

$$p = p_0 - 10\alpha \lg \frac{d}{d_0} + X_\sigma, \tag{6}$$

where p is the received power in dB at distance d from the object, p_0 is the received power in dB at distance d_0 from the object, α is the path loss exponent, and X_σ is the environment-dependent shadow fading coefficient which obeys the Gaussian distribution with zero mean and variance σ^2.

To determine whether object s is sending signals, the distance vector $\mathbf{d}_s$ and power vector $\mathbf{p}_s$ of object s can be substituted into the path loss log-normal shadowing model

to determine if they match the model well. In the case that the parameters p_0 and α in the log-normal model are unknown, they (when s is assumed to be the target object) need to be calculated first. By utilizing maximum likelihood estimation, the procedure for evaluating the parameters of the path loss log-normal shadowing model for object s is as follows.

Let the reference distance d_0 in the path loss log-normal shadowing model be equal to 1 m, implying that p_0 is the received power at a reference distance of 1 m. Since p is subject to Gaussian distribution with mean $p_0 - 10\alpha \lg d$ and variance σ^2, we can define the following likelihood function:

$$L(p_0, \alpha, \sigma^2) = \prod_{n=1}^{N} \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{ -\frac{[p_{sn} - (p_0 - 10\alpha \lg d_{sn})]^2}{2\sigma^2} \right\}. \tag{7}$$

Taking the likelihood function logarithmically, we can obtain

$$\ln L(p_0, \alpha, \sigma^2) = N \ln \frac{1}{\sqrt{2\pi}\sigma} - \sum_{n=1}^{N} \frac{[p_{sn} - (p_0 - 10\alpha \lg d_{sn})]^2}{2\sigma^2}. \tag{8}$$

Then, to calculate the maximum likelihood estimate of the parameters, we can derive equation (8) and find p_0, α, and σ^2, such that the partial derivatives are zero, i.e.,

$$\frac{\partial \ln L}{\partial p_0} = -\sum_{n=1}^{N} \frac{[p_{sn} - (p_0 - 10\alpha \lg d_{sn})]}{\sigma^2} = 0, \tag{9}$$

$$\frac{\partial \ln L}{\partial \alpha} = \lg d_n \sum_{n=1}^{N} \frac{[p_{sn} - (p_0 - 10\alpha \lg d_{sn})]}{\sigma^2} = 0, \tag{10}$$

$$\frac{\partial \ln L}{\partial \sigma^2} = -\frac{1}{2} N \frac{1}{\sigma^2} + \frac{1}{2} \frac{1}{\sigma^4} \sum_{n=1}^{N} [p_{sn} - (p_0 - 10\alpha \lg d_{sn})]^2 = 0. \tag{11}$$

After calculation, we can obtain the following estimates for p_0, α, and σ^2:

$$\widehat{p}_0 = \bar{p} + 10\widehat{\alpha}\bar{d}_{sn}, \tag{12}$$

$$\widehat{\alpha} = \frac{\sum_{n=1}^{N} -1/10(\lg d_{sn} - \bar{d}_{sn})(p_{sn} - \bar{p})}{\sum_{n=1}^{N} (\lg d_{sn} - \bar{d}_{sn})^2}, \tag{13}$$

$$\widehat{\sigma^2} = \frac{1}{N} \sum_{n=1}^{N} [p_{sn} - (\widehat{p}_0 - 10\widehat{\alpha} \lg d_{sn})]^2, \tag{14}$$

where

$$\bar{p} = \frac{1}{N} \sum_{n=1}^{N} p_{sn}, \tag{15}$$

$$\bar{d}_{sn} = \frac{1}{N} \sum_{n=1}^{N} \lg d_{sn}. \tag{16}$$

Using the parameters obtained from the maximum likelihood estimation $\widehat{p}_0$ and $\widehat{\alpha}$, we can obtain the fitted value of the received power $\widehat{p}_{sn}$ according to equation (6) as

$$\widehat{p}_{sn} = \widehat{p}_0 - 10\widehat{\alpha} \lg d_{sn}. \tag{17}$$

To measure the extent to which object s conforms to the path loss log-normal shadowing model, we choose the mean squared error (MSE) of the fitted received power and the measured received power as the metric. Then, the MSE of object s is defined as

$$\mathrm{MSE}_s = \frac{1}{N} \sum_{n=1}^{N} (\widehat{p}_{sn} - p_{sn})^2. \tag{18}$$

After the UAV detects all objects in the flight area, it compares the MSE of all objects and identifies the object with the smallest MSE as the source target, i.e.,

$$\text{Source target} = \arg \min_{s \in S} \mathrm{MSE}_s. \tag{19}$$

In summary, the fusion-based signal source identification algorithm is shown in Algorithm 1.

4.5. Hardware Implementation. We also build a testbed to evaluate the system performance. The hardware component list is described in Table 1.

The experiment system consists of two major parts, the air system and the ground system, as seen in Figure 6. The air system contains various UAVs (in our case, hexacopter UAVs loaded with Raspberry 4Pi B model, which links the binocular camera and the SDR by a USB cable and communicates with the ground system through Wi-Fi). The ground system includes a laptop with a discrete GPU, in which the costly computations are conducted, and a Wi-Fi AP to connect all devices through wireless communications. The laptop is also responsible for simulating the miner in the blockchain network. We use smartphones to play the role of the initiator. All tasks are launched from smartphones. The UAV system is shown in Figure 7. The Raspberry Pi 4 and SDR are mounted in the UAV, and the camera is tied in angle to the ground.

5. Result and Analysis

We used semiphysical simulation to verify the effectiveness of the fusion algorithm. First, we used SDR on the ground to collect power data at different positions, and then, we calculated the path loss coefficient of the actual channel and the received power at a reference distance of 1 m by linear regression. As shown in Figure 8, the object was a USRP B210 device from Ettus Corp, and diagrams for transmitting were written in GNU Radio software. The waveform of the signal was a periodic sine waveform, and the transmitter frequency was set to 907 MHz-927 MHz, the power was set

Input: objects set S=Ø, initiate corresponding distance data vector d_s=Ø, power data vector p_s=Ø
Output: signal source target
1. while UAV keeps flying do
2. if new object *s'* is detected then
3. do S = S ∪{*s'*}
4. end if
5. obtain current power *p* from SDR measurement
6. for *s* in S do
7. obtain current distance *d* of object *s* from binocular depth estimate
8. if *d* and *p* is valid then
9. update *d* and *p* to the end of d_s and p_s
10. end if
11. end for
12. if data achieves certain amounts then
13. target = *None*
14. for *s* in S do
15. calculate $\hat{p}_0$ and $\hat{\alpha}$ of object *s*
16. calculate MSE_s
17. update target by comparing *MSE*
18. end for
19. end if
20. end while

Algorithm 1: Fusion-based signal source identification.

Table 1: Hardware components.

Component		Parameter
UAV air system	Computer	Raspberry Pi 4B
	Camera	Metoak binocular camera
	UAV	Hexacopter Tarot UAV
	SDR	Hackrf One
Ground MEC system	Laptop	Lenovo Legion 7
	Wi-Fi	TP-LINK war1200

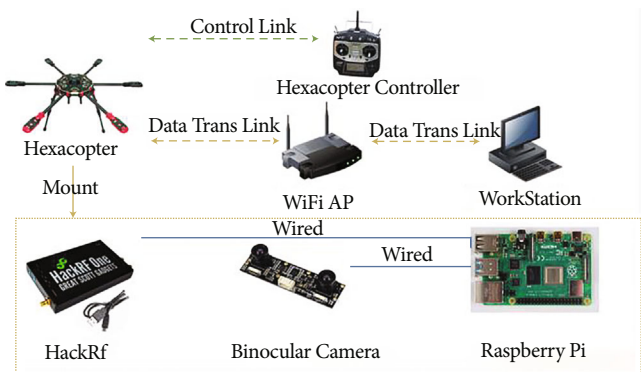


Figure 6: Experiment system architecture. The dashed double arrow lines show the communications between devices.

to 1 W, and the transmitting gain of the antenna was 50 dB. The object was placed at 1.7 m from the ground to simulate the signal propagation in the air as high as possible. We measured the data at each 1 m interval on the ground, and the distance range from the source was from 1 m to 30 m. The experiment was repeated 10 times, and the total received power values were obtained for 300 locations. Figure 9 shows the measured power and distance. After calculation, we obtained $p_0 = -28.91$ dBm, $\alpha = 3.581$, and $\sigma^2 = 23.49$.

5.1. Security Analysis. In our system, all participants and initiators are assumed to be semihonest nodes. Based on this assumption, we are going to analyze the security objectives in this section.

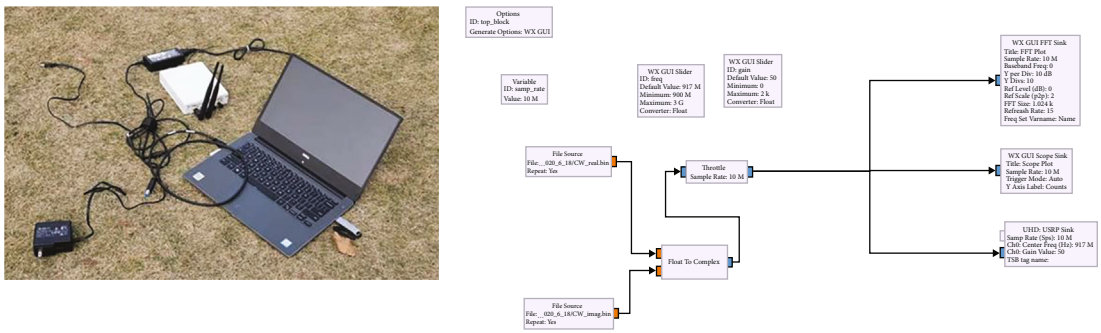
5.1.1. Anonymity between Initiators and Participants. To hide the identities of both the initiator and participants, we use the ring signature proposed by Rivest et al. [38]. The ring signature can hide the real signer. And if the asymmetric encryption function is used in the signature, the security of the signature can be enhanced. The ring signature in our system is based on the asymmetric encryption function.

5.1.2. Data Security. All data segments are encrypted with the receiver's public key; without the receiver's private key, the attacker cannot crack the cryptosystem and obtain the encrypted data segments. At the same time, the key exchange and data communication are through the secure channel created by the miner, which enhances the security.

5.2. Performance of Semiphysical Simulation. We first placed the objects randomly in an area with the size of 30 m × 40 m, generated the coordinates of the objects, and selected an object as the source target. After that, we generated a sequence of N coordinates at uniform intervals of 1 m according to the distance to the object. This sequence is the coordinate sequence of the simulated flight path of the UAV. Then, according to the measured channel parameters, the source distance was substituted into the path loss log-

(a) (b)

Figure 7: The proposed UAV system: (a) the assembled UAV; (b) the ground MEC system.



(a) (b)

Figure 8: Experiment system implementation: (a) USRP B210; (b) GNU radio block diagrams.

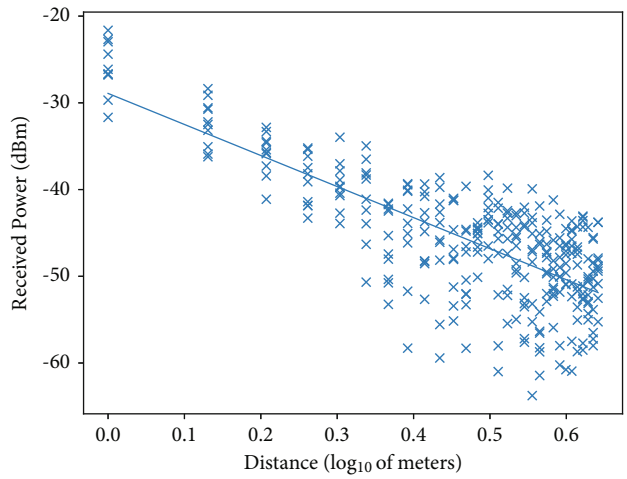


Figure 9: Distance versus power.

normal shadowing model to generate the corresponding measured power value. Next, for all objects, the distance was calculated based on the coordinates of the UAV flight path and the distance of the object coordinates. Noting that the UAV flight passed through N points, we set up an object distance vector with a size of N, as well as a power vector. Figure 10 shows the identification accuracy of the proposed fusion method under different numbers of potential objects. When the number of objects is 2, the accuracy decreases slightly with the increase in experimental rounds and finally reaches a stable value about 70%. When the number of objects is 3, the accuracy also decreases slightly with the increase in experimental rounds and then increases gradually until it is stable at about 53%. We attribute this small fluctuation to the influence of random factors. In the above, we mentioned that objects were randomly placed in the area, and the signal source was randomly selected; a Gaussian noise would also be added when using the log-normal shadowing model to generate the received power at different distances, which would affect the discrimination of objects and then the accuracy. The reason for the decrease in accuracy against the number of objects is that, within the limited

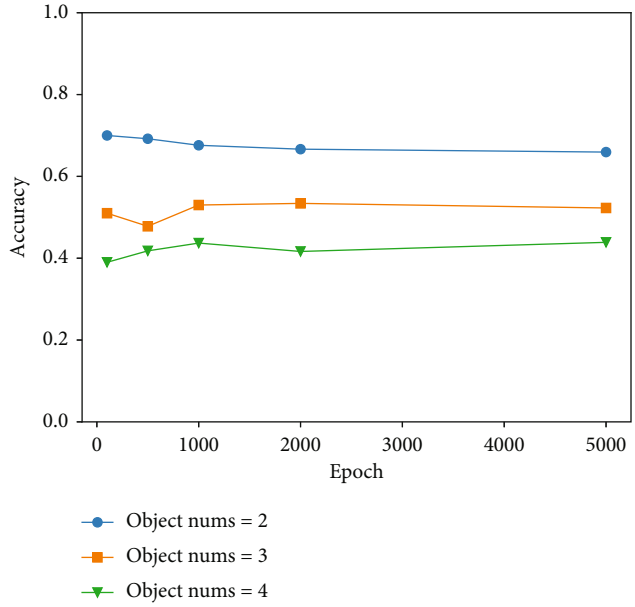


Figure 10: Accuracy under different numbers of objects.

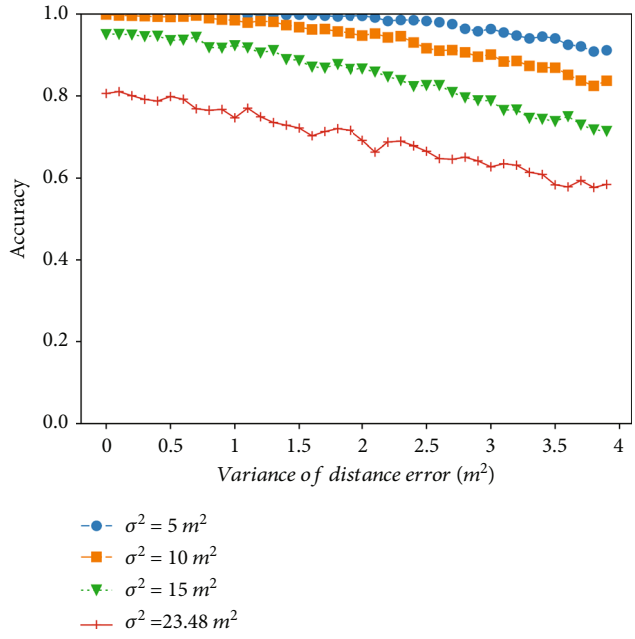
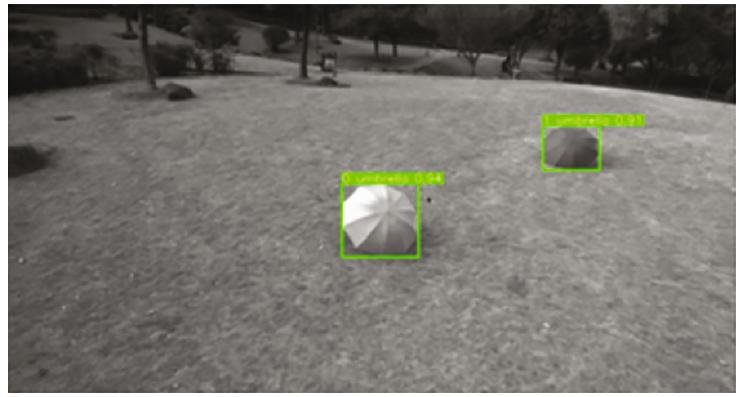


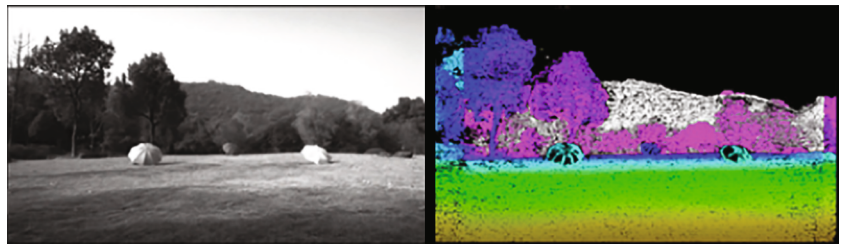
Figure 11: Sensitivity under different distance errors.

area, the increase in the number of objects will also increase the possibility that they have similar distance to the UAV, and thus, the trend of distance versus power change between them tends to be similar.

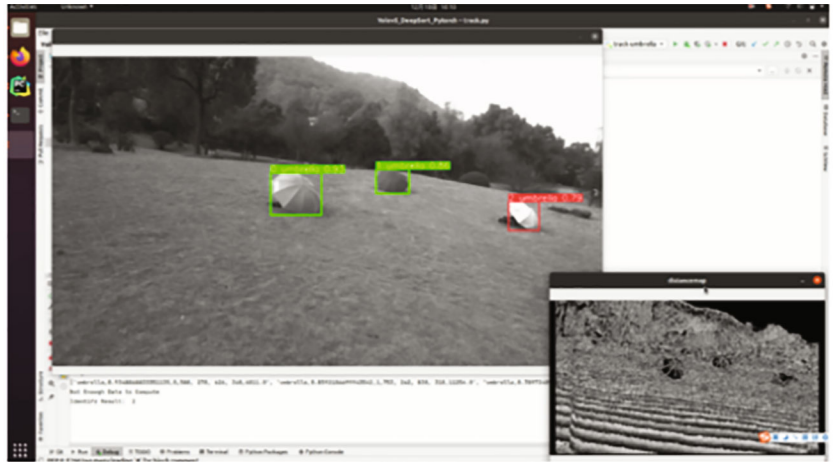
5.3. System Sensitivity Analysis. Figure 11 shows the sensitivity of the fusion method regarding the change of parameters and the depth distance error obtained by the binocular vision in the log-normal model. σ^2 is the variance of the noise mentioned above. The variance indicates the stability of the channel or the received signal-to-noise ratio to a certain extent. To explore the influence of variance on the accuracy of the method, we selected three synthetic and one measured values for experimental simulation. Here, the number of objects was set to 3, the distance between objects was 5 m, the number of experiments was 2000, and the number of distance-received power pairs collected for each experiment was 30. The horizontal coordinate was the variance of the distance error for each measurement in square meter. The four curves indicate the detection accuracy of the fusion



(a)



(b)



(c)

Figure 12: Experiment results: (a) two umbrellas detected; (b) the depth graph; (c) successfully distinguishing the signal source (see the red anchor box).

algorithm when the variance of the shadow fading is set to 5, 10, 15, and 23.48, respectively, while the distance error increases from 0 to 4.0 meters.

When the variance is small, in another word, when the signal-to-noise ratio is high, the discrimination accuracy of the method is high. When σ^2 is equal to 5, the channel is ideal and the accuracy is close to 1.0. With the gradual increase in variance, the accuracy also begins to decline. When σ^2 exceeds 15, the accuracy begins to decline significantly. It is preliminarily inferred that the value of 15 should be a critical point of the method. Also, when the distance error increases, the accuracy decreases gradually. The calculation shows that when the distance error is 4, it will be 2 m different from the real distance in extreme cases. When the object spacing is 5 m, the error will greatly reduce the accuracy. In practice, the distance error estimated by binocular vision is around 1 m, so the accuracy of this method is high enough in real-world applications.

5.4. Physical Experiment. We also launched the UAV to the air to verify our system. The experiment was held on a lawn, and the objects were simply 3 umbrellas, each was placed 3 m away from the other. The whole system worked well, when the 3 umbrellas were in the sight of the camera. By controlling the moving path of the UAV, we got some data. Despite those limitations, the accuracy was very close to the simulation result. Figure 12 shows the setup of the experiment and the real-time output of the system.

6. Conclusion

In this paper, we proposed a signal source identification algorithm to distinguish the target by a blockchain and UAV-enabled system. We used binocular cameras to detect candidate objects and corresponding distances. With the measuring of the received signal strength and mobile edge computing, finally, we used fused object identification techniques to determine the real target object. Blockchain and smart contract technologies were adopted to organize UAVs and users, aimed at providing reliability and security. The accuracy of discrimination could reach 70% for 2 objects and 53% for 3 objects in a semiphysical simulation. The real-world experiment showed the feasibility of the system, and the performance was close to that in the semiphysical simulation.

Regarding future directions for this work, first, satellite positioning (such as GPS and BeiDou) can be utilized to give precise location information of the signal source after identification. Second, only one signal source is considered in this paper; the method will be extended to support multiple signal sources in the future work.

Data Availability

Data is available on request; please contact the corresponding author Huaming Lin (lhm18@live.cn).

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Authors' Contributions

Jian Xiao, Peng Liu, and Huijuan Lu contribute to implementation of wireless signal-aerial image fusion. Huaming Lin and Yi Huang contribute to system architecture design. Hangxiang Fang, Hangguan Shan, and Haoji Hu contribute to object detection and object tracking. Jiayi Xu contributes to binocular depth estimate.

Acknowledgments

This work was supported in part by the Open Project Funding of the Key Laboratory of Electromagnetic Wave Information Technology and Metrology of Zhejiang Province (No. 2020KF0001) and the Natural Science Foundation of China (Grant Nos. 62172134 and 62102125).

References

[1] K. M. Almgren, A. Olofsson, and S. Magnusson, *Radio Signal Source Identification System*, EP, 2001.

[2] Z. Li, A. Giorgetti, and S. Kandeepan, "Multiple radio transmitter localization via UAV-based mapping," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 9, pp. 8811–8822, 2021.

[3] Y. Li, F. Shu, B. Shi, X. Cheng, Y. Song, and J. Wang, "Enhanced RSS-based UAV localization via trajectory and multi-base stations," *IEEE Communications Letters*, vol. 25, no. 6, pp. 1881–1885, 2021.

[4] M. Calì and R. Ambu, "Advanced 3D photogrammetric surface reconstruction of extensive objects by UAV camera image acquisition," *Sensors*, vol. 18, no. 9, p. 2815, 2018.

[5] D. Wierzbicki, "Multi-camera imaging system for UAV photogrammetry," *Sensors*, vol. 18, no. 8, p. 2433, 2018.

[6] J. Mäkynen, C. Holmlund, H. Saari, K. Ojala, and T. Antila, "Unmanned aerial vehicle (UAV) operated megapixel spectral camera," in *Electro-Optical Remote Sensing, Photonic Technologies, and Applications*, vol. 8186, International Society for Optics and Photonics, 2011.

[7] R. J. Moore, S. Thurrowgood, D. Bland, D. Soccol, and M. V. Srinivasan, "A stereo vision system for UAV guidance," in *in 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3386–3391, St. Louis, MO, USA, 2009.

[8] Y. Yu, T. Wang, C. Long, and W. Zhang, "Stereovision based obstacle avoidance strategy for quadcopter UAV," in *in 2018 Chinese Control And Decision Conference (CCDC)*, Shenyang, China, 2018.

[9] Y. M. Mustafah, A. W. Azman, and F. Akbar, "Indoor UAV positioning using stereo vision sensor," *Procedia Engineering*, vol. 41, pp. 575–579, 2012.

[10] R. Fan, J. Jiao, J. Pan, H. Huang, and M. Liu, "Real-time dense stereo embedded in a UAV for road inspection," in *in 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Long Beach, CA, USA, 2019.

[11] K. N. Tahar, A. Ahmad, and W. A. A. W. M. Akib, "UAV-based stereo vision for photogrammetric survey in aerial terrain mapping," in *in 2011 IEEE International Conference on Computer Applications and Industrial Electronics (ICCAIE)*, pp. 443–447, Penang, Malaysia, 2011.

[12] K. Vonehr, S. Hilaski, B. E. Dunne, and J. Ward, "Software defined radio for direction-finding in UAV wildlife tracking," in *in IEEE International Conference on Electro Information Technology*, Grand Forks, ND, USA, 2016.

[13] A. Torabi, M. W. Shafer, G. S. Vega, and K. M. Rothfus, "UAV-RT: an SDR based aerial platform for wildlife tracking," in *in2018 IEEE 88th Vehicular Technology Conference (VTC-Fall)*, pp. 1–6, Chicago, IL, USA, 2018.

[14] S. O. Murphy, C. Sreenan, and K. N. Brown, "Autonomous unmanned aerial vehicle for search and rescue using software defined radio," in *in Vehicular Technology Conference*, Kuala Lumpur, Malaysia, 2019.

[15] Z. Du, C. Wu, T. Yoshinaga et al., "A routing protocol for UAV-assisted vehicular delay tolerant networks," *IEEE Open Journal of the Computer Society*, vol. 2, pp. 85–98, 2021.

[16] D. Ebrahimi, S. Sharafeddine, P. H. Ho, and C. Assi, "Autonomous UAV trajectory for localizing ground objects: a reinforcement learning approach," *IEEE Transactions on Mobile Computing*, vol. 20, no. 4, pp. 1312–1324, 2021.

[17] S. Wu, "Illegal radio station localization with UAV-based Q-learning," *China Communications*, vol. 15, no. 12, pp. 122–131, 2018.

[18] X. Chen, C. Wu, T. Chen et al., "Information freshness-aware task offloading in air-ground integrated edge computing systems," *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 1, pp. 243–258, 2022.

[19] L. Qiao, S. Dang, B. Shihada, M. S. Alouini, and Z. Lv, "Can blockchain link the future?," *Digital Communications and Networks*, 2021.

[20] P. J. Taylor, T. Dargahi, A. Dehghantanha, R. M. Parizi, and K. K. R. Choo, "A systematic literature review of blockchain cyber security," *Digital Communications and Networks*, vol. 6, no. 2, pp. 147–156, 2020.

[21] A. Kapitonov, S. Lonshakov, A. Krupenkin, and I. Berman, "Blockchain-based protocol of autonomous business activity for multi-agent systems consisting of UAVs," in *in 2017 Workshop on research, education and development of unmanned aerial systems (RED-UAS)*, pp. 84–89, Linköping, Sweden, 2017.

[22] A. I. Abhi and S. Y. Shin, "BUS: a blockchain-enabled data acquisition scheme with the assistance of UAV swarm in Internet of things," *IEEE Access*, vol. 7, pp. 103231–103249, 2019.

[23] A. S. Khan, G. Chen, Y. R. Bschons, G. Zheng, and S. Lambotharan, "Trusted UAV network coverage using blockchain, machine learning, and auction mechanisms," *IEEE Access*, vol. 8, pp. 118219–118234, 2020.

[24] S. Rakesh, B. Rojeena, and P. S. Anish, "A new type of blockchain for secure message exchange in VANET," *Digital Communications and Networks*, vol. 6, no. 2, pp. 177–186, 2020.

[25] Z. Liu, C. Zhan, Y. Cui, C. Wu, and H. Hu, "Robust edge computing in UAV systems via scalable computing and cooperative computing," *IEEE Wireless Communications*, vol. 28, no. 5, pp. 36–42, 2021.

[26] K. Christidis and M. Devetsikiotis, "Blockchains and smart contracts for the Internet of things," *IEEE Access*, vol. 4, pp. 2292–2303, 2016.

[27] M. H. Mughal, M. J. Khokhar, and M. Shahzad, "Assisting UAV localization via deep contextual image matching," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 4, pp. 2445–2457, 2021.

[28] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: optimal speed and accuracy of object detection," 2020, arXiv preprint arXiv:2004.10934.

[29] P. Zhu, Y. Sun, L. Wen, Y. Feng, and Q. Hu, "Drone based RGBT vehicle detection and counting: a challenge," 2020, arXiv preprint arXiv:2003.02437.

[30] G. S. Xia, X. Bai, J. Ding et al., "DOTA: a large-scale dataset for object detection in aerial images," in *in 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, 2018.

[31] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," in *in 2017 IEEE International Conference on Image Processing (ICIP)*, Beijing, China, 2017.

[32] H. Hirschmuller, "Stereo processing by semiglobal matching and mutual information," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 2, pp. 328–341, 2008.

[33] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 11, pp. 1222–1239, 2001.

[34] P. Viola and W. M. Wells, "Alignment by maximization of mutual information," *International Journal of Computer Vision*, vol. 24, no. 2, pp. 137–154, 1997.

[35] Y. Zhao, Z. Li, N. Cheng, B. Hao, and X. Shen, "Joint UAV position and power optimization for accurate regional localization in space-air integrated localization network," *IEEE Internet of Things Journal*, vol. 8, no. 6, pp. 4841–4854, 2021.

[36] M. Atif, R. Ahmad, W. Ahmad, L. Zhao, and J. J. P. C. Rodrigues, "UAV-assisted wireless localization for search and rescue," *IEEE Systems Journal*, vol. 15, no. 3, pp. 3261–3272, 2021.

[37] S. Rouhollah and Z. Hadi, "RSS localization using unknown statistical path loss exponent model," *IEEE Communications Letters*, vol. 22, 2018.

[38] R. L. Rivest, A. Shamir, and Y. Tauman, "How to leak a secret," in *in Proc. International Conference on the Theory and Application of Cryptology and Information Security*, pp. 552–565, 2001.

Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 7787866, 12 pages
https://doi.org/10.1155/2022/7787866

Research Article

Privacy Protection of Task in Crowdsourcing: Policy-Hiding and Attribute Updating Attribute-Based Access Control Based on Blockchain

Kunwei Yang,[1] Bo Yang,[1] Yanwei Zhou,[1] Tao Wang,[1] and Linming Gong[2]

[1]*School of Computer Science, Shaanxi Normal University, Xi'an 710119, China*
[2]*School of Computer Science, Xi'an Polytechnic University, Xi'an 710048, China*

Correspondence should be addressed to Bo Yang; byang@snnu.edu.cn

Received 14 December 2021; Accepted 23 February 2022; Published 24 March 2022

Academic Editor: Qingqi Pei

Crowdsourcing is a new way to solve complex problems by using human intelligence. However, the tasks and user information privacy concerned in crowdsourcing have not been adequately addressed. It is necessary to design a privacy protection mechanism for tasks that need to be restricted to specific user groups. Ciphertext-policy attribute-based encryption (CP-ABE) is an efficient and feasible cryptographic tool, particularly for crowdsourcing systems. The encryptor can choose the access policy independently, which limits the scope of decryption users. At present, most CP-ABE schemes adopt a centralized management platform, which poses problems such as high trust-building costs, DDoS attacks, and single point of failure. In this paper, we propose a new access control scheme based on CP-ABE and blockchain, which has the properties of policy hiding and attribute updating. To protect the privacy of worker's attributes, we adopt a test algorithm based on a fully homomorphic cryptosystem to confidentially judge whether the worker's attribute lists match the hidden attributes policy in ciphertext or not before the decryption. Experiment results and comprehensive comparisons show that our mechanism is more flexible, private, and scalable than existing schemes.

1. Introduction

With the rapid development of science and technology, a new generation of information technology represented by 5G, Internet of Things, edge intelligence, and blockchain has emerged, which has realized a comprehensive link between people, machines, and things, and built new infrastructure, application mode, industrial ecology, and service system [1–7]. In the application of new technologies, people are most concerned about information security and privacy protection. Researchers have conducted extensive and in-depth research in many fields to promote the healthy development of related application technologies [8–13].

In recent years, crowdsourcing as a distributed problem-solving model has been widely concerned by researchers. As in Figure 1, the model consists of three parties: requesters, workers, and a crowdsourcing platform. Requester submits tasks through the crowdsourcing platform. Workers get tasks from the platform and get corresponding rewards after completing tasks. At present, many crowdsourcing applications are widely used, such as UBER and Waze [14]. These applications have been integrated into many aspects of daily life.

Due to the diversity of perceptual tasks, it is necessary to select appropriate policies for the privacy protection of specific perceptual tasks. However, the traditional crowdsourcing systems based on a centralized management platform have the following weaknesses. First, it is vulnerable to DDoS attacks, remote hijackings, and so on. Second, there is a potential danger of a single point of failure. Third, users' sensitive information and task solutions are exposed to the risks of leakage. Fourth, from the perspective of privacy

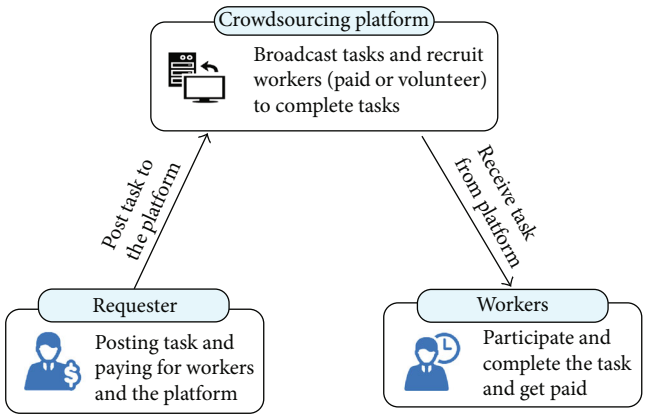


Figure 1: The model of crowdsourcing.

protection, it is impossible to implement the task hiding publishing mechanism within a limited receiver scope. The above problems seriously restrict the development of crowdsourcing system applications.

To solve these problems, we use blockchain [15] and CP-ABE [16] technologies to construct a credible task hiding access control mechanism in crowdsourcing. By using cryptography, timestamps, consensus mechanism, and incentives mechanism, blockchain enables point-to-point transaction and collaboration. Applying blockchain technology to a crowdsourcing system solves problems such as trust, incentive, and decentralization. CP-ABE is a branch of ABE mechanism [17], in which the access control policy is determined by an encryptor. The user's decryption key is associated with its own attribute set, which can be decrypted correctly only when the user attributes meet the policy requirements. Due to this special property, CP-ABE has been widely used in various scenarios.

In this paper, we integrate blockchain with CP-ABE cryptosystems to implement an access control mechanism for task privacy publishing. The main innovations of this paper are summarized as follows.

(i) We propose a new access control mechanism based on blockchain under the crowdsensing system, which has decentralized, fine-grained, flexible, and security features. It can be applied in the task distribution phase that ensuring crowdsourcing tasks can only be accessed by users who meet the requirements of the access policy. The system establishes a secure way to exercise access control between requesters and workers, which is more suitable for complex crowdsensing systems than other schemes

(ii) We propose a new CP-ABE scheme with policy hiding property. In the encryption phase, the access policy is not displayed in the ciphertext tuple, which ensures the privacy of the access policy. Different from the present CP-ABE schemes, in which decryptors need to do excessive calculations to determine whether their attributes meet the access policy in the decryption phase, the new scheme adds a test phase before the decryption. With much less computation than decryption itself, the test uses a fully homomorphic cryptosystem to ensure the privacy of policy and attributes

(iii) Our CP-ABE scheme features attribute updating. It is very flexible to add new attributes in the access policy because we can only generate the public key parameters for the new attributes and the existing public key can remain unchanged. To decrypt the ciphertext with newly added attributes in the access policy, the decryptors must obtain a new secret key including the newly added attributes again. The feature of attribute updating can greatly improve the flexibility and scalability of our scheme

The remainder of the paper is organized as follows. In Section 2, we present the related work. The preliminaries are given in Section 3, an overview of the new access control system in Section 4. Section 5 describes the proposed system, including an access control mechanism and a CP-ABE scheme. We analyze the security of our access control system in Section 6 and make comparisons and performance evaluation of our scheme in Section 7. The last section is the conclusion.

2. Related Work

In a heterogeneous and complex network environment, research on access control technology is heading towards the direction of fine granularity, which takes into consideration users, resources, operation, environment, and other factors. In this section, we would introduce some related work in the field of blockchain-based access control and CP-ABE schemes.

2.1. Blockchain-Based Access Control. Because of the low management efficiency, lack of flexibility, poor scalability, and other problems existing in the current centralized access control mechanism, researchers began to pay attention to the blockchain-based access control mechanism. [18] proposed a privacy-preserving authorization management framework for IoT by using blockchain that enables users to own and control their data. The data access control is implemented through a series of transactions that are used to grant, get, delegate, and revoke access. [19] realized the transfer of user rights through blockchain transactions and stores the transaction results on the blockchain to ensure that the executed access authorization operation cannot be denied. In view of the data privacy problem of cloud, [20] proposed a fine-grained access control scheme based on the blockchain model and attribute-based cryptosystem, which has the nature of privacy-preserving and user-controlled. In the scheme, a smart contract is used to ensure the scalability of the access control. [21] proposed an access management system for cloud federations. It allows federated organizations to enforce attribute-based access control policies on their data in a privacy-preserving fashion. By using blockchain and Intel SGX trusted hardware, the integrity of the policy evaluation process in the scheme is

ensured. To address the problem that roles are used across organizations in the network, [22] used smart contract as the trusted basis in access control and employed the challenge-response mechanism to realize the verification of user roles. [23] proposed a blockchain-based big data access control mechanism based on the attribute-based access control (ABAC) model. To ensure the tamper-resistant, auditability, and verifiability of access control information, the scheme described a transaction-based access control policy and entity attribute information management method. [24] proposed an access control model called timely CP-ABE, where the user legitimacy is verified by blockchain nodes and file sharing is based on a CP-ABE scheme that adds temporal dimension. [25] proposed an access control mechanism in a smart grid scenario based on an identity-based combined encryption, signature, and signcryption scheme. A new consensus algorithm in the power system is designed to solve the key escrow problem. [26] proposed to define an access control system that guarantee the auditability of access control policy evaluation. The core idea of the scheme is to codify access control policies as smart contracts and deploy them on a blockchain. In this way, the decision process of the policy can be executed automatically. By using CP-ABE, [27] proposed a ciphertext policy and attribute hiding access control scheme based on blockchain. To ensure the attribute privacy during authorization validation, they use the ElGamal cryptosystem [28] to secretly match user attributes and access policies. However, we point out that the ElGamal cryptosystem does not have additive homomorphism; so, there are loopholes in the access policy match phase. In addition, the scheme adopts a secure channel to transmit secret key, which increases the communication cost.

2.2. CP-ABE Schemes. The notion of ABE was first introduced by Sahai and Waters in [17], where both ciphertexts and secret keys are associated with sets of attributes. There are two variants of ABE: key-policy ABE (KP-ABE) [29] and ciphertext policy ABE (CP-ABE) [16]. In a KP-ABE scheme, the decryption key is associated with an access policy, and the ciphertext is associated with a set of attributes. In the decryption phase, the ciphertext can be decrypted if and only if the attribute set of the ciphertext satisfies the access policy of the decryption key. On the contrary, in a CP-ABE scheme, the ciphertext is associated with an access policy, and the decryption key is generated over a set of attributes. CP-ABE is perfectly suitable for fine-grained access control environments because it enables data owners to formulate and enforce access policies themselves.

Since [17] proposed the first CP-ABE scheme, researchers have proposed many specific CP-ABE schemes. In the CP-ABE mechanism, the more complex the policy, the more complicated the system is designed, and the more difficult to obtain the security proof of the mechanism. Researches on CP-ABE focus mainly on the design of access policy, which is generally classified into AND gates, access tree, and LSSS matrix. Cheung and Newport [30] presented a CP-ABE-based AND gates that is proven to be secure under the standard model. Subsequently, Nishide et al. [31] and Emura et al. [32] realized policy hiding and efficiency improvement, respectively, on the basis of the scheme [30]. Lai et al. [33] proposed a ciphertext policy hiding CP-ABE scheme, which can be expressed as AND gates on multivalued attributes with wildcards, and proved that it is fully secure. Different from the above types, there are many CP-ABE schemes [16, 34–36] with tree, linear secret sharing scheme (LSSS), and 0-1 coding as access structures, which have strong policy expression ability. For application scenarios that access policies need to be hidden and updated, many anonymous ABE schemes [27, 37–39] have been proposed. In an anonymous ABE, the access policy is hidden so that the user has no idea about the attribute content in the policy. However, most previous schemes cannot securely add new attributes in the access policy because these schemes have a common flaw that a decryptor can combine all old secret key components to reconstruct the exponent for decryption.

3. Preliminaries

In this section, we showcase the associated basic knowledge. See Table 1 for the notations used herein.

3.1. Blockchain. Blockchain was originally known as the underlying technology of Bitcoin. It was not until 2015 that blockchain became a prominent concept by researchers. Consisting of peer-to-peer (P2P) network, consensus protocol, transaction, smart contract, and a series of other technologies, blockchain can provide a trusted and distributed network environment. This new technology has solved the security risks brought by the centralization model. Applications based on blockchain technology can provide a new direction to reduce the middleman role.

3.1.1. P2P Network. Unlike the traditional client/server mode, the P2P network is a net system in which information is exchanged entirely by nodes without a central server. In a P2P network, blockchain nodes can join and exit freely, and the network system can expand and shrink freely.

3.1.2. Consensus Protocol. The consensus mechanism is the cornerstone of blockchain and an important guarantee for the security of the blockchain system. It can be used to solve the consistency problem caused by block distributed storage. The consensus mechanism is the foundation for building trust in the blockchain and contains an incentive mechanism for the effective operation of the blockchain system. The common consensus protocol includes POW [15], POS [40], BFT [41], and mixture of various mechanisms.

3.1.3. Transaction. Blockchain adopts a transaction data model composed of input, output, and digital signature to ensure that every transaction can be tracked. Merkel hash tree is used to package and aggregate all transaction information in a period of time to ensure that the transaction data will not be tampered.

3.1.4. Smart Contract. The concept of smart contract was put forward by Nick Szabo [42] in 1994. It is a computer program that executes and verifies the contract in an

Table 1: Notation description.

Notations	Descriptions
G, G_T	Two cyclic multiplicative groups
$\mathbb{U} = \{A_1, A_2, \cdots, A_n\}$	The attribute universe
$\mathbb{A}$	An attribute set
$\mathbb{P}$	An access policy
A_i	An attribute
k	A security parameter
B.PK	A public key registered in blockchain
B.SK	A private key registered in blockchain
PK	A public key in CP-ABE
SK	A secret key in CP-ABE
MSK	A master secret key in CP-ABE
M	Task information
$E(\cdot)$	A fully homomorphic cryptosystem
H	A collision-free hash function
CT	A CP-ABE ciphertext
Tx	A transaction on the blockchain

information way. The birth of blockchain provides a credible execution environment for smart contracts and accelerates the development of smart contracts. At the same time, the application of smart contracts expands blockchain technology from the earliest monetary system to a wider range of practical application scenarios.

3.2. Attribute and Access Structure. We define the attribute and access structure as provided in [43]. Let $\mathbb{U} = \{A_1, A_2, \cdots, A_n\}$ represent the attribute universe and $\mathbb{A}$ represent the user's attribute set, where $\mathbb{A} \subset \mathbb{U}$. In this paper, we use n-bit string $a_1 a_2 \cdots a_n$ to represent the user's attribute information.

$$\begin{cases} a_i = 1, A_i \in \mathbb{A}, \\ a_i = 0, A_i \notin \mathbb{A}. \end{cases} \tag{1}$$

For instance, let $n = 5$. $\mathbb{A} = 10011$ means that the attribute set consists of the attributes $\{A_1, A_4, A_5\}$.

We also represent the access policy $\mathbb{P}$ with an n-bit string $b_1 b_2 \cdots b_n$ defined as follows.

$$\begin{cases} b_i = 1, A_i \in \mathbb{P}, \\ b_i = 0, A_i \notin \mathbb{P}. \end{cases} \tag{2}$$

For instance, let $n = 5$. The $\mathbb{P} = 10111$ means the access policy $\mathbb{P}$ requires $\{A_1, A_3, A_4, A_5\}$ attributes.

In the attribute universe, each of A_i can be represented in terms of a group element $g_i \in G$, where G is a cyclic group of order N. There is a $\alpha_i \in Z_N^*$ that satisfies $g_i = g^{\alpha_i}$. Let us assume that the $\mathbb{P}$ has z elements equal to 1. We compute $\mathbb{A} \cdot \mathbb{P} = a_1 b_1 + a_2 b_2 + \cdots + a_n b_n$. This implies that if an attribute set $\mathbb{A}$ fulfills the access policy $\mathbb{P}$, then $\mathbb{A} \cdot \mathbb{P}$ must be z. On the contrary, if $\mathbb{A}$ does not fulfills $\mathbb{P}$, the result of $\mathbb{A} \cdot \mathbb{P}$ must be less than z.

3.3. Fully Homomorphic Cryptosystem. The concept of fully homomorphic encryption [44] was proposed by Rivest et al. in 1970s, and it has become an important technology to solve the security problem arising in cloud service. The way to construct such schemes has been a hard problem for cryptographers. The first fully homomorphic cryptosystem based on ideal lattice was proposed by [45]. This scheme can perform any computation on the encrypted data without decrypting and does not affect the confidentiality of the data. The fully homomorphic cryptosystem means that it satisfies the properties of both additive and multiplicative homomorphisms simultaneously. It can be expressed by the following formula.

$$f(E(m_1), E(m_2), \cdots, E(m_k)) = E(f(m_1, m_2, \cdots, m_k)). \tag{3}$$

If f is an arbitrary function, it is called fully homomorphic encryption.

3.4. Composite Order Bilinear Groups. The concept of composite order bilinear groups was proposed in [46]. In this paper, we select two prime numbers of equal length as the order of two subgroups of group G.

Let $\mathcal{G}(1^\kappa)$ be a group generation algorithm and κ denote a security parameter. The algorithm outputs public parameters $\mathbb{G} = (g, p, r, G, G_T, e(\cdot, \cdot))$, such that (1) g is a generator of G, (2) G and G_T are two multiplicative cyclic groups of prime order $N = pr$, and (3) $e : G \times G \longrightarrow G_T$ denotes a computable bilinear map that satisfies the following properties:

(i) Bilinearity: $e(u^a, v^b) = e(u, v)^{ab}$, for all $a, b \longleftarrow_R \mathbb{Z}_N^*$ and $u, v \longleftarrow_R G$

(ii) Nondegeneracy: $e(g, g) \neq 1_{G_T}$, where 1_{G_T} is the generator of G_T

(iii) Computability: given the elements $v, u \in G$, $e(u, v)$ can be computed efficiently

Furthermore, let G_p, G_r denote two subgroups of G with order p, r, and $h_p \in G_p$, $h_r \in G_r$. Then, we have $e(h_p, h_r) = 1$.

3.5. Complexity Assumptions. Let the definition of $\mathbb{G} = (g, N, p, r, G, G_T, e(\cdot, \cdot))$ be the same as above, and $\mathcal{S}$ be an adversary whose purpose is to solve DBDH problem.

Definition 1. (DBDH problem). For the exponents a, b, c, z randomly selected from $\mathbb{Z}_N^*$, two tuples $\mathcal{T}_1 = (g, g^a, g^b, g^c, e(g, g)^{abc})$ and $\mathcal{T}_0 = (g, g^a, g^b, g^c, e(g, g)^z)$ are computationally indistinguishable. The advantage of adversary $\mathcal{S}$ in solving DBDH problem is defined as

$$\mathrm{Adv}_{\mathcal{S}}^{\mathrm{DBDH}}(\kappa) = |\Pr[\mathcal{S}(\mathcal{T}_1) = 1] - \Pr[\mathcal{S}(\mathcal{T}_0) = 1]|. \tag{4}$$

We say that the DBDH assumption holds if for all adversaries $\mathcal{S}$, we have

$$\mathrm{Adv}_{\mathcal{S}}^{\mathrm{DBDH}}(\kappa) \leq \mathrm{negl}(\kappa). \tag{5}$$

3.6. Definition of CP-ABE Scheme. A CP-ABE scheme consists of Setup, Encrypt, KeyGen, and Decrypt algorithms. The specific process is as follows.

Setup(1^{κ}): the setup algorithm takes the security parameter κ as input and outputs a public key PK and a master secret key MK

KeyGen(MK, $\mathbb{A}$): given the master secret key MK and an attribute list $\mathbb{A}$, the keyGen algorithm returns a secret key $SK_{\mathbb{A}}$

Encrypt(PK, M, $\mathbb{P}$).: given PK, a message M, and an access policy $\mathbb{P}$, the encrypt algorithm returns a ciphertext CT

Decrypt(CT, $\mathrm{SK}_{\mathbb{A}}$): given a ciphertext CT and a secret key $\mathrm{SK}_{\mathbb{A}}$, the decrypt algorithm returns the message M if $\mathbb{A} \vDash \mathbb{P}$. Otherwise, it returns $\perp$ with overwhelming probability

Setup(1^{κ}): the setup algorithm takes the security parameter κ as input and outputs a public key PK and a master secret key MK

3.7. Security Model for CP-ABE. In this paper, we use the security model proposed by [30]. A CP-ABE scheme is secure against chosen plaintext attacks (CPA) if no probabilistic polynomial time adversary has a nonnegligible advantage in the following game.

(i) Init: the adversary defines the target access policy $\mathbb{P}$ and sends it to the challenger

(ii) Setup: the challenger executes the setup algorithm and sends PK to the adversary

(iii) Phase 1: the adversary makes key generation queries by submitting an attribute list $\mathbb{A}$. The challenger answers with a secret key $\mathrm{SK}_{\mathbb{A}}$ if $\mathbb{A} \nvDash \mathbb{P}$. The process can be repeated adaptively

(iv) Challenge: the adversary selects two equal-length messages M_0 and M_1 and sends them to the challenger. After selecting a random bit $\mu \in \{0, 1\}$, the challenger generates a ciphertext CT by encrypting M_{μ} with $\mathbb{P}$. Then, it sends CT to the adversary

(v) Phase 2: the adversary makes key generation queries, and the challenger answers as Phase 1

(vi) Guess: finally, the adversary outputs a guess $\mu' \in \{0, 1\}$. If $\mu' = \mu$, the challenger outputs 0; otherwise, it outputs 1

4. The Specific Process of Access Control System

4.1. System Architecture. Our system architecture is shown in Figure 2, which involves four entities, namely, requesters, workers, cloud, and blockchain.

Requesters need to set an access policy first and then encrypt the task information using our CP-ABE scheme. Finally, they post the ciphertext of the task to the cloud and send the ciphertext address of the task to the blockchain via a storage transaction. When the task is completed, the requester will reward workers accordingly.

Workers are a group of users with different attributes who compete for the task to get rewards. The attributes of the workers determine whether the workers can get the plaintext of the task.

Cloud has an extremely large storage capacity to offer data storage services. In our system, the cloud is a data storage platform for storing encrypted task information.

Blockchain is a distributed platform to record relevant transaction information, which is open and transparent to requesters and workers.

4.2. Specific Process. Next, we describe the specific process of the system in detail.

Step 1. In the first step, the requester performs the setup algorithm of CP-ABE as follows.

Setup(1^{κ}): our construction takes a security parameter κ as input and runs the group generator to get $(N = p\ r, G, G_T, e)$, where $G = G_p \times G_r$. It picks the generators g_p of G_p. Let the attribute universe $\mathbb{U} = \{A_1, A_2, \cdots, A_n\}$ be the set of n attributes. For each attribute A_i where $1 \leq i \leq n$, choose random values $\{\alpha_i\}_{1 \leq i \leq n}$ from Z_N^*, set $A_i = g_p^{\alpha_i}$. Algorithm selects random elements $\omega, \beta \in Z_N^*$. Let $Y = e(g_p, g_p)^{\omega}$ and $B = g_p^{\beta}$. The public key is $\mathrm{PK} = (g_p, Y, B, \{A_i\}_{1 \leq i \leq n})$. And the master secret key is $\mathrm{MSK} = (\omega, \beta, \{\alpha_i\}_{1 \leq i \leq n})$.

Step 2. The requester sets an access policy $\mathbb{P} = \{b_1, b_2, \cdots, b_n\}$ and encrypts the task by using Encryption algorithm.

Encrypt(PK, M, $\mathbb{P}$): the requester picks up random values $r_i \in Z_N^*$ for $b_i = 1$, set $r = \sum_{b_i=1} \boxtimes r_i$. Then, the requester computes $C_0 = B^r$, $\tilde{C} = \mathrm{MY}^r$, and $\{C_{i,1}, C_{i,2}\}$ as follows:

$$\left\{C_{i,1}, C_{i,2}\right\} = \begin{cases} \left\{g_p^{r_i}, A_i^{r_i}\right\}, & \text{if } b_i = 1, \\ \left\{T_i, T_i'\right\}, & \text{if } b_i = 0, \end{cases} \tag{6}$$

where $M \in G_T$, T_i, $T_i' \in_R G_r$. The ciphertext is $CT = (C_0, \tilde{C}, \{C_{i,1}, C_{i,2}\}_{1 \leq i \leq n})$.

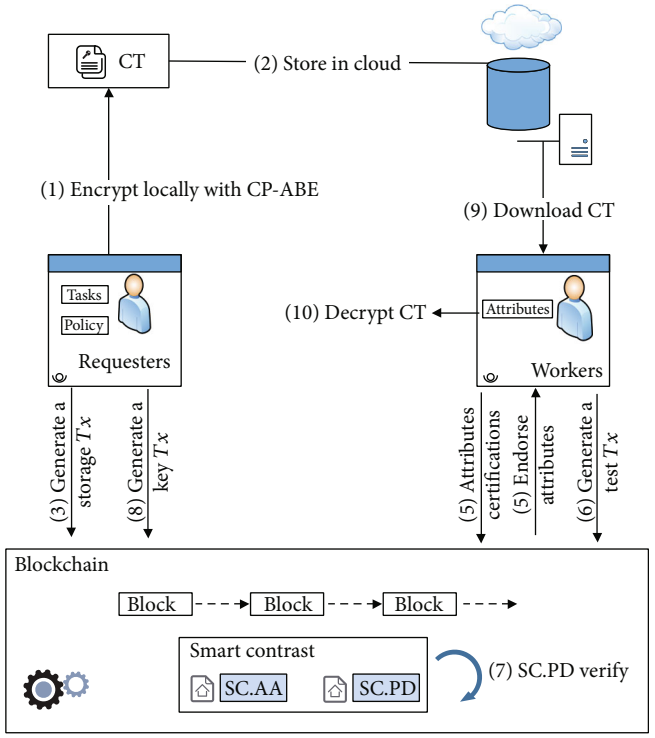


Figure 2: System architecture.

Step 3. The requester uploads the ciphertext of the task to the cloud server.

Step 4. In the storage phase, the requester only needs to publish the ciphertext address of the task to the blockchain through a storage transaction, to reduce the storage cost of the blockchain. The specific process is as follows.

The requester takes $(\mathrm{ID}_s, B_r.\mathrm{SK}, \mathrm{CT}, \mathrm{Address}_{\mathrm{CT}}, \mathbb{P}, z)$ as input, where ID_s is the identifier of the current storage transaction, $\mathrm{Address}_{\mathrm{CT}}$ is the cloud storage address of the ciphertext CT, and z is the number of attributes in the access policy. Then, the requester makes the following calculation.

(1) Calculate the hash value of the CT:

$$h_c = H(\mathrm{CT}). \tag{7}$$

(2) Calculate the ciphertext of access policy vector encrypted by the fully homomorphic cryptosystem

$$E(\mathbb{P}) = (E(b_1), E(b_2), \cdots, E(b_n). \tag{8}$$

(3) Calculate the hash value of the storage transaction:

$$h_s = H(ID_s, \mathrm{Address}_{\mathrm{CT}}, z, E(\mathbb{P}), h_c). \tag{9}$$

(4) Use the private key $B_r.\mathrm{SK}$ to sign the hash value of the storage transaction. It generates a sign (h_s).

Finally, the requester generates a transaction:

$$Tx_{\mathrm{storage}} = (ID_s, \mathrm{Address}_{\mathrm{CT}}, z, E(\mathbb{P}), h_c, \mathrm{sign}\ (h_s)) \tag{10}$$

and publishes it on the blockchain.

Step 5. Workers submit an attribute authentication transaction and get attribute certification by calling smart contract SC. AA that is used to provide attribute management services for workers. Then, workers' attributes are authenticated and endorsed by SC. AA. Meanwhile, the attributes of workers recorded on the blockchain will not change anymore.

Step 6. After getting the authentication of attributes, a worker sends the ciphertext of attributes to the blockchain via an attribute test transaction. It takes $(ID_t, B_w.\mathrm{SK}, \mathbb{A}, ID_s)$ as input and makes the following calculation.

(1) Calculate the ciphertext of the worker's attribute vector encrypted by a fully homomorphic cryptosystem

$$E(\mathbb{A}) = (E(a_1), E(a_2), \cdots, E(a_n)). \tag{11}$$

(2) Calculate the hash value of the test transaction

$$h_t = H(ID_t, E(\mathbb{A}), ID_s). \tag{12}$$

(3) Use the private key B_w.SK to sign the hash value of the test transaction. It generates sign (h_t).

Finally, the worker generates a transaction:

$$Tx_{\text{test}} = (ID_t, E(\mathbb{A}), ID_s, \text{sign}\ (h_t)) \tag{13}$$

and publishes it on the blockchain.

Step 7. It can get $E(\mathbb{A})$, $E(\mathbb{P})$, z through the identifier ID_t and ID_s. The smart contract SC.PD calculates $E(\mathbb{A}) \times E(\mathbb{P})$ by using the above fully homomorphic cryptosystem.

$$\begin{aligned} E(\mathbb{A}) \times E(\mathbb{P}) &= E(a_1) \cdot E(b_1) + E(a_2) \cdot E(b_2) + \cdots + E(a_n) \cdot E(b_n) \\ &= E(a_1 b_1 + E(a_2 b_2) + \cdots + E(a_n b_n) = E\left(z'\right). \end{aligned} \tag{14}$$

Finally, the smart contract SC. PD returns the value$E(z')$.

Step 8. The requester decrypts $E(z')$ to get z'. If $z' = z$, it can determine that the worker's attributes satisfy the CP-ABE policy. Then, the requester will run the KeyGen algorithm to generate the CP-ABE secret key $\text{SK}_{\mathbb{A}}$. Afterwards, the requester encrypts $\text{SK}_{\mathbb{A}}$ with B_w.PK and sends the ciphertext of $SK_{\mathbb{A}}$ to the blockchain.

KeyGen(MSK, $\mathbb{A}$). Firstly, the requester selects a random s from Z_N^* and let $D_0 = g_p^{\omega+s/\beta}$. Then, for every $i \in \{1, \cdots, n\}$, the requester picks up random values $\lambda_i \in Z_N^*$ and computes

$$\left\{D_{i,1}, D_{i,2}\right\} = \left\{g_p^{s+\alpha_i \lambda_i}, g_p^{\lambda_i}\right\}. \tag{15}$$

The secret key $SK_{\mathbb{A}} = (D_0, \{D_{i,1}, D_{i,2}\}_{1 \le i \le n})$.

Step 9. The worker downloads the ciphertext of the task from the cloud via CT address of Tx_{storage}. At the same time, the worker can decrypt and obtain $\text{SK}_{\mathbb{A}}$ by using B_w.SK.

Step 10. By using the private key $\text{SK}_{\mathbb{A}}$, the worker runs the decryption algorithm Decrypt to obtain the task data.

Derypt(CT, $\text{SK}_{\mathbb{A}}$): the worker tries to decrypt the CT $= (C_0, \tilde{C}, \{C_{i,1}, C_{i,2}\}_{1 \le i \le n})$ without knowing $\mathbb{P}$ by using his $\text{SK}_{\mathbb{A}} = (D_0, \{D_{i,1}, D_{i,2}\}_{1 \le i \le n})$ associated with the attribute list $\mathbb{A}$. The decryption process is as follows:

$$\begin{aligned} M &= \frac{\tilde{C} \prod_{i=1}^{n} e(C_{i,1}, D_{i,1})}{e(C_0, D_0) \prod_{i=1}^{n} \boxtimes e(C_{i,2}, D_{i,2})} \\ &= \frac{M \cdot e\left(g_p, g_p\right)^{\omega \cdot r} \prod_{b_i=1} e\left(g_p^{r_i}, g_p^{s+\alpha_i \cdot \lambda_i}\right)}{e\left(g_p^{\beta \cdot r}, g_p^{\omega+s/\beta}\right) \prod_{b_i=1} e\left(g_p^{\alpha_i \cdot r_i}, g_p^{\lambda_i}\right)} \\ &= \frac{M \cdot \prod_{b_i=1} e\left(g_p^{r_i}, g_p^{s}\right)}{e\left(g_p^{r}, g_p^{s}\right)} = M. \end{aligned} \tag{16}$$

4.3. Main Idea. Through such a CP-ABE scheme, we can achieve a policy hiding, updatable, and fine-grained access control scheme in crowdsourcing. For requesters, they only want workers who meet the policy requirements to get the task. For workers, they need to do a policy test confidentially to prove whether their attributes meet the policy requirements. The implementation of policy hiding and attribute updating is as follows.

To achieve the goal of policy hiding, it means that the ciphertext CT does not contain policy $\mathbb{P}$, and the workers can decrypt CT without knowing the access policy $\mathbb{P}$. As mentioned above, we use $n-$bit string $a_1 a_2 \cdots a_n$ and $b_1 b_2 \cdots b_n$ to represent the user's attribute set $\mathbb{A}$ and access policy $\mathbb{P}$, respectively. We chose a composite order group G with order $N = pr$. In the encrypt phase, if $b_i = 1$, let $[C_{i,1}, C_{i,2}]$ be well-formed parameters chosen from G_p. If $b_i = 0$, we set $[C_{i,1}, C_{i,2}]$ as two random elements of G_r. If an attribute A_i is required in a policy, then the worker's attribute set must also have that attribute to be validated by SC.PD. Each worker that satisfies the attribute can obtain the decryption key. Thus, if the set of attributes for the workers meet the policy requirements, it does not need to know the access policy $\mathbb{P}$ to complete the decryption.

The attribute update feature requires that it is easy to update the attribute information in the access policy, even after the setup phase is executed. The reason the previous scheme does not have this feature is that the decryptors can use their old secret key that does not contain new attributes to decrypt the new ciphertext. These schemes have a common flaw that the decryptors may combine all old secret key components to reconstruct the exponent of the secret key for decryption. In our scheme, we embed the composition factor r_i of the random number r in the ciphertext tuple instead of the private key, which forces workers to obtain the private key components corresponding to all attributes specified in the new access policy. If any new attributes were added into the access policy after the workers got their private keys, they cannot decrypt correctly until getting the new secret key components.

5. Security Analysis

Our scheme satisfies several security properties, and the specific analysis is as follows.

(1) Collusion resistance: for an attribute-based encryption scheme, it is very important to prevent collusion attacks between adversaries. In our CP-ABE scheme, to decrypt the ciphertext, adversaries have to get $e(g_p^r, g_p^s)$. When an adversary does not possess an attribute, he needs to conspire with a coconspirator who possesses the attribute. However, in the process of secret key generation, the private key of different adversaries uses different random numbers. By means of collusion, the adversary must get $\prod_{a_i \neq 0} e(g_p^{r_i}, g_p^s) \cdot e(g_p^{r_i}, g_p^{s^*})$ in the decryption phase. However, s is the random number in the adversary's secret key, and s^* is the random number in the coconspirator's secret key; so, they cannot calculate $e(g_p^r, g_p^s)$ together. That is why this scheme has the character of resisting collusion attacks

(2) Task confidentiality: this is the most basic security feature to ensure the security of the task. In our system, the task is encrypted and uploaded to the cloud server platform, which is considered to be curious. In the worst case, the cloud server platform may attempt to restore the task information; however, it either does not have a secret key or the attribute does not satisfy the access policy. Therefore, the scheme can ensure the privacy of the task

(3) Decentration: by using blockchain, we can realize an end-to-end crowdsourcing task management. In this process, requesters and workers can interact directly. It avoids DDoS attacks, single point of failure, and leakage of important data that may be encountered on a centralized management platform

(4) Policy privacy protection: in our scheme, the policy is treated as private data that need to be protected. We propose the CP-ABE with policy-hiding property. In the encryption phase, the generated ciphertext does not contain access policy information, which can protect the privacy of the policy. To authenticate the worker's attributes, a policy test algorithm that uses a fully homomorphic cryptosystem is adopted to estimate whether the attribute lists match the hidden attributes policy in ciphertext or not before the decryption. Such an approach ensures the privacy of the policy

(5) Integrity and traceability: our scheme can ensure the integrity of task data and the traceability of access control information through blockchain. Workers can compare the hash value of the task ciphertext in the cloud server platform with the information stored in the blockchain to determine whether the task has been modified. Meanwhile, all authorization records are stored as immutable access transactions in the blockchain; therefore, no one can deny their behavior

(6) Security analysis of CP-ABE: we now prove that the above CP-ABE scheme is selectively secure under the DBDH assumption

Theorem 1. *Assume that there is a probabilistic polynomial-time adversary $\mathcal{S}$ which can break out our CP-ABE scheme in a chosen plaintext attacks model with nonnegligible advantage $\varepsilon(\kappa)$, then a simulator can be constructed to distinguish the DBDH tuple $(g, g^a, g^b, g^c, e(g,g)^{abc})$ from the random tuple $(g, g^a, g^b, g^c, e(g,g)^z)$ with nonnegligible advantage $1/2\varepsilon(\kappa)$.*

Proof. We first let the Sim set the security parameter κ and run the group generator to get the public parameters $(N = pr, g_p, G, G_T, e)$, where $G = G_p \times G_r$, $g_p \in G_p$. The DBDH challenger gives a DBDH tuple $(g_p, g_p^a, g_p^b, g_p^c, Z) \in G^4 \times G_T$ to Sim, where Z is either $e(g_p, g_p)^{abc}$ or $e(g_p, g_p)^z$ with equal probability. The Sim proceeds as follows:

(i) Init: during this phase, Sim receives the challenge access policy $\mathbb{P}$ from $\mathcal{S}$

(ii) Setup: to provide a public key to $\mathcal{S}$, Sim sets Y to be $e(g_p^a, g_p^b) = e(g_p, g_p)^{ab}$. This implies $\omega = ab$. Let the attribute universe be $\mathbb{U} = \{A_1, A_2, \cdots, A_n\}$, Sim chooses random $\alpha_1, \cdots, \alpha_n, \beta \in Z_N^*$, sets $\{A_i = g_p^{\alpha_i}\}_{1 \le i \le n}$ and $B = g_p^{\beta}$. Then, Sim publishes PK as in the real scheme

(iii) Phase 1: $\mathcal{S}$ submits a set $\mathbb{A} = \{a_1, a_2, \cdots, a_n\}$, provided $\mathbb{A} \vDash \mathbb{P}$, and it means that there is at least one $k \in \{1, \cdots, n\}$ that satisfies $b_k = 1$, but $a_k = 0$. Sim answers with a secret key $SK_{\mathbb{A}}$ for $\mathbb{A}$ as follows:

Sim picks up random values $s' \in Z_N^*$. For every $i \in \{1, \cdots, n\}$, Sim chooses random $\lambda_i' \in Z_N^*$. It sets $s = \beta s' - \omega$, and the D_0 component of the secret key can be computed as $D_0 = g_p^{\omega + s/\beta} = g_p^{s'}$. Sim computes the components $\{D_{i,1}, D_{i,2}\} = \{g_p^{\beta s' + \alpha_i \lambda_i'}, g_p^{\lambda_i' + (\omega/\alpha_i)}\}$. When $\lambda_i = \lambda_i' + (\omega/\alpha_i)$, the components $\{D_{i,1}, D_{i,2}\}$ are as follows:

$$D_{i,1} = g_p^{\beta s' + \alpha_i \lambda_i'} = g_p^{\beta s' + \alpha_i \left(\lambda_i - \frac{\omega}{\alpha_i}\right)} = g_p^{s + \alpha_i \lambda_i},$$
$$D_{i,2} = g_p^{\lambda_i' + \frac{\omega}{\alpha_i}} = g_p^{\lambda_i}. \tag{17}$$

The secret key $SK_{\mathbb{A}} = (D_0, \{D_{i,1}, D_{i,2}\}_{1 \le i \le n})$.

(iv) Challenge. $\mathcal{S}$ submits two challenge messages M_0 and M_1 of equal length. Sim chooses $\mu \in \{0, 1\}$ at random and encrypts M_μ based on $\mathbb{P}$. Then, sets $C_0 = (g^c)^\beta$, $\tilde{C} = M_\mu Z$. For the policy $\mathbb{P}$, Sim continues chooses random values $r_i \in Z_N^*$ for $b_i = 1$, set $r = \sum_{b_i = 1} r_i$. Obviously, Sim can compute the ciphertext components $\{C_{i,1}, C_{i,2}\}$ easily

(v) Phase 2: same as Phase 1

(vi) Guess: $\mathcal{S}$ produces a guess μ' of μ. If $\mu' = \mu$, Sim

outputs 1 and otherwise outputs 0. If $Z = e(g_p, g_p)^{abc}$, then CT is a valid ciphertext, in which case the advantage of $\mathcal{S}$ is ε. Hence,

$$\left|\Pr\left[\mu' = \mu \mid Z = e\left(g_p, g_p\right)^{abc}\right] - \frac{1}{2}\right| = \varepsilon(k). \quad (18)$$

If $Z = e(g_p, g_p)^z$, then $\tilde{C}$ is completely random from the view of $\mathcal{S}$. Therefore,

$$\left|\Pr\left[\mu' = \mu \mid Z = e\left(g_p, g_p\right)^{z}\right] - \frac{1}{2}\right| = \text{negl}(k), \quad (19)$$

where $\text{negl}(k)$ is negligible. Hence,

$$\begin{aligned}
&\Pr\left[\mu' = \mu\right] = \Pr\left[Z = e\left(g_p, g_p\right)^{abc}\right] \\
&\Pr\left[\mu' = \mu \mid Z = e\left(g_p, g_p\right)^{abc}\right] + \Pr\left[Z = e\left(g_p, g_p\right)^{z}\right] \\
&\Pr\left[\mu' = \mu \mid Z = e\left(g_p, g_p\right)^{z}\right] = \frac{1}{2}\left(\frac{1}{2} \pm \varepsilon(k)\right) \\
&+ \frac{1}{2}\left(\frac{1}{2} \pm \text{negl}(k)\right) = \frac{1}{2} \pm \frac{1}{2}\varepsilon(k) \pm \frac{1}{2}\text{negl}(k), \text{Adv}_{\mathcal{S}}^{\text{CPA}}(k) \\
&= \left|\Pr\left[\mu' = \mu\right] - \frac{1}{2}\right| = \frac{1}{2}\left|\varepsilon(k) \pm negl(k)\right| \approx \frac{1}{2}\varepsilon(k). \quad (20)
\end{aligned}$$

From the above analysis, we can see that the Sim's advantage in the DBDH game is $1/2\varepsilon(k)$. □

6. Comparisons and Efficiency Evaluation

In this section, we first make comprehensive comparisons of our scheme with related work in terms of security, efficiency, and performance features. Then, we implement our CP-ABE scheme to analyze the efficiency of the algorithm.

6.1. Comparisons. We make a horizontal comparison with the relevant blockchain-based access control schemes in terms of important features, including policy privacy, fine granularity, attribute update, policy test, decentration, and framework. As seen in Table 2, these schemes do not consider the privacy of the policy except [27]. There is a potential problem that the attacker may infer the scope of the user group through the policy information. Centralized entities are introduced in schemes [22, 25], which results in some privacy and security concerns. Meanwhile, these three schemes can only achieve coarse-grained access control. [19, 23] support dynamic updating of policy attributes, which makes the schemes extensible. Only the scheme [27] is capable of supporting the policy test to judge whether the attribute lists match the hidden attributes policy in ciphertext or not before the decryption. However, there is an obvious mistake in the paper that the ElGamal cryptosystem does not have additive homomorphism; so, there are loopholes in the access policy match phase. Therefore, none of these blockchain-based access control schemes can support policy hiding and testing. Our scheme adopts a policy-hiding CP-ABE scheme and a fully homomorphic cryptosystem to realize policy hiding and testing. At the same time, our scheme supports attribute updating. Furthermore, a secure communication channel is not necessary anymore in our system.

Table 2: Comparisons of blockchain-based access control schemes.

Scheme	Policy privacy	Fine granularity	Attribute update	Policy test	Decentration
[22]	×	×	×	×	×
[23]	×	✓	✓	×	✓
[26]	×	✓	×	×	✓
[25]	×	×	×	×	×
[19]	×	✓	✓	×	✓
[18]	×	✓	×	×	✓
[24]	×	✓	×	×	✓
[27]	✓	✓	×	✓	✓
Our	✓	✓	✓	✓	✓

Aiming at the efficiency problem, we compare our CP-ABE scheme with some related CP-ABE schemes. In Table 3, the symbols PK, MK, SK, and CT represent the public key, the master secret key, the secret key, and the task ciphertext, respectively. We use L_G, L_{G_T}, and L_{Z_p} to denote the number of groups G, the target group G_T, and prime group Z_q, respectively. Let n, l, $|A_{\mathbb{A}}|$, $|A_{\mathbb{P}}|$ denote the number of attributes, the number of elements in an attribute category, the number of elements in the user A's attribute set, and the number of attributes in the policy set. Through the horizontal comparison, we can see that under the premise of obtaining the relevant security features, our scheme does not reduce the efficiency and even outperforms the relevant schemes in terms of ciphertext size. It is better for saving storage space on the cloud.

6.2. Efficiency Evaluation. We implement our CP-ABE scheme based on pairing-based cryptography (PBC) library on a laptop with Windows 10, Intel Core i5-8250U CPU, 2.90 GHz, and 16 GB RAM. The size of public parameters and message size is important indicators to evaluate the calculated performance of a CP-ABE scheme. In this experiment, we use type A1 pairing and let the composite N be the universe size. The composite N in our experiments consists of two prime numbers of 517 bits, which means that $|Z_N| = |G| = |G_T| = 1024$ bits.

Our main concern is how the efficiency of the scheme changes with the increase of the number of attributes. The execution result of the algorithm is shown in Figure 3. For each phase, we run the algorithm 10 times and then adopt the average value. As is illustrated, the two phase algorithm time increases as the number of attributes grows. This is due to the calculation of variables in each algorithm

Table 3: Comparison of size of keys and ciphertext in CP-ABE scheme.

Scheme	PK	MK	SK	CT
[30]	$(3n+1)L_G + L_{G_T}$	$(3n+1)L_{Z_q}$	$(2n+1)L_G$	$(n+1)L_G + L_{G_T}$
[31]	$(nl+1)L_G + L_{G_T}$	$(nl)L_G + 2L_{Z_q}$	$(2n+1)L_G$	$(nl+n+1)L_G + L_{G_T}$
[17]	$3L_G + L_{G_T}$	$L_G + L_{Z_q}$	$(2\|A_{\mathbb{A}}\|+1)L_G$	$(2\,\|\,A_{\mathbb{P}}\|+1)L_G + L_{G_T}$
[32]	$(nl+2)L_G + L_{G_T}$	$(nl+1)L_{Z_q}$	$2L_G$	$2L_G + L_{G_T}$
[47]	$(n+1)L_G + L_{G_T}$	$(n+1)L_{Z_q}$	$(\|A_{\mathbb{A}}\|+1)L_G$	$(\|A_{\mathbb{P}}\|+1)L_G + L_{G_T}$
[36]	$(n+2)L_G + L_{G_T}$	L_G	$(\|A_{\mathbb{A}}\|+2)L_G$	$(2\|A_{\mathbb{P}}\|+1)L_G + L_{G_T}$
[35]	$(n+2)L_G + L_{G_T}$	$L_G + L_{Z_q}$	$(\|A_{\mathbb{A}}\|+2)L_G$	$(2\|A_{\mathbb{P}}\|+1)L_G + L_{G_T}$
[27]	$3L_G + L_{G_T}$	$L_G + 2L_{Z_q}$	$(n+1)L_G$	$(nl+1)L_G + L_{G_T}$
Our	$(n+2)L_G + L_{G_T}$	$(n+2)L_{Z_q}$	$(2n+1)L_G$	$(2n+1)L_G + L_{G_T}$

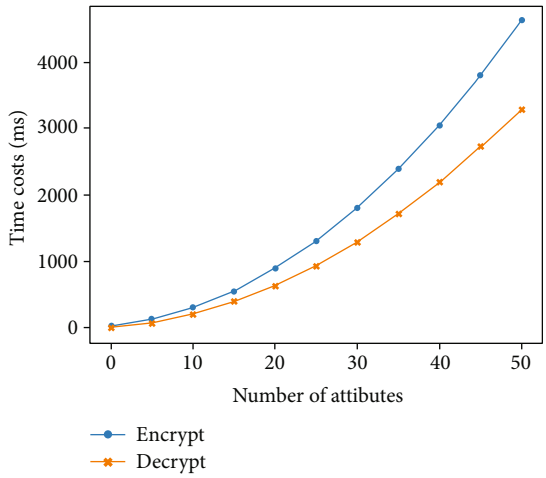


Figure 3: Performance analysis of our CP-ABE scheme.

depending on the number of attributes. Compared with other algorithms, the advantage of our algorithm is that it can support attribute update, policy hide, policy test, and other properties at the same time, while the efficiency of the algorithm does not decrease too much. Therefore, our scheme is more suitable for the crowdsourcing system with higher requirements for privacy protection.

7. Conclusion and Future Work

In this paper, we present a privacy protection mechanism for tasks in crowdsourcing, which realizes autonomous access control by adding blockchain to avoid a series of problems faced by central institutions. To solve the privacy of crowdsourcing tasks and access policies, we propose a new CP-ABE scheme with an expressive AND gate access structure that supports policy hiding and attribute updating. At the same time, we adopt a test algorithm based on a fully homomorphic cryptosystem to confidentially judge whether the worker's attribute lists match the hidden attributes policy in ciphertext or not before the decryption. Compared with previous schemes, our scheme has more advantages in flexibility, scalability, and privacy. In the future, we will consider an expressive and constant-size attribute-based access control based on blockchain.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors' Contributions

Bo Yang, Yanwei Zhou, Tao Wang, and Linming Gong contributed equally to this work.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (Grant nos. U2001205, 61772326, 61802241, and 61802242) and the Fundamental Research Funds for the Central Universities (Grant nos. GK202003079, GK202007033, and 2020TS087).

References

[1] L. Liu, J. Feng, Q. Pei et al., "Blockchain-enabled secure data sharing scheme in mobile-edge computing: an asynchronous advantage actor-critic learning approach," *IEEE Internet of Things Journal*, vol. 8, no. 4, pp. 2342–2353, 2021.

[2] K. Yu, L. Tan, S. Mumtaz et al., "Securing critical infrastructures: deep-learning-based threat detection in iiot," *IEEE Communications Magazine*, vol. 59, no. 10, pp. 76–82, 2021.

[3] K. Yu, Z. Guo, Y. Shen, W. Wang, J. C.-W. Lin, and T. Sato, "Secure artificial intelligence of things for implicit group recommendations," *IEEE Internet of Things Journal*, vol. 9, no. 4, pp. 2698–2707, 2022.

[4] F. Ding, G. Zhu, M. Alazab, X. Li, and K. Yu, "Deep-learning-empowered digital forensics for edge consumer electronics in 5g hetnets," *IEEE Consumer Electronics Magazine*, vol. 11, no. 2, pp. 42–50, 2022.

[5] L. Liu, C. Chen, Q. Pei, S. Maharjan, and Y. Zhang, "Vehicular edge computing and networking: a survey," *Mobile Networks and Applications*, vol. 26, no. 3, pp. 1145–1168, 2021.

[6] L. Zhao, J. Li, A. Y. Al-Dubai, A. Y. Zomaya, G. Min, and A. Hawbani, "Routing schemes in software-defined vehicular networks: design, open issues and challenges," *IEEE Intelligent Transportation Systems Magazine*, vol. 13, no. 4, pp. 217–226, 2021.

[7] L. Liu, M. Zhao, M. Yu, M. A. Jan, D. Lan, and A. Taherkordi, "Mobility-aware multi-hop task offloading for autonomous driving in vehicular edge computing and networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 1–14, pp. 1–14, 2022.

[8] L. Tan, K. Yu, F. Ming, X. Chen, and G. Srivastava, "Secure and resilient Artificial intelligence of things: a honeynet approach for threat detection and situational awareness," *IEEE Consumer Electronics Magazine*, p. 1, 2021.

[9] L. Tan, K. Yu, N. Shi, C. Yang, W. Wei, and H. Lu, "Towards secure and privacy-preserving data sharing for covid-19 medical records: a blockchain-empowered approach," *IEEE Transactions on Network Science and Engineering*, vol. 9, no. 1, pp. 271–281, 2022.

[10] K. Yu, M. Arifuzzaman, Z. Wen, D. Zhang, and T. Sato, "A key management scheme for secure communications of information centric advanced metering infrastructure in smart grid," *IEEE Transactions on Instrumentation and Measurement*, vol. 64, no. 8, pp. 2072–2085, 2015.

[11] Q. Feng, D. He, S. Zeadally, M. K. Khan, and N. Kumar, "A survey on privacy protection in blockchain system," *Journal of Network and Computer Applications*, vol. 126, pp. 45–58, 2019.

[12] J. Feng, L. Liu, Q. Pei, and K. Li, "Min-max cost optimization for Efficient hierarchical federated learning in wireless edge networks," *IEEE Transactions on Parallel and Distributed Systems*, p. 1, 2022.

[13] L. Zhao, H. Li, N. Lin, M. Lin, C. Fan, and J. Shi, "Intelligent content caching strategy in autonomous driving toward 6g," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–11, 2021.

[14] G. Wang, B. Wang, T. Wang, A. Nika, H. Zheng, and B. Y. Zhao, "Defending against sybil devices in crowdsourced mapping services," in *Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services, MobiSys 2016*, pp. 179–191, Singapore, 2016.

[15] S. Nakamoto, *Bitcoin: A peer-to-peer electronic cash system*, Technical Report, 2008, https://bitcoin.org/bitcoin.pdf.

[16] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *2007 IEEE Symposium on Security and Privacy (S&P 2007)*, pp. 321–334, Oakland, California, USA, 2007.

[17] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Advances in Cryptology - EUROCRYPT 2005, 24th annual international conference on the theory and applications of cryptographic techniques*, vol. 3494, pp. 457–473, Aarhus, Denmark, 2005.

[18] A. Ouaddah, A. A. E. Kalam, and A. A. Ouahman, "Fairaccess: a new blockchain-based access control framework for the internet of things," *Security and communication networks*, vol. 9, no. 18, 5964 pages, 2016.

[19] D. D. F. Maesa, P. Mori, and L. Ricci, "Blockchain based access control," in *Distributed applications and interoperable systems -17th IFIP WG 6.1 international conference, DAIS 2017, held as part of the 12th international federated conference on distributed computing techniques, DisCoTec 2017*, vol. 10320, pp. 206–220, Neuchatel, Switzerland, 2017.

[20] Y. Zhang, D. He, and K. R. Choo, "Bads: Blockchain-based architecture for data sharing with ABS and CP-ABE in iot," *Wireless Communications and Mobile Computing*, vol. 2018, 127836589 pages, 2018.

[21] S. Alansari, F. Paci, and V. Sassone, "A distributed access control system for cloud federations," in *37th IEEE International Conference on Distributed Computing Systems, ICDCS 2017*, pp. 2131–2136, Atlanta, GA, USA, 2017.

[22] J. P. Cruz, Y. Kaji, and N. Yanai, "RBAC-SC: role-based access control using smart contract," *IEEE Access*, vol. 6, pp. 12240–12251, 2018.

[23] A. D. Liu, X. H. Du, N. Wang, and S. Z. Li, "Blockchain-based access control mechanism for big data," *Journal of Software*, vol. 9, pp. 2636–2654, 2019.

[24] M. Jemel and A. Serhrouchni, "Decentralized access control mechanism with temporal dimension based on blockchain," in *14th IEEE International Conference on e-Business Engineering, ICEBE 2017*, vol. 4-6, pp. 177–182, Shanghai, China, 2017.

[25] Y. Zhou, Y. Guan, Z. Zhang, and F. Li, "A blockchain-based access control scheme for smart grids," in *2019 International Conference on Networking and Network Applications, NaNA 2019*, pp. 368–373, Daegu, Korea (South), 2019.

[26] D. D. F. Maesa, P. Mori, and L. Ricci, "A blockchain based approach for the definition of auditable access control systems," *Computers & Security*, vol. 84, pp. 93–119, 2019.

[27] S. Gao, G. Piao, J. Zhu, X. Ma, and J. Ma, "Trustaccess: a trustworthy secure ciphertext-policy and attribute hiding access control scheme based on blockchain," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 6, pp. 5784–5798, 2020.

[28] T. E. Gamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," *IEEE Transactions on Information Theory*, vol. 31, no. 4, pp. 469–472, 1985.

[29] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine- grained access control of encrypted data," in *Proceedings of the 13th ACM Conference on Computer and Communications Security, CCS 2006*, pp. 89–98, Alexandria, VA, USA, 2006.

[30] L. Cheung and C. C. Newport, "Provably secure ciphertext policy ABE," in *Proceedings of the 2007 ACM Conference on Computer and Communications Security, CCS 2007*, pp. 456–465, Alexandria, Virginia, USA, 2007.

[31] T. Nishide, K. Yoneyama, and K. Ohta, "Attribute-based encryption with partially hidden encryptor-specified access structures," in *Applied cryptography and network security, 6th international conference, ACNS 2008*, vol. 5037, pp. 111–129, New York, NY, USA, 2008.

[32] K. Emura, A. Miyaji, A. Nomura, K. Omote, and M. Soshi, "A ciphertext policy attribute-based encryption scheme with constant ciphertext length," in *Information security practice and experience, 5th international conference, ISPEC 2009*, vol. 5451, pp. 13–23, Xi'an, China, 2009.

[33] J. Lai, R. H. Deng, and Y. Li, "Fully secure cipertext-policy hiding CP-ABE," in *Information security practice and experience -7th international conference, ISPEC 2011*, vol. 6672, pp. 24–39, Guangzhou, China, 2011.

[34] V. Goyal, A. Jain, O. Pandey, and A. Sahai, "Bounded ciphertext policy attribute based encryption," in *Automata,*

Languages and Programming, 35th International Colloquium, ICALP 2008, vol. 5126, pp. 579–591, Reykjavik, Iceland, 2008.

[35] A. B. Lewko, T. Okamoto, A. Sahai, K. Takashima, and B. Waters, "Fully secure functional encryption: attribute-based encryption and (hierarchical) inner product encryption," in *Advances in cryptology - EUROCRYPT 2010, 29th annual international conference on the theory and applications of cryptographic techniques*, vol. 6110, pp. 62–91, Monaco / French Riviera, 2010.

[36] B. Waters, "Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization," in *Public Key Cryptography - PKC 2011 - 14th International Conference on Practice and Theory in Public Key Cryptography*, vol. 6571, pp. 53–70, Taormina, Italy, 2011.

[37] Y. Zhang, D. Zheng, and R. H. Deng, "Security and privacy in smart health: Efficient policy-hiding attribute-based access control," *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 2130–2145, 2018.

[38] J. Hao, C. Huang, J. Ni, H. Rong, M. Xian, and X. S. Shen, "Fine-grained data access control with attribute-hiding policy for cloud-based IoT," *Computer Networks*, vol. 153, pp. 1–10, 2019.

[39] J. Li, H. Wang, Y. Zhang, and J. Sheng, "Ciphertext-policy attribute-based encryption with hidden access policy and testing," *KSII Transactions on Internet and Information Systems*, vol. 10, no. 7, pp. 3339–3352, 2016.

[40] J. Bonneau, A. Miller, J. Clark, A. Narayanan, J. A. Kroll, and E. W. Felten, "Sok: Research perspectives and challenges for bitcoin and cryptocurrencies," in *2015 IEEE Symposium on Security and Privacy, SP 2015*, pp. 104–121, San Jose, CA, USA, 2015.

[41] L. Lamport, R. E. Shostak, and M. C. Pease, "The byzantine generals problem," *ACM Transactions on Programming Languages and Systems*, vol. 4, no. 3, pp. 382–401, 1982.

[42] N. Szabo, "Formalizing and securing relationships on public networks," *First Monday*, vol. 2, no. 9, 1997.

[43] F. Guo, Y. Mu, W. Susilo, D. S. Wong, and V. Varadharajan, "CP-ABE with constant-size keys for lightweight devices," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 5, pp. 763–771, 2014.

[44] R. L. Rivest, L. Adleman, and M. L. Dertouzos, "On data banks and privacy homomorphisms," *Foundations of secure computation*, vol. 4, no. 11, pp. 169–180, 1978.

[45] C. Gentry, "Fully homomorphic encryption using ideal lattices," in *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009*, pp. 169–178, Bethesda, MD, USA, 2009.

[46] D. Boneh, E. Goh, and K. Nissim, "Evaluating 2-dnf formulas on ciphertexts," in *Theory of cryptography, second theory of cryptography conference, TCC 2005*, vol. 3378, pp. 325–341, Cambridge, MA, USA, 2005.

[47] L. Ibraimi, Q. Tang, P. H. Hartel, and W. Jonker, "Efficient and provable secure ciphertext-policy attribute-based encryption schemes," in *Information security practice and experience, 5th international conference, ISPEC 2009*, vol. 5451, pp. 1–12, Xi'an, China, 2009.

Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 6833535, 16 pages
https://doi.org/10.1155/2022/6833535

Research Article

A Partitioned DAG Distributed Ledger with Local Consistency for Vehicular Reputation Management

Naipeng Li,[1] **Yuchun Guo**,[1] **Yishuai Chen**,[1] **and Jinchuan Chai**[2]

[1]*School of Electronics and Information Engineering, Beijing Jiaotong University, Beijing 100044, China*
[2]*National Railway Track Test Center, China Academy of Railway Sciences, Beijing 100015, China*

Correspondence should be addressed to Yishuai Chen; yschen@bjtu.edu.cn

Received 11 November 2021; Revised 3 March 2022; Accepted 8 March 2022; Published 23 March 2022

Academic Editor: Qingqi Pei

Vehicular reputation maintenance with distributed ledger is aimed at establishing trust among vehicles randomly meeting in a Vehicular Ad-hoc Network (VANET). It is, however, challenging in VANET, as congested areas in road networks, brought by traffic tides or accidents, challenge the ledger performance. Meanwhile, the reputation update is highly dependent on transaction consensus of the distributed ledger. To solve the problem, this paper proposes deploying directed acyclic graph- (DAG-) based distributed ledgers on vehicles, which use the vehicular distribution to adapt the unpredictable reputation update. Specifically, we first propose a partitioned DAG-based distributed ledger to manage vehicular reputation in partitioned VANET. Secondly, we introduce a novel reputation evaluation method to encourage vehicles to contribute to VANET interaction and ledger consensus maintenance, which can remedy the topology churn of the ledger network due to the mobility of VANET. Finally, we design a reputation update method based on the consistency of transactions in the partition to facilitate trust establishment. Experimental results on a real-world dataset show that the proposed ledger and reputation update method is effective and feasible in the large-scale dynamic VANET.

1. Introduction

With the rapid evolution of Vehicular Ad-hoc Networks (VANETs) and intelligent technology, intelligent vehicles have further demands for exchanging information with surrounding smart objects like other intelligent vehicles, smart traffic lights, and Road Side Units (RSUs) [1, 2]. However, the unique features of VANETs, such as high mobility and volatility, make the antiattack and privacy protection become major concerns [17]. Establishing the trust for received information or connectable nodes needs a vehicular reputation management system. Recently, due to providing privacy protection and decentralized trust among unfamiliar vehicles [17], employing Distributed Ledger Technologies (DLTs), such as blockchain, for vehicular reputation management has become a hot research topic [1, 4, 6, 9].

Nevertheless, a vehicular reputation management system with distributed ledger also faces two challenges. The first challenge is ledger maintenance. To maintain the consistency of the transaction, the distributed ledger needs consensus mechanisms like Proof of Work (PoW) [7] or Proof of Stake (PoS) [8] in the blockchain ledger. However, it is challenging for the VANET node to satisfy the requirements of consensus mechanisms, e.g., computing power and stable communication route [6]. The second challenge is trust establishment. Evaluating the other's reputation is an excellent way to establish trust for an interaction. A node can evaluate reputation for one time when the interaction begins rather than assessing the context of each exchanging message. However, the VANET, a decentralized network, challenges the reputation update.

Existing studies for vehicular reputation management are mainly based on the blockchain platform [6, 20]. There are

two kinds of nodes in a blockchain platform, the full node and the simplified node. The former keeps the entire blockchain ledger and composes the backbone blockchain network. The latter only creates and consumes transactions that store reputation ratings on others or some nodes' historical behaviors but do not maintain the ledger. The researchers use the RSUs as the full nodes to satisfy computational and stable bandwidth resources and the vehicles as the simplified nodes to quantify and encapsulate the vehicular interactive behaviors into the transactions [3, 5, 6, 9, 17, 20]. However, it is tricky for an RSU-based blockchain ledger to guarantee timely reputation updates. Vehicles can only connect and issue transactions to nearby vehicles or RSUs [21]. When a traffic jam or accident occurs, the vehicle density will dramatically increase in the spot and overload adjacent RSUs, which delays the consensus of the ledger and update of reputation.

Distributed ledger with a directed acyclic graph- (DAG-) based architecture seems to have better scalability than traditional blockchain architectures, which provide promising solutions to solve the issue of uneven reputation update workload. Firstly, the DAG-based distributed ledger (DDL) has a higher throughput than the blockchain. Instead of competing for the block-level consensus, DDL verifies and approves the transactions in different parts of the ledger network parallelly with homogenous nodes. A DDL node is required to approve recently issued transactions to help these pending transactions receive enough approvals quickly. Secondly, some DDL systems, such as the Tangle [10], are designed for the IoT scenario composed of low-resource nodes, e.g., vehicles. Although some works have been done on DDL-enabled VANET, they mainly carried out the feasibility tests [13, 14], and two key issues have still not been properly solved. (1) DDL needs to sort the parallelly approved transactions [11], but this brings about the complexity of the design. (2) DDL requires enough nodes to ensure the high throughput and security of the ledger. It is a challenge to guarantee this in the mobile VANET.

To address the above challenges, this paper proposes a partitioned DDL with local consistency for reputation management in VANETs. We designed the ledger and reputation update method based on the insight of spatiotemporal sensitivity. On the one hand, limited by the sensing range, some interactions occur only between the nearby vehicles [12]. For example, the traffic lights at an intersection are only helpful for nearby vehicles that also only these vehicles can check the trustworthiness of light information instantly. On the other hand, unlike financial applications, reputation management in VANETs does not require a transaction to reach a consensus among all nodes. Specifically, a vehicle needs someone's reputation only when establishing trust with a meeting vehicle, so the vehicles with different routes do not need each other's reputation in practice. We argue that a vehicle could independently choose which vehicles to follow according to its own itinerary needs. Ensuring the related transactions are consistent among the vehicles in a particular range is enough. The main contributions of this paper include the following.

(i) We design a partitioned DAG-based distributed ledger based on the Tangle architecture for reputation management in the VANETs

(ii) We present a vehicular reputation evaluation method by assessing the node's interactive quality and the contribution to maintaining the transaction consensus

(iii) We propose a reputation update method based on local transactional consistency to reduce the update latency and improve the trust establishment

(iv) To demonstrate the effectiveness of the proposed ledger framework, we conduct the simulations on a real-world dataset, and simulation results reveal that the proposed framework is effective and feasible in the large-scale VANETs and the reputation update delay also converges when the VANETs size is growing exponentially

The remainder of this paper is organized as follows. Section 2 surveys the existing reputation system for VANETs and summarises the related DDL works. Section 3 describes the framework overview and system model. The details of the proposed ledger are carried out in Section 4. Section 5 proposes the reputation definition and update method. We conduct simulations and discuss the numeric results in Section 6. Finally, Section 7 concludes the paper.

2. Related Work

In this section, we classify existing DLT-based reputation/trust management systems and present existing works about expanding the throughput of the distributed ledger systems.

2.1. DLT-Enabled Reputation/Trust Management in VANET. In a DLT-enabled reputation/trust management system in VANET, the distributed ledger helps the vehicles build the consensus of the data, trust, or opinion related to the participant's reputation. The state-of-the-art systems can be classified into two categories: access-to-trust system and evaluate-to-trust system. We introduce them in detail as follows:

(1) Access-to-trust system

Access-to-trust systems require that any nodes get permission first before they are considered trustworthy and build trusted communication [15, 16]. In general, the systems maintain a white list or black list to control this communication permission. Lu et al. [17] utilize two blockchains to record the workflow of the Certificate Authority (CA) and management history of all vehicles separately, and the former monitors the credibility of the CA, and the latter maintains the reputation of the system nodes and assists the CA in the issuance of certificates. With the development of smart contract [18], Javaid et al. [4] and Liu et al. [19] all adopted the smart contract to control the registering and access of the honest vehicles, and only the vehicles with permission can communicate with each other freely. Furthermore, Wang et al. [20] use smart contracts to manage the access of vehicles, and vehicles can obtain the evaluation results of the other vehicle's reputation by submitting the request to the specific smart contract. To overcome the dynamic network size of the VANET, Javaid et al. [4] modified the Proof of Work (PoW) mechanism to

adapt to the incoming traffic generated by the vehicles. Kudva et al. [22], Khalid et al. [23], and Liu et al. [19] build their systems based on the consortium blockchain platform, which operates only with a fixed number of preauthorized nodes so it can assign to some powerful nodes to keep the performance of their ledger system. All of the above works focus on communication efficiency but lack investigation into the incentives of the nodes in a decentralized system.

(2) Evaluate-to-trust system

Evaluate-to-trust systems directly assess the credibility of the transmitted data, including based on the voting of different data sources about the same event, the reputation of the source sender, or even the empirical probability of the event occurrence. Kang et al. [24]use the interaction frequency, event timelines, and trajectory similarity of the source vehicles to evaluate their message's credibility; Road Side Units (RSUs) collect the reputation opinions or other shared data and ensure the consensus of these records by PoW. To solve similar problems, Yang et al. [6] use the location of the sending vehicles to evaluate the message's credibility, and RSUs collect multiple messages that report the same event from different vehicles, calculate the sending vehicles' offsets, and add them to the blockchain through a consensus mechanism combining PoW and Proof of Stake (PoS). Based on the above work, Lu et al. [1] required the vehicles, in addition, to continue to collect the opinions on the event after receiving it from its initiator and also to query the initiator's reputation from the RSUs to compute the event's credibility, which will eventually be transmitted back to the RSUs to update the initiator's reputation. In these systems, the evaluation of the reputation is conducted after the nodes share messages. Unlike the above systems, Li et al. [25] proposed an active detection method based on the probe to find the possible misbehavior nodes before the sharing, and the system divided the VANET into multiple fixed partitions and set some fixed powerful servers to provide stable blockchain services. However, although all the above works allow the vehicles to self-assess the message's credibility or vehicle's trustworthiness for specific interaction, they still rely on the RSUs to provide and update the vehicles' reputation.

The most existing DLT-based reputation/trust management systems adopted the blockchain platform as their infrastructure framework. They implemented the RSUs as the miner (running the backbone network of the blockchain) because they have better computing resources and more stable network links than the vehicles in VANET. However, it is easy to cause the delay of reputation update due to the limitation of the RSU bandwidth and block size [6] when the inflow traffic increases. Diallo et al. [26] and Zhang et al. [27] have tried to reduce the update delay by using other consensus technologies, such as Practical Byzantine Fault Tolerance (PBFT). However, PBFT has a high communication complexity, and the consensus cluster should not be too large. Therefore, there are no suitable flexible frameworks that can simultaneously cope with the dynamic network topology, fragility network connection, and the lower update delay requirements in VAENT.

2.2. The DAG-Based Distributed Ledger. DAG ledgers potentially offer many advantages over traditional blockchain architectures for DLTs, including scalability and faster transaction speeds (Ferraro). Many DAG-based DLT projects, such as NANO [28], Byteball [29], and Tangle [10], have been in operation for many years and have been tested in practical applications. DAG ledgers organize transactions according to the DAG structure instead of packing them into the block. The consensus is conducted in parallel and runs over the transaction level with stochastic attachment mechanisms instead of constructing a chain of blocks. All transactions must be strictly ordered by their timestamp, which forces the above methods to adopt an additional puissant and centralized component, such as the coordinator in the Tangle, to check and determine the order of all transactions. Bartolomeu et al. [14] deploy the Tangle in VANET, and their experiment result shows that the transaction confirmation delay has been significantly reduced than the blockchain solution, and the performance is comparable with Tangle's main network. However, they only validated the feasibility of DAG-based consensus deployment on VANET and there is no further research on specific VANET applications. In terms of reputation management in VANET, Zhang et al. [27] and Kang et al. [24] both try to apply the DAG structure in subregions that a miner covered to improve transaction processing speed in subregions, but their ledger is still based on the blockchain platform.

Therefore, this paper designs a DAG-based vehicular distributed ledger and implements it with the Tangle architecture to optimize reputation management in VANET.

3. Application Overview

In this section, we first present the application scenarios of sharing in VANET and the interactive model and then introduce fundamental concepts of the Tangle project as backdrop. For the clarity of the following discussion, the key notations are summarized in Table 1.

3.1. Application Model. Figure 1 shows the overview of our reputation maintenance framework where the VANET entity, including vehicles and RSUs, arises sharing interaction and the vehicular reputation, maintained by a DDL. The framework contains two layers: the sharing and vehicular reputation management layers. We require a node's reputation to be calculated by auditing the node's history interactive behavior that accumulates in the sharing layer, and the node's historical interaction is packaged into the transaction. A node will generate a transaction based on its last interaction and attach it to the DDL in the reputation management layer. To ensure all the nodes can run the DDL equally, we assume that every node has an essential computational resource and can hold a full copy of the ledger.

Another important assumption is that the peer-to-peer interactive application in VANET scenario, such as environmental awareness, is mostly geographically independent. Take the traffic density perception at an intersection as an example; Figure 1 shows that the nodes around the intersection S_1 can be seen as a subpartition of VANET, and S_1

Table 1: Notions.

Symbol	Definition	Symbol	Definition
p_i	A node that can interact with others and issue transaction	μ	The reputation
$p_{j\longrightarrow i}$	The node answered the request of p_i	σ	The metric of interactive quality
$y_{j\longrightarrow i}$	The received originally rating of $p_{j\longrightarrow i}$	η	The metric of consensus contribution
ψ_i	The transaction	R^+, R^-, R^*	The calculated ratings of positive interaction, negative interaction, and consensus contribution
$\psi_{j\longrightarrow i}$	The transaction directly approved by ψ_j	w_i	The weight of ψ_i
$\psi_{j\longrightarrow m\longrightarrow i}$	The transaction indirectly approved by ψ_j	N_c	The set of all the active nodes in a period, $\lvert N_c\rvert$ is the number of these nodes
$\{\psi\}^i$	The set of transactions that contains the historical interactions of p_i	$\mathscr{A}_i$	The cumulative weight of ψ_i
$\mathscr{L}(t)$	The set of tips at time t	Θ	The threshold of local consistency

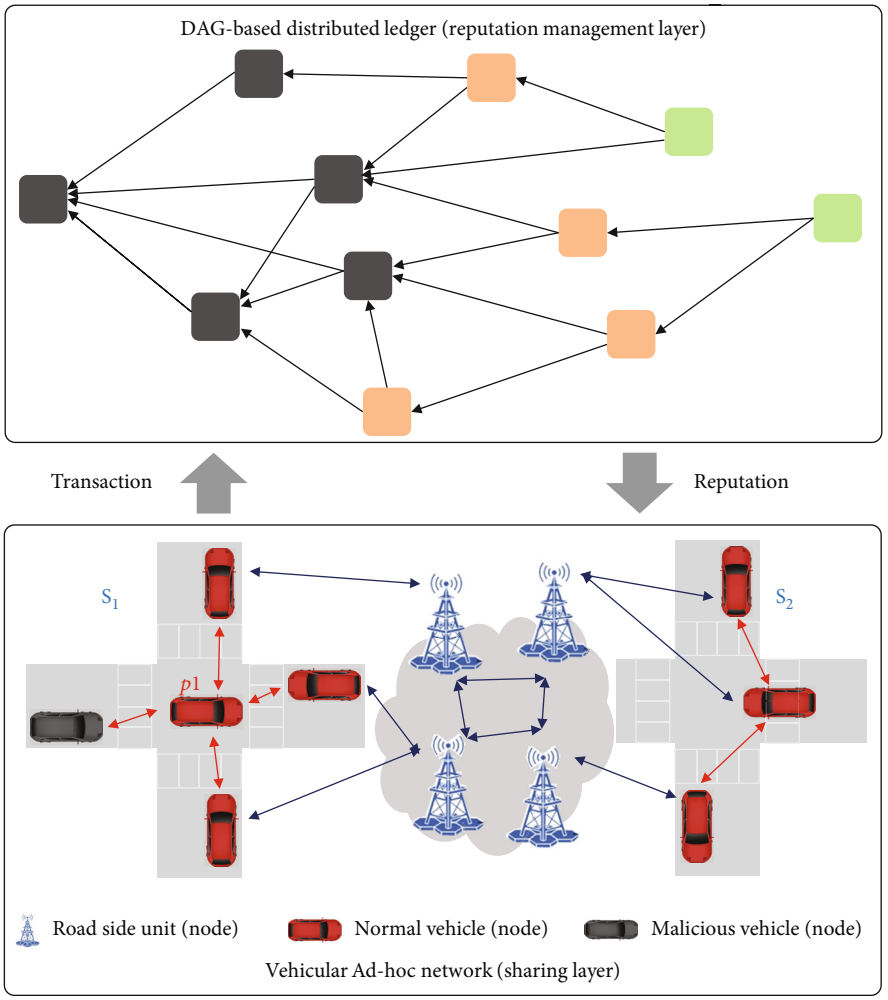


Figure 1: Overview of trust maintenance with a DAG-based distributed ledger.

consists of at least $n \geq 5$ vehicles and $m \geq 2$ RSUs. It is acceptable that all the traffic data contained in the message perceived by the p_1 can only be verified by the nodes in the same partition (that is, S_1). So, we can have $S = \{S_1, S_2, \cdots, S_K\}$, where S is the entire VANET and consists of K partitions. These partitions are connected by the RSUs, and any node belongs to at least one partition. Obviously, some applications need the nodes in different partitions to

cooperate, such as route plans, and we will discuss these more complicated scenarios in future work.

3.2. Interactive Model. In terms of most applications in VANET, we consider two typical interactive models, sharing and cooperating.

3.2.1. Sharing Model. To inform some warnings or share some knowledge, some vehicles may positively broadcast unencrypted information. Figure 2 shows that there are n nodes ($n \geq 2$, Figure 2 gives V_1 and V_2 as an example) that broadcast the knowledge of a specific event E at the same time. If a node has interests in E, it can collect a message set $\mathrm{Msg} = \{\mathrm{msg}_1, \mathrm{msg}_2, \cdots, \mathrm{msg}_n\}$. Then, the nodes need to assess the credibility of each of the received messages and evaluate a possible result of event E (because they do not know the truth about it). Based on the calculated result of event E, the node can judge whether the received messages are the same or contrary to the calculated result of E and evaluate a rating about interactive behavior of the information source node.

3.2.2. Cooperating Model. To get advice and assistance, some vehicles may ask for help from others. Figure 3 shows a general process, a node (p_1) in need broadcasts its request to the surrounding nodes first, and then, others may answer the request at their will. If there are n answered nodes, p_1 can receive some responses; for easy understanding, we also use an answering message set $\mathrm{Msg} = \{\mathrm{msg}_1, \mathrm{msg}_2, \cdots, \mathrm{msg}_n\}$ for the unified presentation with the same as sharing mode. The difference is only the p_1 can judge whether the answered node provided an effective result and evaluate the rating of interactive behavior for each answered node. To simplify the discussion, we assume that all requested nodes are honest.

3.3. DAG-Based Distributed Ledger Model. All the transactions are stored in a DAG architecture and are represented by vertexes (dark grey rectangles shown in Figure 4). A new transaction needs to approve several old (especially issued recently) transactions, and the approval relationships are represented by edges. A new edge (from the new vertex to the old vertices that stand for the selected previous transactions) is added while a new transaction is issued simultaneously. This adding process is also called transaction attachment. Some key concepts of the DDL are introduced as follows.

Transaction Approval: as shown in Figure 4, there are two types of approval relationship, direct and indirect approval. Direct approval is represented by a directed edge and indirect approval is represented by a path that consists of several transactions and direct edges connecting them. For example, E_1 indicates that ψ_c approved ψ_a, E_2 indicates ψ_b approved ψ_c, and E_4 and E_5 indicate ψ_d and ψ_e approved ψ_a indirectly.

Tip: transaction that has received no approval is called tip. A tip may be a newly issued transaction (e.g., ψ_e or ψ_f) or an old transaction but has not been approved even once (e.g., ψ_c). Define $\mathscr{L}(t)$ to be the set of tips at time t. In general, an issuing node is suggested to select tips from $\mathscr{L}(t)$ to approve, so the size of $\mathscr{L}(\mathrm{t})$ determines the growth and health of the DDL. Once a transaction is approved, it is no longer a tip, but it also needs to accumulate enough direct and indirect approval to be regarded as secure and final confirmation.

Cumulative weight: cumulative weight (CW) is a metric for measuring how trustworthy a transaction is for security consideration. Suppose $\mathscr{A}_i$ presents the CW of ψ_i and is calculated by the weights of all the transactions, including directly and indirectly, that approved ψ_i. In general, when a transaction's CW reaches (only monotonically increasing) a threshold, we say that it is confirmed, which also means it is correct and immutable. We will introduce the details of how the weights increase and threshold setting in Section 5.

4. Partitioned DAG-Based Distributed Ledger in VANET

This section first introduces how to record the details of the interaction into a transaction and how to verify. Then, we present the definition of the CW considered under the partitioned DDL. To deal with the fragility of connection and topology in VANET, we introduce a local consistency threshold and an extended tip selection algorithm to improve the throughput of transactional consensus while ensuring the ledger's security.

4.1. Historic Interaction. We need a way for the node to obtain others' reputations when establishing the trust. The existing works usually update the "balance" or "bias" of reputation. However, these solutions do not allow nodes to adjust reputations according to different situations. We consider the "auditing" method, which records the interaction details into transactions, and nodes calculate the reputation for anyone in their desired ways when needed.

Two interactive models have the different roles of the node to record each interactive detail. For a sharing model, RSUs can generate the transaction to record an interactive event in the partition it is deployed. If there are many RSUs, a rotation method can balance the workload of transaction generation. For a cooperating model, the requested node is responsible for generating the transaction when finishing a round of interaction. Take the cooperating model as the example, and we define the transaction as shown in Figure 5.

$$\Psi \coloneqq \langle d, \gamma, s, t \rangle, \tag{1}$$

where d is the interaction data, γ is the transaction approval data, and s and t represent the encrypted script and timestamp, respectively. The detail of d is

$$d \coloneqq \left\langle p_i, \left\{ p_{j \longrightarrow i} \right\}, \left\{ y_{j \longrightarrow i} \right\} \right\rangle, \tag{2}$$

where p_i denotes the issuing node that is also a requested node; $\{p_{j \longrightarrow i}\}$ and $\{y_{j \longrightarrow i}\}$ refers to all the nodes that answered the p_i and the corresponding ratings, respectively. We use a Bayesian method [6] to inference the $\{y_{j \longrightarrow i}\}$. γ represents the transaction approval relationship and composed of the following:

$$\gamma \coloneqq \langle \{\psi_{i \longrightarrow m}\}, \{\ \text{PoW Nonce}\ \}, \text{PoW target} \rangle, \tag{3}$$

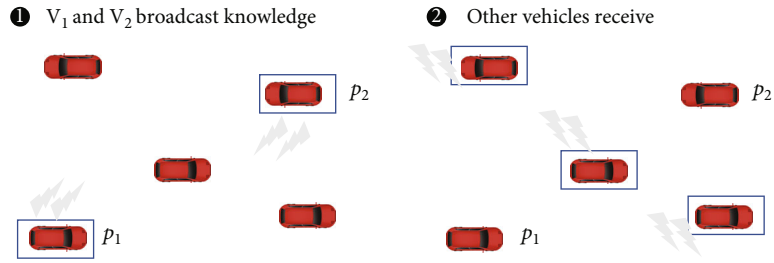


Figure 2: Two vehicles sharing data with others.

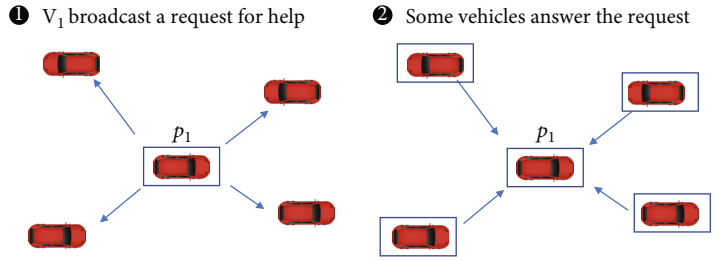


Figure 3: A vehicle asks and establishes cooperation to others.

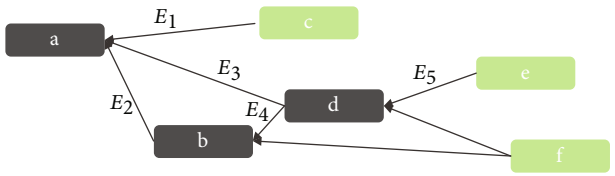


Figure 4: Representation of approval between the transactions.

where $\{\psi_{i\longrightarrow m}\}$ denotes the selected and approved transaction set. PoW Nonce and PoW target are parameters for a PoW mechanism that used to prevent malicious nodes from issuing large-scale false transactions to attack the ledger network. In general, the target is set to a small value and does not bring heavy PoW workload for a vehicle.

4.2. Transaction Verification. Each node needs to verify the newly received transaction to avoid malicious and fake transaction attacks. Moreover, the transaction weight is also calculated if it passes the verification.

The verification includes three steps; take ψ_i as an example; they are as follows:

(1) In d, whether the p_i and $\{p_{j\longrightarrow i}\}$ exist

(2) In γ, whether all the selected transactions in $\{\psi_{i\longrightarrow m}\}$ exist, and verify their PoW NONCE

(3) If step 1 and step 2 pass, calculate the transaction's weight

For step 1, we assume that all the interactive messages can be recorded with some methods, such as the smart contract among the reference nodes.

For step 2, the validation of γ requires that the selected and approved transactions must have been transmitted to all other nodes (at least in several nearby partitions) before they can be approved. In fact, the most recent tips that the issuing node cab select at time t can only be issued at $t-h$, where h, called the waiting period [10], includes both the PoW time and the minimum transmission time for transaction transmission to most nearby nodes.

For step 3, any node that received the transaction needs to calculate its weight. The weight is calculated only once and stored only at the local.

In our framework, we define the issuing node's interactive behaviors as the weight of its issuing transaction. Some nodes invest a lot, including frequently and actively responding or issuing transactions (verifying and approving the other transactions to assist the DDL). For a peer-to-peer data sharing system, it is obvious that the nodes working hard and getting higher interaction ratings should have more credibility. So, we define the weight of the transaction issued by p_i as

$$w_i = \epsilon_1 e_i + \epsilon_2 \sum_{k\in\{\psi\}^i} y_k, \tag{4}$$

where e_i denotes the number of valid transactions issued by p_i and $\{\psi\}^i$is a set of transactions that contain the historical interactions of p_i. y_k refers to the corresponding rating and $\epsilon_1 + \epsilon_2 = 1$. Obviously, e_i and the size of $\{\psi\}^i$ are changed over time, so we only calculate the weight at once when it is issued.

4.3. Cumulative Weight. The CW of a transaction can be calculated by the sum of the weights from all its successor transactions. If we set $w = 1$ for each transaction, the CW represents how much approval this transaction achieved.

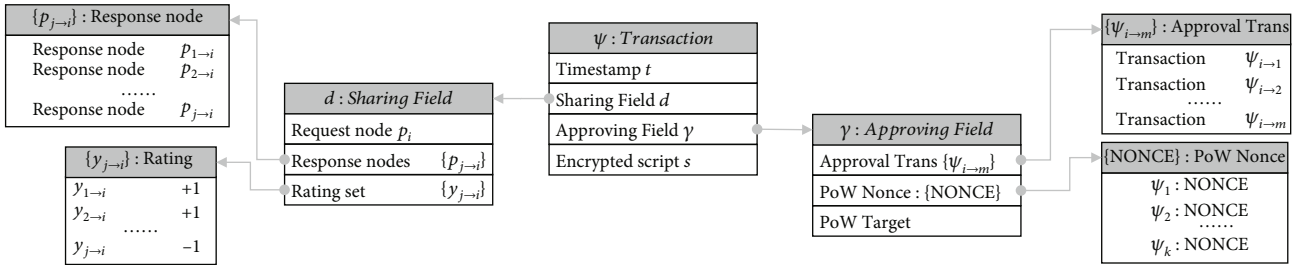


Figure 5: Data structure of the transaction.

For ψ_i, its CW is

$$\mathscr{A}_i = \sum_{j\in\{\psi_{j\longrightarrow i},\psi_{j\longrightarrow m\longrightarrow i}\}} w_j, \tag{5}$$

where $\{\psi_{j\longrightarrow i},\psi_{j\longrightarrow m\longrightarrow i}\} = \{\psi_{j\longrightarrow i}\} + \{\psi_{j\longrightarrow m\longrightarrow i}\}$ denotes a set of transactions that directly and indirectly approve ψ_i, w_j is the weight of ψ_j, and any transaction counts once. Figure 6 demonstrates an example of how a transaction accumulates weight, where the rectangle represents transaction, and the number outside the parentheses in the rectangle represents transaction weight. The transaction c selects a and b which, respectively, are represented by the edges E_2 and E_3, so a and b both accumulate a direct approval, and their accumulative weight is added by the weight of c (it is 12 in Figure 6). b and c approve a directly; d and e approve a indirectly. If $w_b = 1$, $w_c = 12$, $w_d = 2$, and $w_e = 1$, then $\mathscr{A}_a = 5 + 1 + 12 + 2 + 1 = 21$.

4.4. Local Consistency Threshold. The existing DDLs own independent components to strictly sequence transactions, such as Tangle's coordinator [11]. Strictly ordered transactions are very important for financial applications to defend against the double attack. However, managing reputation does not need to be strict. We argue that a transaction, approved by enough but not be strictly ordered globally, is secure for auditing reputation for the reference nodes. DDL defines that a transaction reaches consensus when it achieves enough approval, making the transaction difficult to tamper. Combining the above discussion of the CW, it is easy to realize that the transaction accumulated a high CW which can be seen as secure and cannot easily be falsified. We need to find how much CW can ensure the security of transactions for auditing reputation.

The local consistency threshold is proposed to enable the ledger node to judge whether the transaction is secure by itself instead of relying on the third component. Before introducing the details of the local consistency threshold, let us clarify two essential and reasonable assumptions in this paper: (1) if two nodes do not meet within their trips (refer to the trip where they need the VANET applications), they do not need the reputation of each other. First, in terms of the collaboration, nodes only care about the nodes running in several nearby partitions; e.g., the traffic light data of a specific road section is only meaningful and can only be verified by the nearby nodes. Then, in terms of time, the node's behavior that is too old is no longer suitable for evaluating reputation for some security issue consideration [30]. (2) We consider that if a node has always been well-behaved in issuing a transaction (verification and selecting tips) and participating in the interaction (sharing and cooperating), then the ledger will eventually accept the transaction issued by this node with a high probability.

According to the above discussion, we set a period h, limiting all transactions' valid time. When a period ends, the ledger will be reset. We also define a set of transactions N_c, containing all the transactions for a node cared. However, the fewer nodes will lead to security risks for a distributed ledger, so we bring the workload of the nodes to increase the transaction's weight. Now, we can focus on the transaction consensus in $n(n \geq 1)$ partitions and define Θ as the local consistency threshold to assist nodes to infer the transaction's credibility, and it can be expressed by

$$\Theta = \frac{L}{|N_c|} \sum_{j\in N_c} \max_{k\in\{\psi_k\}^{(j)}} \left\{ w_k^{(j)} \right\}, \tag{6}$$

where $\{\psi_k\}^{(j)} = \{\psi_1, \psi_2, \cdots, \psi_k\}^{(j)}$ denotes the set of all the transactions issued by node p_j, $w_k^{(j)}$ is the weight of the $\psi_k^{(j)}$, so $\max_{k\in\{\psi_k\}^{(j)}} \{w_k^{(j)}\}$ presents the largest weight of the transaction issued by p_j. N_c represents a set of active nodes (issued transactions in a period) in n partitions cared about; $|N_c|$ is the number of these nodes. L is a positive hyperparameter that controls the evolution speed of the ledger, and the nodes can adjust it to cope with the scale change of the interest partitions. Nodes could make their judgments on whether the transaction is confirmed. Algorithm 1 introduces the detail of the transaction consensus process.

4.5. Tip Selection Algorithm. We propose a modification to the attachment mechanism of the Tangle. This modification ensures the transaction is verified and secure in the partitioning VANET and preserves essential features of the Monte Carlo Markov Chain (MCMC) selection algorithm [10].

Firstly, the issuing nodes need to verify whether or not the transaction selected for approval is mutually consistent with each other. If detecting an inconsistency, the tip selection process must be rerun until a consistent $\mathscr{L}(t)$ is found. In addition, creating m independent random walks in a path of DAG contains the transactions issued by the nodes running in the interested partitions in the current period. The walk starts at the genesis site and moves along the edges. The

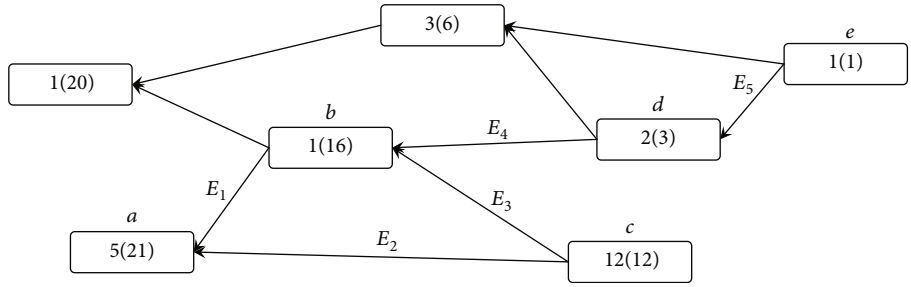


Figure 6: A part of ledger.

input: A tip ψ_i,local consistency thresholdΘ, current tip set $\mathscr{L}(t)$, sub-DAG consisted of the confirmed transactions G_t^c and sub-DAG consisted of the unconfirmed transactions G_t^u
output: Updated ledger $G_{t+1,}^c G_{t+1,}^u$ and the updated tip set $\mathscr{L}(t+1)$
1 Extract all approval transactions $\{\psi_{i\longrightarrow m}\}$packaged in $\{\psi_i\}$;
2 for each $\{\psi_{i\longrightarrow m}\}$**do**
3 calculate and add weight ω_i to ψi;
4 add ω_i to Cumulative Weight $\mathscr{A}_{i\longrightarrow m}$of $\psi i_{i\longrightarrow m}$;
5 if $\psi_{i\longrightarrow m} i\pi\ \mathscr{L}(t)$to G_t^u
6 add ψ_i to $\mathscr{L}(t)$to to G_t^u
7 move $\psi_{i\longrightarrow m}$from $\mathscr{L}(t)$to G_t^u
8 else
9 wait for a punish time;
10 add ψ_i to $\mathscr{L}(t)$;
11. Extract the transaction $\{\psi_{i\longrightarrow m\longrightarrow\eta}\}$that indirectly approved by
12 for each $\psi_{i\longrightarrow m\longrightarrow\eta}$ in $\{i_{i\longrightarrow m\longrightarrow\eta}\}$**do**
13 add w_i to $\mathscr{A}_{i\longrightarrow m\longrightarrow n}$ of $\psi_{i\longrightarrow m\longrightarrow n}$;
14 **if** $\psi_{i\longrightarrow m\longrightarrow n} > \Theta$ then
15 move $\psi_{i\longrightarrow m\longrightarrow n}$ to G_t^c
16 **final**;
17 **return** $G_{t+1}^c, G_{t+1}^c, \mathscr{L}(t+1)$;

Algorithm 1: Consensus algorithm for transactions on partitions.

Table 2: Data field in NSL.

Field	Symbol
Node ID	p_i
Iterative consensus metric	$\langle t, R^* \rangle$
Cumulative interactive metric	$\langle t, R^+, R^- \rangle$

probability of stepping along an edge from site ψ_j to site ψ_k is

$$f\left(-\alpha\left(\mathscr{A}_j - \mathscr{A}_k\right)\right), \tag{7}$$

where $f(\cdot)$ is an exponential function and α is a positive constant. $\mathscr{A}_j$ and $\mathscr{A}_k$ are the CWs of ψ_j and ψ_k, respectively. For a new transaction, suppose the walk should reach $Q(\mathrm{Q} \geq \mathrm{m})$ tips, and the issuing node selects the tip satisfying

$$\min_{l<Q}\left\{\sum_{i=1}^{m}\left|\Theta - \mathscr{A}_i^{(l)}\right|\right\}, \tag{8}$$

where $\mathscr{A}_i^{(l)}$ denotes the CW of transactions directly approved by ψ_l that is the end of a walk. Finally, we also need to walk to the m tips that their selected transactions are about to be or just recent security.

5. Reputation Update with Transaction Local Consistency

In this section, we first present the definition of each part of the vehicular reputation. Then, we will describe the reputation update method based on the partitioning and valid period.

5.1. Vehicular Reputation. Our DDL requires the vehicle to be a node and contributes to the ledger maintenance. Therefore, we argue that the expression of reputation needs to contain the node's behaviors in ledger maintenance and VANET interaction. Interactive behavior refers to the quality of data that the node shares when interacting, and it is stored in the relevant transactions issued after each interaction occurs. The maintenance behavior of the ledger, we also

input: *NSL*, a confirmed transaction ψ_i, issued node p_i and the node that waited to interact with p_j
output: The updated reputation μ of p_j
1/∗∗∗ Update NSL ∗∗∗/ Extract response nodes $\{p_{j\longrightarrow i}\}$ and corresponding interactive ratings $\{y_{j\longrightarrow i}\}$;
2 **for** each $\{p_{j\longrightarrow i}\}$ in $\{p_{j\longrightarrow i}\}$**do**
3 update $NSL(p_{j\longrightarrow i})[\langle t, R^+, R^- \rangle]$ based on the corresponding interactive rating $\{y_{j\longrightarrow i}\}$;
4 Extract R^* in ψ_i and update $\langle t, R_* \rangle$ to $NSL(p_i)$;
5/∗∗∗ Calculate Reputation ∗∗∗/ Obtain the current time t_u;
6 Calculate quality σ based on $NSL(p_j)[\langle t, R^+, R^- \rangle]$;
7 Select an exponential function to calculate $\eta = f(-\beta R^*)$;
8 Obtain the reputation μ;
9 **final**;
10 **return**μ;

Algorithm 2: Reputation update algorithm.

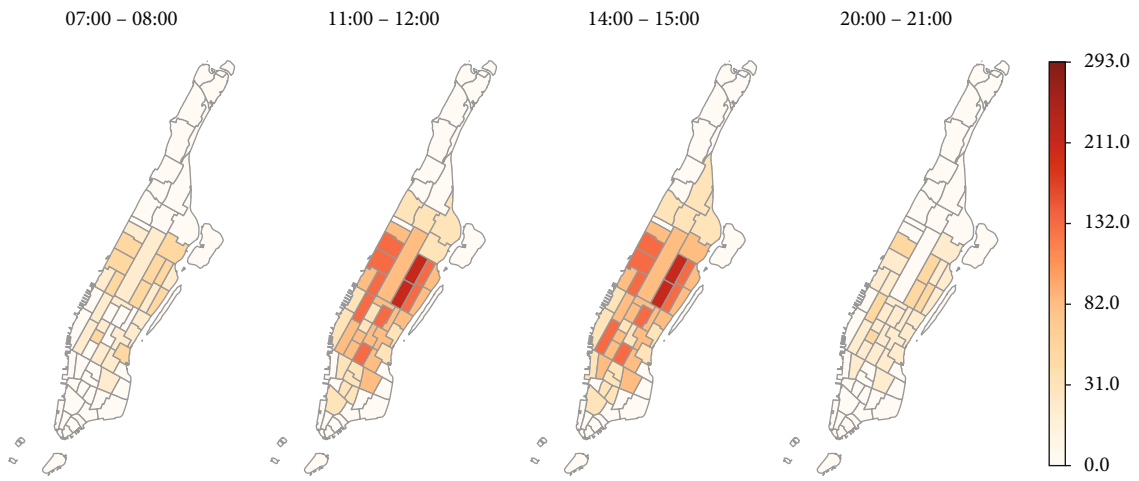


Figure 7: Distribution of the partitions and the vehicle heat in different hours in Manhattan.

call consensus contribution, refers to the node's performance in ledger maintenance and is calculated by other nodes when they verify a new transaction issued by the node.

In summary, when omitting the symbol of interest partitions and valid period, the reputation is

$$\mu = \tau_1 \sigma + \tau_2 \eta, \tag{9}$$

where σ and η represent the interactive quality and consensus behavior, respectively; we will discuss them in the following subsection. $\tau_1 + \tau_2 = 1$; they are used to adjust the ratio of the two measures in different scenarios. For example, in the initial phase of each valid period, the ledger needs as many as possible nodes to join to maintain the transaction consensus, at this case, $\tau_2 > \tau_1$.

5.1.1. Interactive Quality. To simply the discussion, we only consider the cooperating model because it can easily extend to the sharing model. Assume that a node has M_1-positive ratings and M_2-negative ratings, so that all the received ratings $M = M_1 + M_2$. Let $R_m^{(b+)}$ be the positive rating that p_b received at mth response at time t and $R_m^{(b-)}$ represent the negative rating. So we have

$$R_{t_u}^+ = \sum_{t=t_0}^{t_u} \sum_{m=1}^{M_1} R_m^{(b+)}, \tag{10}$$

$$R_{t_u}^- = \sum_{t=t_0}^{t_u} \sum_{m=1}^{M_2} R_m^{(b-)}, \tag{11}$$

where t_0 is the initial time of current valid period and t_u represents the current time. The interactive quality is calculated as follows:

$$\sigma_{t_u} = \frac{\theta_1 \cdot R_{t_u}^+ - \theta_2 \cdot R_{t_u}^-}{R_{t_u}^+ + R_{t_u}^-}, \tag{12}$$

where θ_1 and θ_2 are sensitivity weight; let $R^+ = R_{t_u}^+, R^- = R_{t_u}^-$, then

$$\theta_1 = \frac{F(R^+)}{F(R^+) + F(R^-)}, \theta_2 = \frac{F(R^-)}{F(R^+) + F(R^-)}. \tag{13}$$

$F(\cdot)$ represents the sensitivity function such as $F(x) = x$, $F(x) = x^2$, and $F(x) = x^3$; the sensitivity of the positive or

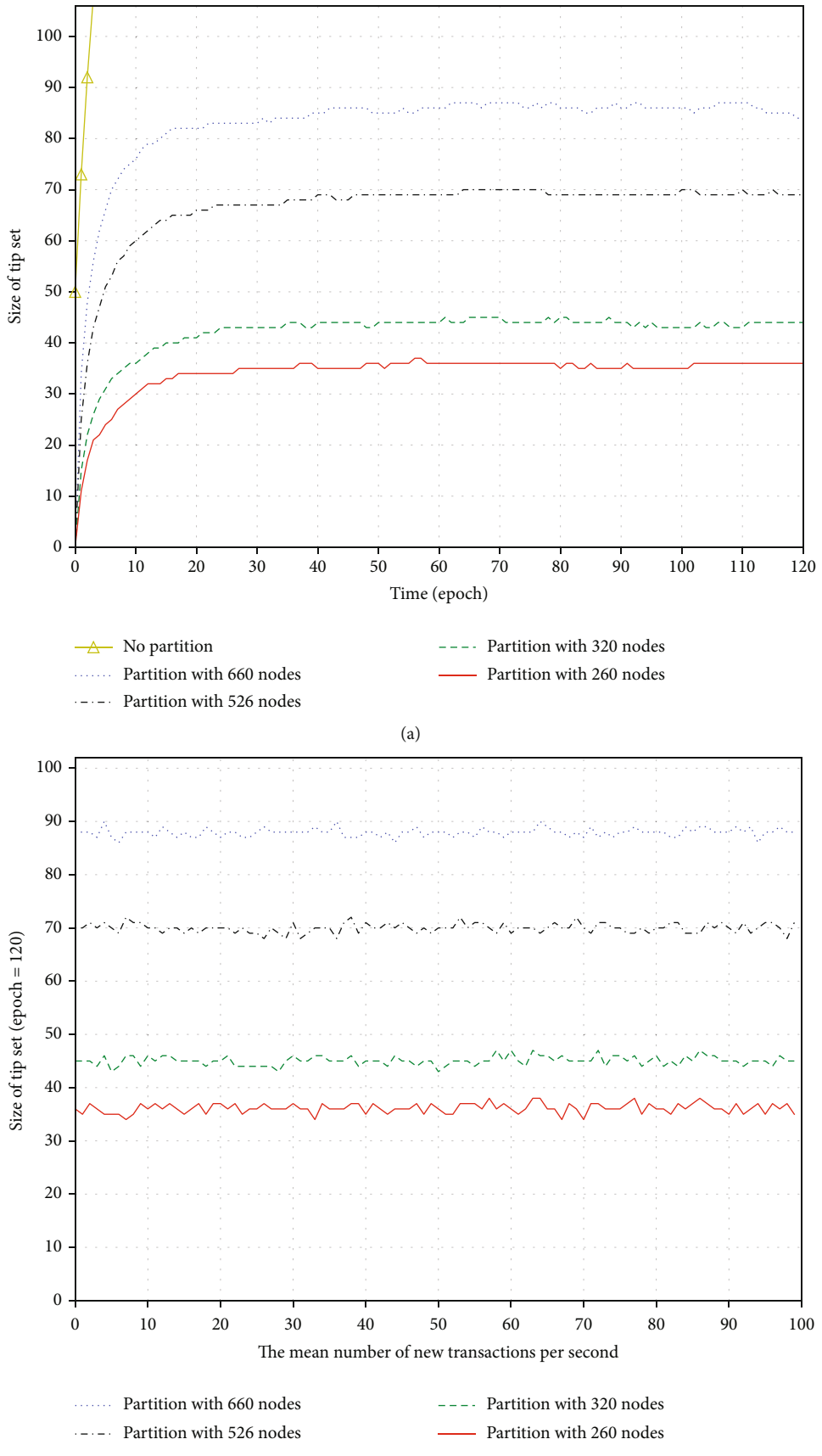


Figure 8: (a) The number of tips in partitions of different sizes. (b) The number of tips vs. TPS in partitions of different sizes, epoch = 120.

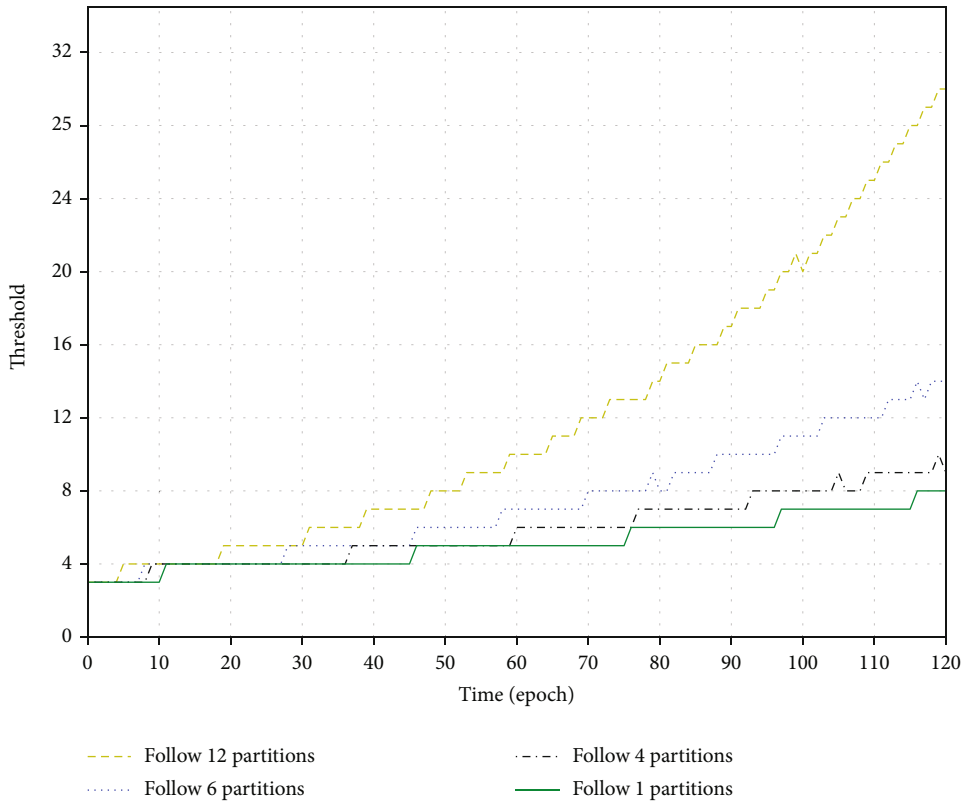


Figure 9: Local consistency threshold vs. epoch, $L = 1$.

negative rating in the metric of interactive quality can be controlled by θ. θ could adjust the weight of contribution rating based on the different requirements. For example, when the shared data may threaten the safety of humans, the weight of negative evaluation should be increased.

5.1.2. Consensus Contribution. When the node selects tips, its behavior determines its contribution to the consensus of DDL. The selections are represented by the edges and are public to all nodes, and they are important on ledger evolution. When receiving a new transaction ψ_i issued by p_b, a node needs to check the CWs of all the transaction in $\{\psi_{i\longrightarrow m}\}$ and calculate a consensus rating $R_{b*}(t)$ of p_b, and the calculation can be expressed by

$$R_b^*(t_u) = R_b^*(t_u - 1) + \sum_{m=1}^{m=M} (\Theta - \mathscr{A}_{i\longrightarrow m}(t_u)), \tag{14}$$

where $\mathscr{A}_{i\longrightarrow m}(t_u)$ presents the CW of the $\psi_{i\longrightarrow m}$ approved by ψ_i at time t_u. If $\Theta - \mathscr{A}_{i\longrightarrow m}(t_u) \le 0$, it denotes that the issuing node selected a confirmed transaction, which represents a bad behavior, of course; otherwise, it denotes a tip is selected or there is other issued unconfirmed transaction recently, which means a good behavior. Therefore, if the node performs positively, then $0 \le R_b^*(t_u) \le 1$; otherwise, $-1 \le R_b^*(t_u) \le 0$. The iterative method is used because the CW of a transaction always increases along with time. Define the metric of consensus contribution as

$$\eta = f(-\beta R_b^*), \tag{15}$$

where β is an attenuation factor; it can be adjusted with the ledger network change even if node p_b does nothing in a time interval. Thus, the metric of consensus contribution of a node's reputation can be calculated by auditing the local transactions at any given time.

5.2. Reputation Update. Since the dimension of the DDL grows with time, it would be nonfeasible to search and audit the related transactions for reputation calculation even in one partition. A possible solution is to maintain additional data structures to store intermediate reputation calculations to save the computing power and time required for transaction search. We called this additional data structure as Node Status List (NSL). Each node should initialize an NSL when first connecting the ledger or a new valid period starting. Table 2 presents the data field contained in each row of the list. The first column is node ID. The second is a set of tuples, and the elements recorded the calculating timestamp and consensus metric. The third is a tuple containing the interactive metric and the update timestamp.

Now, we consider the reputation update method based on the intermediate data structure. The node can calculate the transaction weight and the vehicular reputation by searching the NSL. If a new transaction passes the verification or exceeds the threshold, each receiving node will update the specific field at local. Algorithm 2 describes the transaction consensus with node status data update.

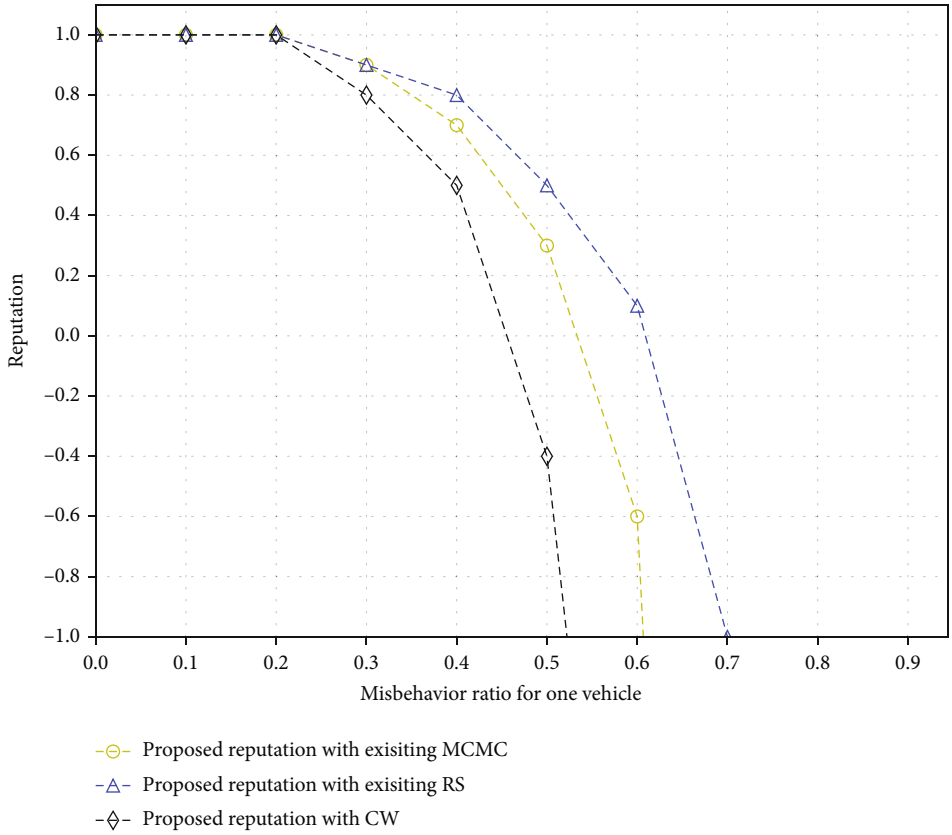


Figure 10: Reputation vs. misbehavior ratio for one vehicle, $f(\cdot) = x^3$.

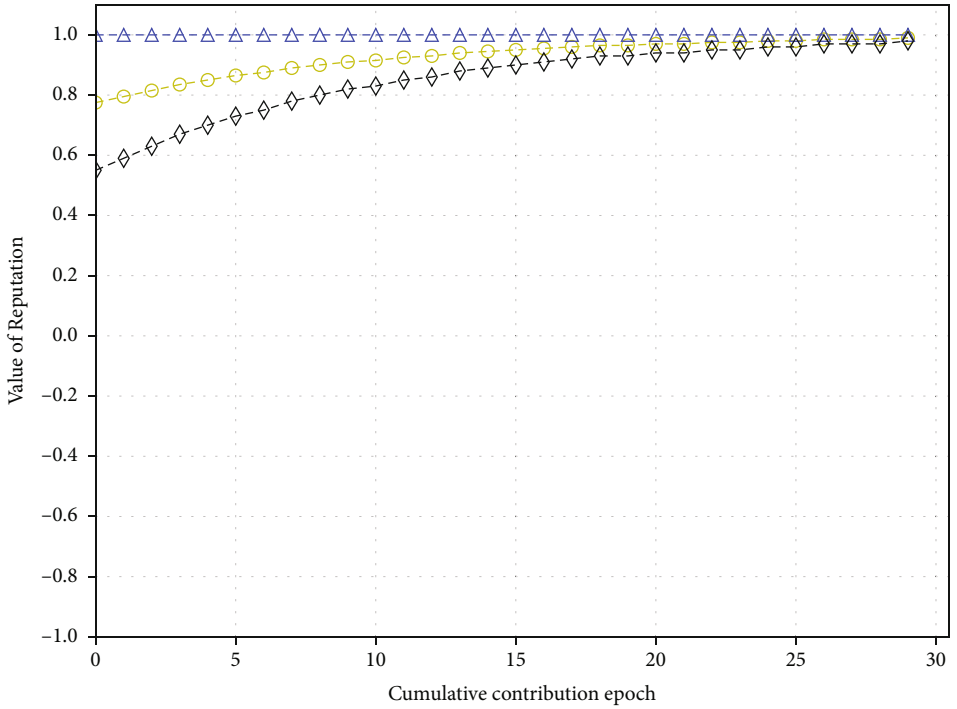
6. Experimental Result and Analysis

We build a trip set based on the New York yellow cab [31] and extract the trajectories at daytime of several days in December 2020. The vehicle number in the evaluation scenario is a uniform distribution between 5800 and 7900 and distributed in 55 partitions. All the partitions are fixed according to predivided partitions in [31], and we filter partitions with less than 20 vehicles and merge these vehicles into nearby partitions. Figure 7 shows the distribution of the partitions and vehicles' heat at different times of one day. We can find that vehicle numbers in different partitions at the same period are very different, and so do the number of vehicles in the same partition at different periods.

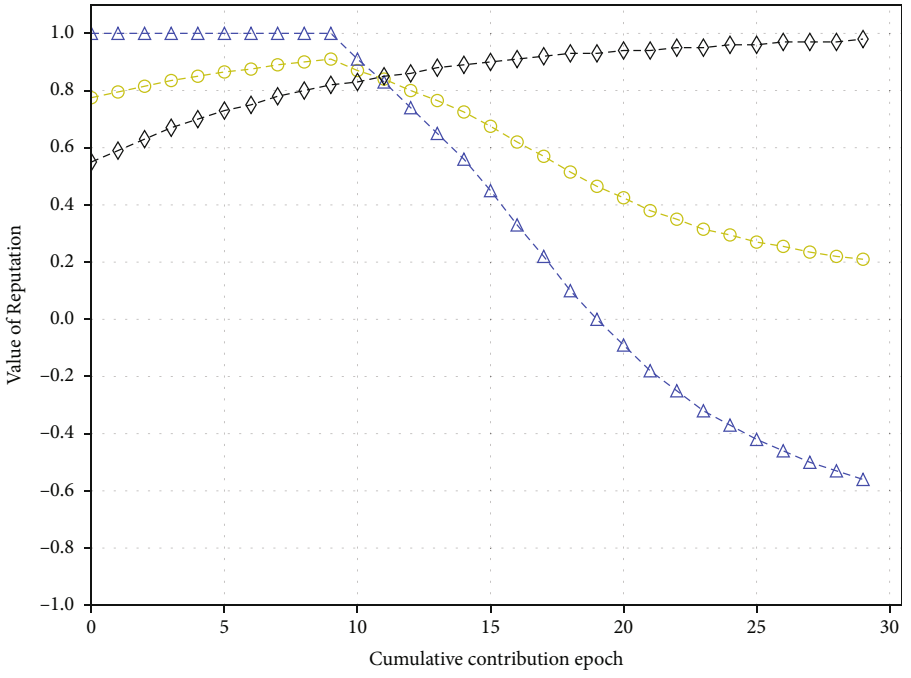
In addition, the entry of vehicles into VANET follows the Poisson process with an average of λ [32], so does the arrival of transactions. Assume that vehicles issuing the transactions follow the power-law distribution [33, 34] and the number of the reference vehicles for the interaction is a uniform distribution between 3 and 10. Considering the simulation is conducted in daytime, we set the system's throughput per second (TPS) in a partition to be large and positively correlated with the number of nodes. Suppose the global arrival rate $\lambda_{all} \leq 1000$, and for each partition, its arrival rate λ_k is also allocated according to the proportion of vehicles owned by it. Thus, for any partitions in our evaluations, suppose $3 < \lambda_k < 86$.

The simulation is implemented using Python 3.8.3 in Windows 10 system with a Lenovo laptop, which has four cores and 32 GB memory. Cryptography is the Python cryptography library (v2.8) [35] and the Hashlib standard library (3.7.7) [36].

6.1. Convergence of the Proposed Ledger. We first investigated the convergence performance of our proposed DAG-based distributed ledger. Figure 8(a) shows that the size of $\mathscr{L}(t)$ increases quickly in the first few epochs and reaches a stable state after around 20 epochs. The main reason for the rapid accumulation of tips in the early stage is the waiting period u. After a tip is released, it will take a while to be "seen" and verified by the nodes. Meanwhile, when a new validity period begins, the node's reputation and each metric are reset to 0.5, so the tips' weights and the parent's selection are very close. The transaction selection can be thought of as random in all transactions, which will cause some tips to be unable to be verified in time. However, as the number of epochs increases, the size of $\mathscr{L}(t)$ becomes stable, which verifies the convergence performance of our proposed DAG ledger in the case of the partitioning method. In Figure 8(b), we observe the change of the size of $\mathscr{L}(t)$ around 120th epoch though adjusting the TPS in partitions. It can be seen that the convergence performance will not be affected because of the definition of the tip and the verifying-before-issuing mechanism of the transaction; that is, when a new transaction is

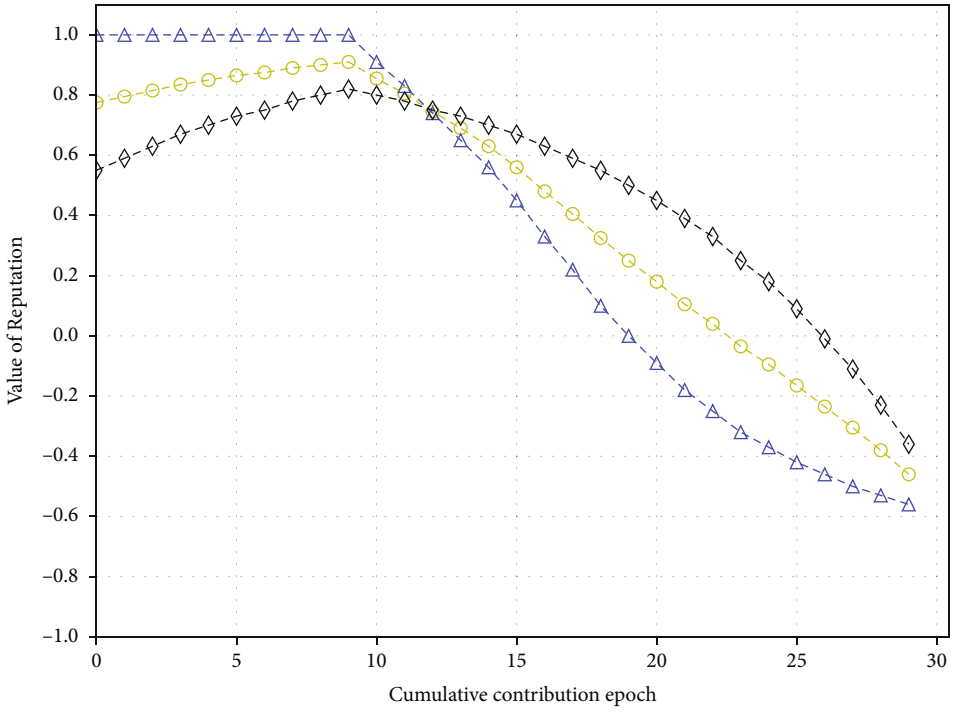


(a) Honest node. The node reputation is reset to 0.5 at the beginning of each validity period. When the node starts to contribute, the evaluated reputation accumulate from 0

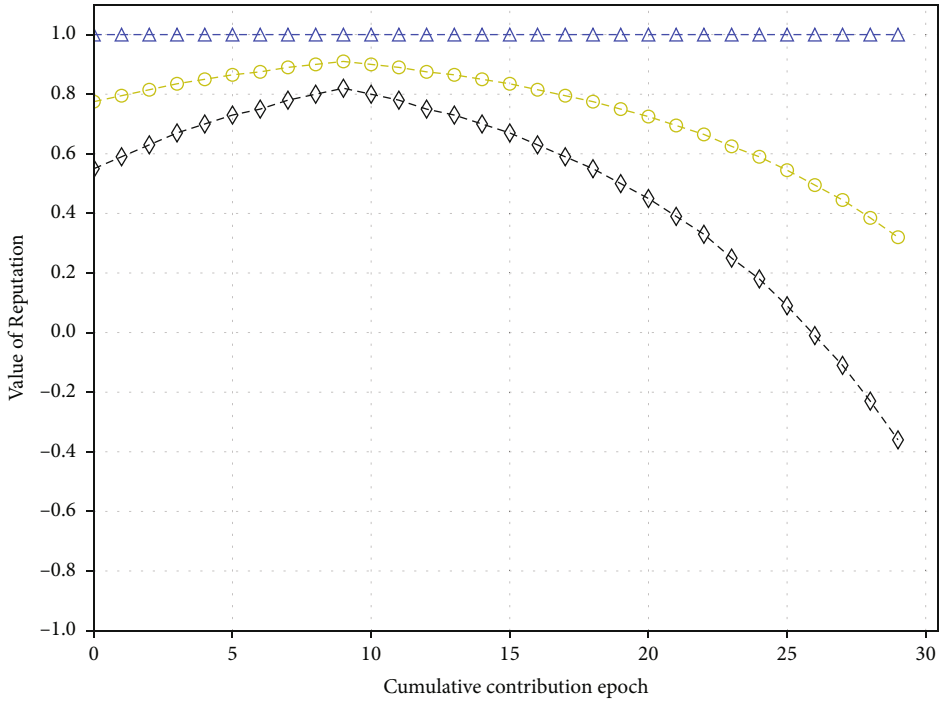


(b) Corrupted node (from epoch = 10), the bad behavior only involves sharing part; the consensus part is normal

Figure 11: Continued.



(c) Corrupted node (from epoch = 10), the bad behavior only involves consensus part; the sharing part is normal



(d) Corrupted node (from epoch = 10), the bad behavior involves both sharing and consensus parts

Figure 11: Reputation accumulation process, $\tau_1 = \tau_2 = 0.5$, $f(\cdot) = x^3$.

added, several tips need to be verified. Therefore, the increase of TPS will also improve tip verification. So, the convergence performance of our proposed DAG ledger in the case of the partitioning method was verified.

6.2. Local Consistency in Multipartitions. Next, we investigated the performance of the local consistency threshold. Figure 9 shows the results of the threshold changing with the number of the followed partitions. We also set the node's participation following the power-law distribution, and only the top 20% active nodes are considered when calculating the threshold starting from the middle of a period (randomly from 40 to 60 epochs in each evaluation). The results show that the threshold increase is slow in the initial few epochs. This is because the threshold is positively correlated with the reputation of the active nodes, and it will be reset to 0.5 in the initial stage of a new validity period. Furthermore, reputation always rises slowly at the beginning of the validity period. Even the most active nodes also need to spend multiple epochs to conduct one good behavior (answering the request from others or computing the PoW for approving transactions) and reach consistency with other related nodes.

Note that in the later epochs, the threshold grows slowly at the scenario of followed fewer partitions, and this is because many nodes have left the followed partitions before their reputation accumulated high enough. However, the reputation grows faster in the scenario of following more partitions, and this is because we can observe the nodes in a more extensive range (involved more partitions), so the nodes have enough time to accumulate a sufficiently high reputation. Moreover, we summarize that some nodes (a taxi will operate for a long time and drive within some fixed partitions) can accumulate much and soon based on the power-law distribution. Therefore, the local consistency threshold is effective; short-travel nodes only need to pay attention to fewer partitions and the recent behavior of the nodes, while long-travel nodes need to pay attention to more partitions and the long-term behavior of the nodes.

6.3. Performance of Reputation Update. Then, we test the resiliency of our reputation representation against misbehavior of vehicles in Figure 10. The misbehavior includes bad collaboration performance (obtained a lower shared rating) or selecting an old and approved transaction to attach. In Figure 10, we can see that all the schemes with different tip selection algorithms can reduce the node's reputation below 0.5 when the node's misbehavior ratio exceeds 50%. Besides, when the reputation is lower than 0.5, the decline is very fast, mainly caused by the bad consensus behavior. Typically, if a node becomes lazy, it would select a fixed transaction set to save its computational power. Meanwhile, the CW of any approved transaction inevitably increases (raised by the indirect approval) with epoch, and the exponential function in Formula (5) also speeds up the decline. This also provides incentives to the node to select the tip.

Last, Figure 11 shows the impact of different metrics on reputation when bad behavior accumulates. We can conclude that when there is bad sharing behavior, the reputation begins to decline rapidly, while the consensus become bad, and the downward trend is slow. This is mainly controlled by the setting of hyperparameter. We can strengthen the weight of consensus metric by adjusting the proportion of τ For example, if the sensitivity function $f(\cdot) = x$, then the sharing metric will decrease linearly. $f(\cdot) = x^3$ can reduce some misbehavior caused by inevitable communication delay; sharing metric will begin to decline rapidly after the misbehavior exceeds the tolerance limit. In addition, when there are few numbers of the activated vehicles, the reputation system can increase the weight of the consensus metric by recommending a large τ_2, so as to attract more vehicles to help verify new transactions, which is also to accumulate its own reputation (consensus metric).

7. Conclusion

This paper proposes a partitioned DDL for maintaining the vehicular reputation to support the trust establishment in VANET. We design the transaction for the vehicular reputation auditing using the details of interactions among vehicles. To encourage the vehicle to maintain the ledger, we design a vehicular reputation evaluation method by aggregating the contribution in vehicular interaction and ledger consensus maintenance. Besides, a reputation update method based on the consistency of transactions in one or several partitions is presented to allow any vehicle to evaluate other's reputations anywhere and anytime. Simulation results demonstrate that our partitioned DDL is practical in real-world scenarios and achieves a better detection rate of bad behavior than the baselines with various tip selection algorithms. Future work is in progress to consider how to partition the VANET better to improve the vehicle's safety during its trip.

Data Availability

The vehicle tip data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] Y. Lu, J. Zhang, Y. Qi et al., "Accelerating at the edge: a storage-elastic blockchain for latency-sensitive vehicular edge computing," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–15, 2021.

[2] S. Smys and H. Wang, "Security enhancement in smart vehicle using blockchain-based architectural framework," *Journal of Artificial Intelligence*, vol. 3, no. 2, pp. 90–100, 2021.

[3] C. Pu, "A novel blockchain-based trust management scheme for vehicular networks," in *2021 wireless telecommunications symposium (WTS)*, pp. 1–6, CA, USA, 2021.

[4] U. Javaid, M. N. Aman, and B. Sikdar, "A scalable protocol for driving trust management in internet of vehicles with blockchain," *IEEE Internet of Things Journal*, vol. 7, no. 12, pp. 11815–11829, 2020.

[5] J. Cui, F. Ouyang, Z. Ying, L. Wei, and H. Zhong, "Secure and efficient data sharing among vehicles based on consortium

Blockchain," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–11, 2021.

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

[7] S. Nakamoto and A. Bitcoin, *A peer-to-peer electronic cash system*, 2008, https://bitcoin org./bitcoin.pdf.

[8] S. King and S. Nadal, *PPCoin: peer-to-peer crypto-currency with proof-of-stake*, self-published paper, 2012.

[9] J. Kang, Z. Xiong, D. Niyato, D. Ye, D. I. Kim, and J. Zhao, "Toward secure blockchain-enabled internet of vehicles: optimizing consensus management using reputation and contract theory," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 3, pp. 2906–2920, 2019.

[10] S. Popov, *The tangle*, White paper, 2018.

[11] Coordicide Team, IOTA Foundation, *The coordicide*, 2019, https://files.iota.org/papers/Coordicide WP.pdf.

[12] H. Liu, Y. Zhang, and T. Yang, "Blockchain-enabled security in electric vehicles cloud and edge computing," *IEEE Network*, vol. 32, no. 3, pp. 78–83, 2018.

[13] V. Hassija, V. Chamola, V. Gupta, and G. S. S. Chalapathi, "A framework for secure vehicular network using advanced blockchain," in *2020 international wireless communications and Mobile computing (IWCMC)*, pp. 1260–1265, Limassol, Cyprus, 2020.

[14] P. C. Bartolomeu, E. Vieira, and J. Ferreira, "IOTA feasibility and perspectives for enabling vehicular applications," in *2018 IEEE Globecom Workshops (GC Wkshps)*, pp. 1–7, Abu Dhabi, United Arab Emirates, 2018.

[15] Y. Lu, Y. Qi, and S. Qi, "Say no to price discrimination: decentralized and automated incentives for price auditing in ride-hailing services," *IEEE Transactions on Mobile Computing*, p. 1, 2020.

[16] Y. Lu, J. Zhang, Y. Qi et al., "Safety warning! Decentralised and automated incentives for disqualified drivers auditing in ride-hailing services," *IEEE Transactions on Mobile Computing*, vol. 21, p. 1, 2021.

[17] Z. Lu, Q. Wang, G. Qu et al., "BARS: a blockchain-based anonymous reputation system for trust management in VANETs," in *2018 17th IEEE international conference on trust, security and privacy in computing and communications/12th IEEE international conference on big data science and engineering (TrustCom/BigDataSE)*, pp. 98–103, New York, NY, USA, 2018.

[18] N. Szabo, *Smart contracts*, 2014, http://szabo.best.vwh.net/smart.contracts.html.

[19] H. Liu, D. Han, and D. Li, "Fabric-IoT: a blockchain-based access control system in IoT," *IEEE Access*, vol. 8, pp. 18207–18218, 2020.

[20] D. Wang, L. Zhang, C. Huang, and X. Shen, "A privacy-preserving trust management system based on blockchain for vehicular networks," in *2021 IEEE wireless communications and networking conference (WCNC)*, pp. 1–6, Nanjing, China, 2021.

[21] L. Liu, M. Zhao, M. Yu, M. A. Jan, D. Lan, and A. Taherkordi, "Mobility-aware multi-hop task offloading for autonomous driving in vehicular edge computing and networks," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–14, 2022.

[22] S. Kudva, S. Badsha, S. Sengupta, H. la, I. Khalil, and M. Atiquzzaman, "A scalable blockchain based trust management in VANET routing protocol," *Journal of Parallel and Distributed Computing*, vol. 152, pp. 144–156, 2021.

[23] A. Khalid, M. S. Iftikhar, A. Almogren, R. Khalid, M. K. Afzal, and N. Javaid, "A blockchain based incentive provisioning scheme for traffic event validation and information storage in VANETs," *Information Processing & Management*, vol. 58, no. 2, p. 102464, 2021.

[24] J. Kang, R. Yu, X. Huang et al., "Blockchain for secure and efficient data sharing in vehicular edge computing and networks," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4660–4670, 2019.

[25] F. Li, Z. Guo, C. Zhang, W. Li, and Y. Wang, "ATM: an active-detection trust mechanism for VANETs based on blockchain," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 5, pp. 4011–4021, 2021.

[26] E. Diallo, O. Dib, and A. K. Al, "An improved PBFT-based consensus for securing traffic messages in VANETs," in *2021 12th international conference on information and communication systems (ICICS)*, pp. 126–133, Valencia, Spain, 2021.

[27] X. Zhang, R. Li, W. Hou, and H. Zhao, "V-Lattice: a lightweight blockchain architecture based on DAG-lattice structure for vehicular ad hoc networks," *Security and Communication Networks*, vol. 2021, 17 pages, 2021.

[28] C. LeMahieu, *Nano: a feeless distributed cryptocurrency network*, 2018, https://nano. org/en/whitepaper.

[29] A. Churyumov, *Byteball: a decentralized system for storage and transfer of value*, white paper ed edition, , 2016https://byteball.org/Byteball.pdf.

[30] Q. Li, A. Malip, and K. M. Martin, "A reputation-based announcement scheme for VANETs," *IEEE Transactions on Vehicular Technology*, vol. 61, no. 9, pp. 4095–4108, 2012.

[31] *New York City Taxi and Limousine Commission*https://www1.nyc.gov/site/tlc/about/data.page.

[32] Q. Cui, N. Wang, and M. Haenggi, "Vehicle distributions in large and small cities: spatial models and applications," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 11, pp. 10176–10189, 2018.

[33] Y. Li, D. Jin, Z. Wang, L. Zeng, and S. Chen, "Exponential and power law distribution of contact duration in urban vehicular ad hoc networks," *IEEE Signal Processing Letters*, vol. 20, no. 1, pp. 110–113, 2012.

[34] T. Chen, Z. Li, Y. Zhu et al., "Understanding ethereum via graph analysis," *ACM Transactions on Internet Technology (TOIT)*, vol. 20, no. 2, pp. 1–32, 2020.

[35] *Cryptography library (v2.8)*https://cryptography.io/en/latest/.

[36] *Hashlib standard library (3.7.7)*https://docs.python.org/3/library/hashlib.html.

Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 9358531, 13 pages
https://doi.org/10.1155/2022/9358531

Research Article

Protecting Check-In Data Privacy in Blockchain Transactions with Preserving High Trajectory Pattern Utility

Xiufeng Xia, Tingting Hou, Xiangyu Liu, Chuanyu Zong, and Shengsheng Mu

School of Computer Science, Shenyang Aerospace University, Shenyang 110136, China

Correspondence should be addressed to Tingting Hou; 192106074080@email.sau.edu.cn

Received 16 December 2021; Revised 25 January 2022; Accepted 9 February 2022; Published 15 March 2022

Academic Editor: Qingqi Pei

Because the blockchain is secure and untamperable, it has been widely used in many industries, such as the financial industry, digital tokens, and e-commerce logistics. The remarkable security feature of the blockchain is that the blockchain verifies the transaction initiated on each block through the node, and its process is broadcast throughout the whole network to let everyone know. On the one hand, this ensures the security of every transaction, but on the other hand, it is easy to cause privacy disclosure problems for transaction users. Therefore, under the premise of ensuring the security of the blockchain, it has become a hot issue to protect the sensitive information of transaction users. A check-in privacy protection (CPP) algorithm based on check-in location generalization is proposed in this paper, which can be applied to blockchain transactions to solve the privacy leakage problem of transaction users' sensitive information. CPP algorithm not only protects the privacy of check-in data but also keeps the high utility of trajectory pattern data. Firstly, location types are recommended in the sensitive check-in location generalization based on the user's trajectory pattern by using Markov chain technology. Secondly, to make sure that the generalized locations can be scattered as much as possible to prevent the attacker from deducing back, a heuristic rule is designed to select the generalized location based on the recommended location types, and at the same time, the similarity between the anonymous trajectory and the original trajectory is maintained. In addition, a generalized location search strategy is designed to improve the efficiency of the algorithm. Based on the real spatial-temporal check-in data, the results of the experiment indicate that our algorithm can effectively protect the privacy of sensitive check-in while ensuring the high utility of trajectory pattern data.

1. Introduction

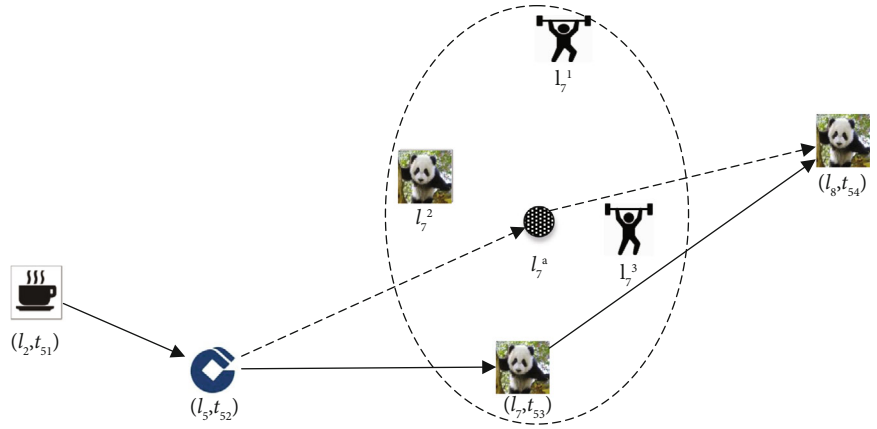
In recent years, the blockchain [1] has been broadly used in the financial industry, digital tokens, e-commerce logistics, and many other industries due to its characteristics of security and untampering. The significant security feature of the blockchain is that the blockchain authenticates each transaction initiated on each block through the node, and its process is broadcast throughout the network for everyone to know. This not only ensures the security of the transaction but also brings privacy harm to the transaction users. Hence, under the premise of ensuring the security of the blockchain [2, 3], it is already an issue worthy of attention to protect the sensitive information of transaction users. With the constant development of mobile networks [4, 5], vehicular networks [6–9], wireless communications network [10], and GPS-enabled devices, a mass of check-in data [11] of mobile users has been collected and utilized.

Check-in data contains the characteristics of human behavior, which plays a key role in major social science issues such as disease transmission, epidemic prevention and control, poverty eradication, urban planning, and other important life applications such as route recommendation and bus travel. Government and many research institutions hope to create more value through data mining. The trajectory contains many sensitive check-in data. Users' private information (home address, religious belief, interests, health, and other private information) will be obtained and used by malicious attackers, assuming these sensitive check-in data is leaked. Therefore, protecting sensitive check-in data in trajectory has become a challenging problem.

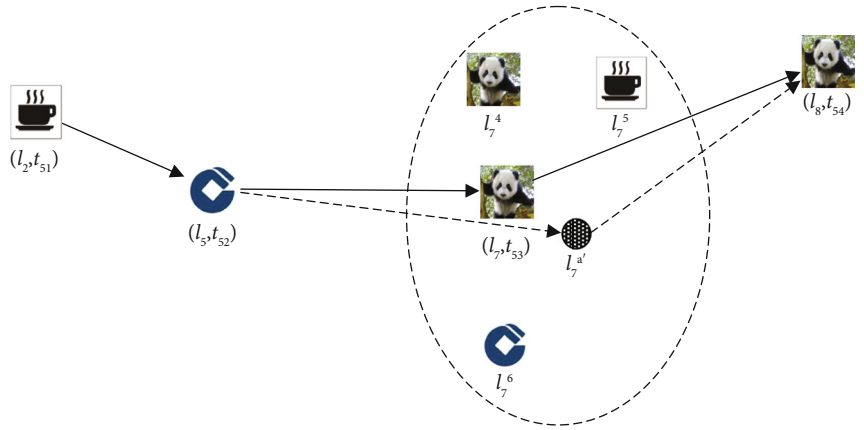
ID	Trajectory
tr_1	$<(l_6,t_{11}),(l_1,t_{12}),(l_4,t_{13}),(l_9,t_{14}),(l_3,t_{15})>$
tr_2	$<(l_4,t_{21}),(l_8,t_{22}),(l_2,t_{23})>$
tr_3	$<(l_9,t_{31}),(l_5,t_{32}),(l_4,t_{33}),(l_7,t_{34}),(l_1,t_{35})>$
tr_4	$<(l_2,t_{41}),(l_4,t_{42}),(l_6,t_{43})>$
tr_5	$<(l_2,t_{51}),(l_5,t_{52}),(l_7,t_{53}),(l_8,t_{54})>$
tr_6	$<(l_4,t_{61}),(l_2,t_{62}),(l_7,t_{63})>$

Type of location	Sign	Locations
Zoo (T_1)		l_7,l_8,l_9
Fitness room (T_2)		l_1,l_3
Coffee shop (T_3)		l_2,l_6
Bank (T_4)		l_4,l_5
Anonymous central location		$l_7^a,l_7^{a'}$

(a) A set of user history trajectory



(b) The random method is used to protect the sensitive check-in in the trajectory



(c) This experiment is used to protect the sensitive check-in in the trajectory

Figure 1: User trajectory set and anonymous trajectory.

Check-in data means the user visits a certain place at a certain time. Sensitive check-in refers to user hopes to keep check-in data from being leaked. The user u's historical trajectory set $T_r = \{\mathrm{tr}_1, \mathrm{tr}_2, \mathrm{tr}_3, \mathrm{tr}_4, \mathrm{tr}_5, \mathrm{tr}_6\}$ is shown in Figure 1(a), and four check-in data in chronological order are included in the trajectory $\mathrm{tr}_5 = <(l_2, t_{51}), (l_5, t_{52}), (l_7, t_{53}), (l_8, t_{54})>$. The check-in data (l_7, t_{53}) indicates that user u visits location l_7 at time t_{53}. The location type of l_7 is the zoo, and t_{53} belongs to user u's office time. User u does not want to disclose the check-in data; therefore, (l_7, t_{53}) is set to sensitive check-in of user u. Currently, there is no privacy protection technology for sensitive check-in, and location privacy protection [12–15] is nearest to the problem.

Location generalization [16] is a popular location privacy protection method, and it has the characteristics of retaining the user's complete location information, law computation, and simple mechanism. However, these current location generalization methods do not consider the user trajectory pattern factor, which may reduce the privacy protection degree of sensitive check-in or even directly reveal the real sensitive check-in of user.

For example, the trajectory in Figure 1(b) is an anonymous trajectory obtained by using location generalization technology under 4-anonymity privacy requirement (4-anonymity means that the probability of identifying the sensitive check-in location based on this anonymous trajectory is no more than 1/4). Generalized locations l_7^1, l_7^2, and l_7^3 are obtained by a random method in literature [17], and these and the real sensitive check-in location l_7 form an anonymous location set to participate in the trajectory release; thus, the attacker cannot guess the real check-in location of user u at time t_{53}. This anonymous trajectory has two problems: (1) it does not conform to the user trajectory pattern. Location type of l_5 is bank. Based on user history trajectory, it can be seen that the next possible location type is the zoo, the coffee shop, and the bank from the current location type with the probability of visit being 4/7, 2/7, and 1/7, respectively. However, the location type of l_7^1 and l_7^3 is the fitness room. Obviously, the attacker can easily deduce that (l_7^1, t_{53}) and (l_7^3, t_{53}) are false check-in based on user history trajectory. (2) The similarity between the anonymous trajectory and the original trajectory decreases. l_7^a is the anonymous central location generated after random generalization, and $l_7^{a'}$ is the anonymous central location generated after another anonymization. As shown in Figure 1(b), the shape of the anonymous trajectory $<(l_2, t_{51}), (l_5, t_{52}), (l_7^a, t_{53}), (l_8, t_{54}) >$ differs greatly from that of the original trajectory $<(l_2, t_{51}), (l_5, t_{52}), (l_7, t_{53}), (l_8, t_{54}) >$. In Figure 1(c), the shape of the anonymous trajectory $<(l_2, t_{51}), (l_5, t_{52}), (l_7^{a'}, t_{53}), (l_8, t_{54}) >$ is closer to that of the original trajectory $<(l_2, t_{51}), (l_5, t_{52}), (l_7, t_{53}), (l_8, t_{54}) >$. Due to the above two problems, the probability of identifying sensitive check-in will be greater than 1/4 and lead to privacy disclosure of sensitive check-in. To solve the problem, this paper proposes a check-in privacy protection algorithm based on check-in location generalization to protect the privacy of check-in data and keep the high utility of trajectory pattern data.

The main contributions are as follows:

(1) We propose a check-in privacy protection (CPP) algorithm based on check-in location generalization

(2) We recommend generalized location types by using Markov chain technology, and design a heuristic rule to select generalized locations

(3) We optimize the generalized location search strategy to improve the efficiency of the algorithm

(4) Extensive empirical studies show that our algorithm performs efficiently to protect check-in data while preserving the high utility of trajectory pattern data

The rest of the paper is organized as follows. Section 2 analyzes related work. Section 3 presents some important concepts and problem definition. Section 4 elaborates our scheme in detail. Section 5 evaluates the performance of CPP. We conclude this paper in Section 6.

2. Related Work

Sweeney [12] first proposed the concept of the k-anonymity model, and it was first applied in the relational database. Subsequently, Gruteser and Grunwald and Gruteser and Liu [18, 19] applied the k-anonymity model to location privacy protection. The core idea of it is that the anonymous server selects $k-1$ generalized locations to form an anonymous set with user real location, and the k locations cannot be distinguished from each other. Gedik and Ling [20] proposed the Clique Cloak algorithm, which constructed the anonymous region based on the graph model combined with time and space factors, and transformed the problem of anonymous set into the problem of finding $k-1$ neighbors in the graph model. Wang et al. [21] proposed a generalized location generation scheme based on semantic information and query probability, which can generate $k-1$ generalized locations related to user location semantic information. Niu et al. [22] proposed an enhanced DLS algorithm, which can select $2k$ generalized locations with high query probability similarity to the real location by calculating the location entropy and then select $k-1$ generalized locations from them by calculating the product of location distance. Lu et al. [23] proposed two generalized location generation algorithms CirDummy and GridDummy to realize location k-anonymity considering the shape of user privacy region.

Dwork [13] first proposed the differential privacy protection method, which protects privacy by adding noise to distort data. The differential privacy protection technology with mathematical theory and strict mathematical definition has two characteristics: first, it is not affected by attackers with background knowledge, and second, it is not affected by changing the specific data. Xiong et al. [24] proposed a spatial crowdsourcing algorithm based on a reward mechanism, which protects location privacy by adding Laplace noise to location data. Xu et al. [25] proposed a hybrid location privacy protection method, which divided locations into discrete locations and nondiscrete locations. For discrete locations, differential privacy technology was directly used for noise processing; while for nondiscrete locations, a k-means clustering algorithm based on differential privacy technology was used for generalization processing. However, excessive noise will lead to poor data availability and serious errors. Thus, Ping et al. [26] proposed PriLocation, a differential privacy protection method for noise reduction, to solve effectively this problem caused by excessive noise.

The basic idea of the location privacy protection method based on encryption technology is to encrypt the user's query information. Even if the attacker obtains the query information, he cannot know the real privacy information behind the query information. Zhang and Ni [14, 15] proposed a neighbor query method PRN-KNN, which uses a spatial encryption algorithm to enable users to quickly query k-neighbor candidate sets and introduces pseudo-random number secret rules to effectively reduce algorithm processing time. Papadopoulos et al. [27] used security hardware to assist PIR protocol and protected user location privacy through KNN query. Encryption-based location privacy protection technology can better ensure data availability and service accuracy, but the disadvantage is a large amount of calculation.

3. Preliminaries and Problem Definition

The check-in data set of user u is represented as $C_u = \{c_i \mid i \in [1, m]\}$. The check-in data $c_i = (l_i, t_i)$ indicates that user u visits location l_i at time t_i, where t_i is the check-in time, and l_i is the specific location on the map, such as Northeastern University, Wanda Plaza, and Beiling Park, and (x, y) is the latitude and longitude of a specific location, respectively. T_i represents the location type of a specific location, such as universities, shopping centers, and parks.

Definition 1 Sensitive check-in. Given trajectory $\mathrm{tr} = <(l_1, t_1), (l_s, t_s), \cdots, (l_n, t_n)>$, if the user does not want to check in, (l_s, t_s) was exposed, so (l_s, t_s) is called sensitive check-in. As shown in Figure 1(b), (l_7, t_{53}) is a sensitive check-in in trajectory tr_5.

Definition 2 Trajectory pattern matrix M. Given an $m * m$ matrix, $T_1, \cdots, T_m$ represents the location type, and $M(T_i, T_j)$ represents the probability that the user travels from location type T_i to location type T_j.

As shown in Table 1(a), the location type of the zoo, the fitness room, the coffee shop, and the bank are, respectively, denoted T_1, T_2, T_3, and T_4, respectively. The user trajectory pattern matrix M is obtained according to the transfer situation of location type in user's historical trajectory set T_r, where $M(T_1, T_2)$ represents the transfer probability that the user travels from location type T_1 to the next location type T_2. In Figure 1(a), location type T_1 includes l_7, l_8, and l_9. The next location of l_7 is l_1 (T_2) and l_8 (T_1). The next location of l_8 is l_2 (T_3). The next location of l_9 is l_3 (T_2) and l_5 (T_4). Therefore, the value of $M(T_1, T_2)$ is 2/5.

Definition 3 Check-in location generalization. Given a check-in data (l_s, t_s), generalization operation refers to convert location l_s of check-in (l_s, t_s) to a location set $L' = \{l_s, l_1', l_2', \cdots, l_i'\}$, there are $1 + i$ locations in L', and the probability that any location in the set appears between moment t_{s-1} and moment t_{s+1} is equal.

Definition 4 Anonymous trajectory. The trajectory obtained after replacing the sensitive check-in location l_s in the original trajectory with the anonymous center location l_s^a after anonymization.

As shown in Figure 1(b), the anonymous trajectory is represented as $<(l_2, t_{51}), (l_5, t_{52}), (l_7^a, t_{53}), (l_8, t_{54})>$.

Definition 5 Trajectory pattern similarity. Given the original trajectory pattern matrix M and the anonymous trajectory pattern matrix M' (the order of the matrix is m), the trajectory pattern similarity is shown in Formula (1):

$$\mathrm{sim}\left(M, M'\right) = \frac{\sum M(i,j) * M'(i,j)}{\sqrt{\sum M(i,j)^2}\sqrt{\sum M'(i,j)^2}} \quad i, j \in (1, m). \tag{1}$$

Table 1: Trajectory pattern matrix.

(a) User trajectory pattern matrix

Matrix M	T_1	T_2	T_3	T_4
T_1	0.2	0.4	0.2	0.2
T_2	0	0	0	1
T_3	0.25	0.25	0	0.5
T_4	4/7	0	2/7	1/7

(b) Anonymous trajectory pattern matrix

Matrix M'	T_1	T_2	T_3	T_4
T_1	3/19	8/19	4/19	4/19
T_2	0	0	0	1
T_3	0.25	0.25	0	0.5
T_4	16/29	0	8/29	5/29

As shown in Figure 1(b), the anonymous trajectory pattern matrix M' is obtained by anonymizing the original trajectory pattern matrix M, and the value of trajectory pattern similarity $\mathrm{sim}(M, M')$ is 99.93%.

Definition 6 Check-in k-anonymity. Given sensitive check-in (l_s, t_s), the generalized location set $c' = \{l_s^1, \cdots, l_s^m\}$ is get through the check-in location generalization operation, where size $(c') >= k$, so that the leakage rate of check-in location is not greater than $1/k$, namely, check-in k -anonymity.

Definition 7 Location exposure rate LE. The generalized location is expressed as l', the location anonymous set is composed of the real check-in location l and $k-1$ generalized locations, namely, $\mathrm{LAS} = \{l, l^1, l^2, \cdots, l^{k-1}\}$. Given the user location anonymous set LAS, the attacker uses background knowledge to identify LAS and infers the probability of the user real check-in location as shown in Formula (2):

$$\mathrm{LE} = \frac{1}{|\mathrm{LAS}| - \left|\mathrm{LAS}'\right|}. \tag{2}$$

$|\mathrm{LAS}|$ represents the total number of locations in an anonymous set, and $|\mathrm{LAS}'|$ indicates that the attacker can identify the number of generalized locations.

Definition 8 Distance between trajectories. Given original trajectory, sensitive check-in location l_s, anonymous trajectory, and anonymous center location l_s^a, the distance between trajectories is defined as the Euclidean distance between two locations as seen in Formula (3):

$$\mathrm{tr\_dist}(l_s, l_s^a) = \sqrt{(l_s.x - l_s^a.x)^2 + (l_s.y - l_s^a.y)^2}. \tag{3}$$

Input: sensitive check-in location l_s, privacy protection threshold k;
Output: an anonymous set of locations containing k locations.
1. $LAS \longleftarrow \varnothing$;
2. T= MC-LTR (M, $r(M)$, sub_T,s);
3. L= GLS ($D - index$, R, tr);
4. S = LATP (k, T, M, T_s);
5. $Cand_l$= DLS (S, k, l_s);
6. LAS = $Cand_l \bigcup l_s$;
7. if(LE<1/k)then
8. Return LAS.
9. else
10. Return $\varnothing$.

Algorithm 1: Check-in privacy protection algorithm based on generalization of check-in location.

Problem definition. Given check-in data set C_u, sensitive check-in set S_u of user u, real trajectory tr, and privacy protection threshold k, the location anonymous set LAS is obtained by generalizing the sensitive check-ins in sensitive check-in set based on trajectory pattern. The generalized check-ins in LAS not only meet check-in k-anonymity but also ensure the maximum similarity of trajectory pattern.

4. Check-In Privacy Protection Algorithm Based on Generalization of Check-In Location

In this section, the check-in privacy protection algorithm based on check-in location generalization (Algorithm 1) is proposed. The main idea is to select the generalized location based on the original trajectory pattern matrix in the process of check-in location generalization so that the generalization operation can change the similarity between the original trajectory pattern matrix and the anonymous trajectory pattern matrix as little as possible. Thus, high data availability of anonymous trajectory in trajectory patterns is guaranteed. The algorithm framework of this paper is shown in Figure 2. The algorithm framework can show that users' sensitive check-ins are protected by the four algorithms (Algorithms 1–4) proposed in this paper, and this method can be used to protect user identity information in blockchain transactions.

First, the Markov chain-location type recommendation (MC-LTR) algorithm is used to recommend the set of location types for sensitive check-ins (line 2). Generalizing location search (GLS) algorithm is used to search the specific location in the generalization area (line 3). The location assignment based on trajectory pattern (LATP) algorithm is adopted to allocate the number of generalized locations corresponding to the recommended location type, and the aim is to ensure that the change of the anonymized trajectory pattern matrix is minimal (line 4). The dummy location selection (DLS) algorithm is used to obtain the candidate array of generalized locations (line 5). As shown in Formula (4), score is a heuristic function, whose value measures the influence of the distance product between the generalized locations and the sensitive check-in location and the distance between trajectories before and after anonymity. The higher the value, the more scattered between the generalized locations and the sensitive check-in location, and the closer the distance between the anonymous trajectory and real trajectory is. Finally, the CPP algorithm returns an anonymous location set containing k locations (line 6).

$$\text{Score} = \frac{\prod \text{dist}(l_i, l_j)}{\text{dist}(l_s, l_s^{\,a})} \left(l_i \neq l_j\right). \tag{4}$$

For example, we protect sensitive check-in (l_7, t_{53}) in trajectory tr_5.The random choice of location type is likely to expose the user's sensitive check-in, so the MC-LTR algorithm is used to ensure that the generalized location type conforms to the user's historical trajectory pattern. The location type of sensitive check-in location l_7 is T_1, and the location type of next moment predicted based on Markov chain includes T_1, T_2, T_3, and T_4, and the recommendation probability is 4/35, 0, 1/14, and 4/49, respectively. Therefore, the recommended set of location types for sensitive check-in (l_7, t_{53}) is $T = \{T_1, T_4, T_3\}$. Searching the specific location corresponding to the recommended location type mainly considers two factors: historical average speed and time accessibility. In the query area, GLS algorithm will be used to put the searched specific locations corresponding to each location type in the set into location queue L, namely, $L = [l_1^{\,1}, l_1^{\,2}, l_4^{\,1}, l_4^{\,2}, l_4^{\,3}, l_3^{\,2}]$, wherein the location type T_1 contains two specific locations $l_1^{\,1}$ and $l_1^{\,2}$, and the location type T_4 contains three specific locations $l_4^{\,1}$, $l_4^{\,2}$ and $l_4^{\,3}$, and the location type T_3 contains one specific location $l_3^{\,2}$. Due to need to achieve the 4-anonymity protection, three generalized locations are selected from location queue L to ensure that the anonymous set can achieve optimal protection. By using the LATP algorithm, one location type meeting the requirement of anonymity is selected at a time, and the number array S of generalized location types is obtained. Among them, $S[T_1] = 2$, $S[T_4] = 1$, and $S[T_3] = 0$. The dispersion between locations and the change situation of the original trajectory's shape and the anonymous trajectory's shape are considered. The purpose is to prevent the location exposure and ensure trajectory similarity. Finally, the DLS algorithm is used to select 3 candidates from the location

Input: sensitive check-in location l_s, privacy protection threshold k;
Output: an anonymous set of locations containing k locations.

1. $LAS \leftarrow \emptyset$;
2. T= MC-LTR $(M, r(M), sub\_T, s)$;
3. L= GLS $(D - index, R, tr)$;
4. S = LATP (k, T, M, T_s);
5. $Cand_l$= DLS (S, k, l_s);
6. LAS = $Cand_l \cup l_s$;
7. **if** $(LE < 1/k)$ **then**
8. **Return** LAS.
9. **else**
10. **Return** $\emptyset$.

Figure 2: The algorithm framework of CPP.

Input: trajectory pattern matrix M, reverse trajectory pattern matrix $R(M)$, sensitive sub-trajectory type $sub\_T = \{T_{before}, T_s, T_{behind}\}$, recommended location type quantity threshold s;
Output: set T of location types.
1. Initialize $T[i]$, $i \in [1, |M|]$;
2. if $(sub\_T[0] == \emptyset)$ then
3. $T[i] = R(M)[sub\_T[2]][i], i \in [1, |M|]$;
4. else if $(sub\_T[2] == \emptyset)$ then
5. $T[i] = M[sub\_T[0]][i], i \in [1, |M|]$;
6. else then
7. $T[i] = M[sub\_T[0]][i] \times R(M)[i][sub\_T[2]]$; $i \in [1, |M|]$;
8. sort$(T[i])$;
9. Select the first s location types with higher probability values and put them into T;
10. Return T.

Algorithm 2: The Markov-chain location type recommendation algorithm.

Input: distance index *D-index*, query distance R, any location l_i;
Output: location queue L.
1. $L = \emptyset$;
2. pos =1, end =lens (*D-index* $[l_i]$);
3. While (pos<end) do
4. mid=(post+end)/2;
5. if *D-index*[mid]<R then
6. pos=mid+1;
7. else if *D-index*[mid]>R then
8. end=mid-1;
9. else
10. pos=end=mid;
11. L=*D-index* $[l_i]$, $i \in (1, pos)$;
12. Return L.

Algorithm 3: The generalized location search algorithm.

queue L and put them into the location anonymous set LAS, namely, LAS = $\{l_1^1, l_1^2, l_4^2\}$.

4.1. Recommendations for Generalizing Location Types. This section mainly introduces recommendations of the location type for sensitive check-in based on the MC-LTR algorithm (Algorithm 2). For example, by using this algorithm, the location type recommendation is made during generalized sensitive check-in (l_7, t_{53}). Check-in data is an integral part of user trajectory, and user trajectory patterns can reflect user behavior characteristics, so the selection of location type should be in accordance with the user trajectory movement pattern. Because the sensitive check-in location l_7 is located in the middle of the trajectory, two predictions are needed to realize the location type recommendation. According to the previous moment location type T_4 of the sensitive

Input: privacy protection threshold k, generalized location type set T, trajectory pattern matrix M, sensitive location type T_s, location queue L;
Output: generalized location type quantity array S.
1. Initialize $S[T_i]=0$, $i \in [1,|T|]$;
2. $S[T_s]=\min(k\text{-}1, L(T_s))$;
3. While $\sum S[T_i]<k\text{-}1$ do
4. for each location type $T_i \epsilon T$ $(T_i \neq T_s)$do
5. if $S[T_i]<=L(T_i)$ then
6. $S[T_i]$++;
7. update M based on $S[T_i]$ to change M to M^{T_i};
8. Calculate $sim(M, M^{T_i})$;
9. $S[T_i]$−;
10. $T_{max}= T_i$ which maximizes $sim(M, M^{T_i})$;
11. M=$M^{T_{max}}$;
12. Return S.

Algorithm 4: The location assignment based on trajectory pattern algorithm.

check-in, it is predicted that the generalized location types of the sensitive check-in are T_1, T_2, T_3, and T_4, and the probabilities are 4/7, 0, 2/7, and 1/7, respectively. The transfer probability of each generalized location type to the next moment location type for sensitive location is 1/5, 0, 1/4, and 4/7, respectively. Therefore, the recommended set of generalized location types for sensitive check-in is represented as $T = \{T_1, T_4, T_3\}$.

As shown in Table 2, the user reverse trajectory pattern matrix $R(M)$ is obtained in reverse time based on the user trajectory set T_r. In Algorithm 2, M and $R(M)$ are taken as inputs to recommend set T of location types that meets the trajectory pattern.

Algorithm 2 shows the recommendation process of location types when generalizing sensitive check-in. The algorithm takes into account three kinds of location situations of sensitive check-in in the trajectory. When the sensitive check-in location is located at the beginning of the trajectory, the reverse trajectory pattern matrix is used for the recommended location type of the sensitive check-in (lines 2 and 3). When the sensitive check-in location is located at the end of the trajectory, the trajectory pattern matrix is used for the recommended location type of the sensitive check-in (lines 4 and 5). When the sensitive check-in location is located at the nonhead-tail location of the trajectory, the combination of two trajectory pattern matrices is used to recommend location type for the sensitive check-in (lines 6 and 7). Finally, the first s location types with high recommendation probability values are selected from the recommended location types and put into the set T of location types.

4.2. Search for Specific Generalization Location. This section mainly introduces the use of a generalized location search algorithm (Algorithm 3) to generate location queue L. According to the recommended location type, the specific location of the corresponding location type should be searched in the query area. At the same time, the specific location selected should meet the time accessibility. For example, the query areas of sensitive check-in (l_7, t_{53}) are

Table 2: Reverse trajectory pattern matrix.

Matrix $R(M)$	T_1	T_2	T_3	T_4
T_1	1/6	0	1/6	2/3
T_2	2/3	0	1/3	0
T_3	1/3	0	0	2/3
T_4	0.2	0.2	0.4	0.2

Table 3: Distance index for each sensitive location.

l_1	l_5^6	l_3^1	l_2^2			
	0.10	0.20	0.40			
l_2	l_1^6	l_5^1				
	0.14	0.18				
l_4	l_1^8	l_9^4	l_4^7			
	0.20	0.22	0.65			
l_5	l_7^2	l_4^6	l_1^1			
	0.24	0.36	0.45			
l_7	l_1^1	l_1^2	l_3^2	l_4^1	l_4^2	l_4^3
	0.18	0.19	0.58	0.68	0.69	0.72
l_8	l_4^1	l_4^1	l_1^1	l_4^3		
	0.11	0.32	0.50	0.75		

Table 4: Generalized location quantity allocation.

Location type	T_1	T_2	T_3	T_4
Number of locations	2	0	1	3
Allocated quantity	2	0	0	1

circular areas with check-in location l_5 at the previous time and check-in location l_8 at the later time as the center and $\bar{v}(t_{53} - t_{52})$ and $\bar{v}(t_{54} - t_{53})$ as the query radius, respectively, where $\bar{v}$ is the average speed calculated from the user's historical trajectory. First, the specific locations corresponding to each location type in the query area are put into location

Input: generalized location type quantity array S, real sensitive location l_s, privacy protection threshold k, location queue L;
Output: generalized location candidate array $Cand_l$.
1. Initialize $Cand_l[T_i]$=0, $i \in$[1,|S|];
2. while lens ($Cand_l$)<k-1 do
3. for each location type $T_i \in S$ && $Cand_l[T_i] <= S[T_i]$ do
4. if lens ($Cand_l$)==0 then
5. Select the location furthest from the real sensitive location from the generalized locations in T_i and add it to $Cand_l$;
6. else
7. Select the location with the maximum Score in T_i and add it to $Cand_l$;
8. Return $Cand_l$;

Algorithm 5: The dummy location selection algorithm.

queue L, namely, $L = [l_1^1, l_1^2, l_4^1, l_4^2, l_4^3, l_3^2]$. There are two locations belonging to location type T_1, there are three locations belonging to location type T_4, there is one location belonging to location type T_3, and then, $l.D(R)$ is defined to represent a group of locations within distance R of sensitive location l. As shown in Table 3, search the location of distance sensitive location l_7 within the 0.7 km, that is, $l_7.D(0.7) = \{l_1^1, l_1^2, l_4^1, l_4^2, l_3^2\}$, and the unit of distance is kilometer (km). The generalized location search algorithm proposed in this paper implements the breadth-first search on the query area to realize the location search. $l.D$ is used to store the searched candidate locations and the corresponding distance index (D-index), and then, the binary search algorithm is used to select the qualified locations, in order to save the running time of the algorithm.

Definition 9 Distance index (D-index). Given a sensitive location l, the distance index (D-index) between this location and other locations is defined as a list $l.D$. The elements stored in the list are candidate locations and the distance data between each candidate location and the sensitive location l, and the distance data in $l.D$ are arranged in order from small to large.

4.3. Generalized Location Quantity Allocation. This section mainly introduces the generalized location quantity allocation algorithm based on the trajectory pattern (Algorithm 4). The purpose is to determine the number of specific locations allocated for the recommended location type and to ensure the maximum similarity of the trajectory pattern matrix. As shown in Table 4, five generalized locations $(l_1^1, l_1^2, l_4^1, l_4^2, l_3^2)$ are found for sensitive check-in (l_7, t_{53}) through the generalized location search algorithm. Because the privacy protection threshold k is 4, so three of the five generalized locations are selected to ensure the maximum similarity of the trajectory pattern matrix. In the generalized location allocation algorithm based on trajectory pattern, the same generalized location type as the sensitive check-in location is first assigned (line 2), so two generalized locations of location type T_1 are assigned. The selection of the remaining generalization location is determined by adding a generalization location of different location type at a time and calculating the similarity value of the corresponding trajectory pattern matrix (lines 4-9). When the generalized location type T_3 is added, the similarity of trajectory pattern is 99.81%. When the generalized location T_4 is added, the similarity of trajectory pattern is 99.93%. So, a generalization location of type T_4 is assigned and returns a generalization location type quantity array S.

4.4. Selection of Candidate Generalized Location. This section mainly introduces the selection of candidate generalized locations by candidate location selection algorithm (Algorithm 5). The generalized location candidate array needs to meet two conditions: (1) the locations in the array are as scattered as possible, which can effectively prevent the antideduction of the attacker. (2) The center location of the area formed by each location is as close as possible to the sensitive check-in location, which ensures that the anonymous trajectory is similar to the original trajectory. Among them, the traditional method to ensure the dispersion between locations is to calculate the sum of the distance between locations $\sum_{i \neq j} \text{dist}(l_i, l_j)$. However, the product of the distance method $\prod_{i \neq j} \text{dist}(l_i, l_j)$ can better reflect the dispersion of locations in most cases. As shown in Figure 3, A and B are selected generalization locations, and C and D are to be selected generalization locations. When selecting the third generalization location, both C and D can meet the requirements if the sum of distance method is used, because $\text{tr\_dist}(D, A) + \text{tr\_dist}(D, B) = \text{tr\_dist}(C, A) + \text{tr\_dist}(C, B)$. However, the product of distance method is used, and we should choose the generalization location C, because $\text{tr\_dist}(C, A) * \text{tr\_dist}(C, B) > \text{tr\_dist}(D, A) * \text{tr\_dist}(D, B)$, and the anonymous region formed by the generalization location A, B, and C is more scattered, so the product of distance method is adopted in this algorithm.

4.5. Algorithm Complexity Analysis. In the MC-LTR algorithm, generalized location types are recommended through the trajectory pattern matrix. Suppose $|M|$ is the order of the user trajectory pattern matrix, so, the time complexity of the MC-LTR algorithm is $O(|M| + |M|\log_2^{|M|})$. In the GLS algorithm, we use the binary search algorithm to find specific generalized locations that match the location type, and the algorithm complexity of the location queue at any sensitive location is $O(\log_2^{|D-\text{index}[l_i]|})$. In the LATP algorithm, we need to assign generalized locations for the recommended location type, because the privacy protection threshold is k,

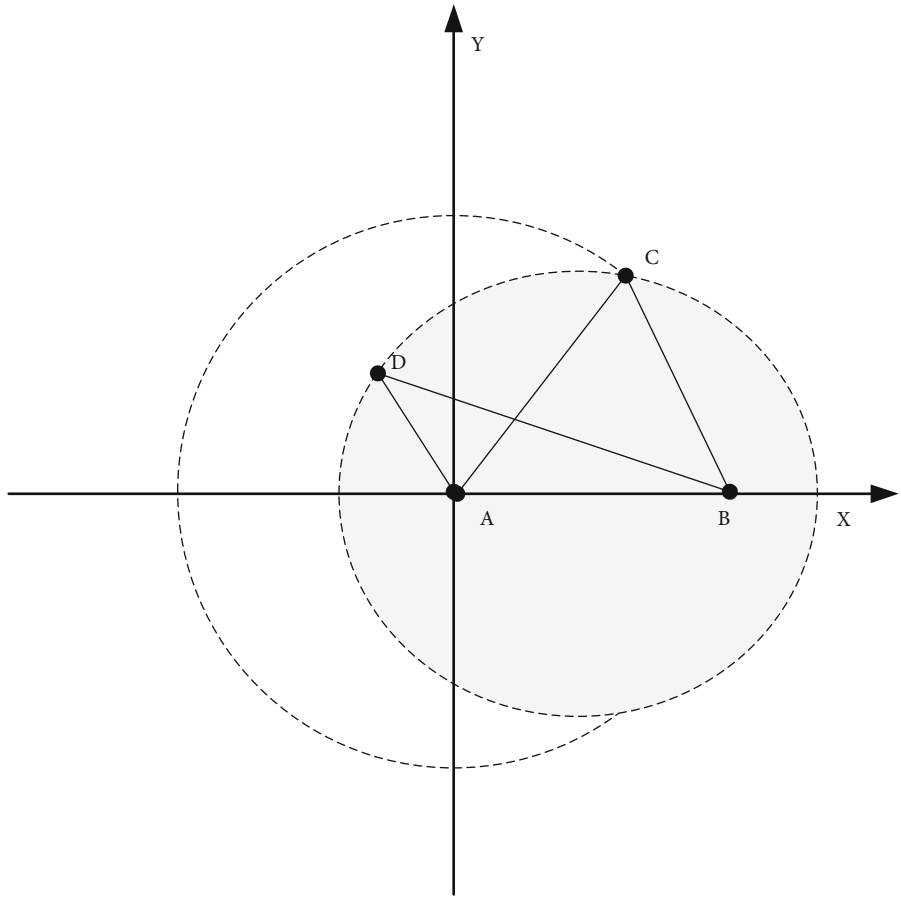


Figure 3: Distance product scene graph.

Table 5: Experimental data statistics.

Experimental data set	Number of users	Number of locations	Number of check-ins	Number of trajectories	Number of location types
Brightkite	5000	274761	3185493	168	766316
Gowalla	5000	193989	1501739	121	436665

so, the generalization location needs to be allocated through $k-1$ cycles, and each cycle needs to update the matrix and calculate the matrix similarity. The algorithm's time complexity is $O((k-1)[(|s|-1)(3+|M|*|M|)]+2)=O(k*s|M|^2)$. In the DLS algorithm, we need to choose $k-1$ candidate generalization locations from the generalization region, and it is necessary to judge $|S[T_i]|$ locations every time, so the algorithm complexity is $O(k*|S|)$. Therefore, the time complexity of the CPP algorithm is $O(k*s|M|^2)$.

5. Experimental Evaluation and Analysis

This section analyzes and evaluates the performance of the proposed check-in privacy protection algorithm based on generalized check-in location. The data used in the experiment comes from two real data sets Brightkite and Gowalla disclosed by the complex network analysis platform of Stanford University. The map data of California where these two data sets are located are also obtained. Firstly, this paper deletes and filters users whose cumulative check-in days are less than 50 days in the data set and then deletes the trajectory that contains only one check-in data in a single trajectory. Finally, this article selects 5000 users and their corresponding data from the two data sets. Table 5 shows the relevant information of the experimental data.

This paper proposes a check-in privacy protection algorithm based on the generalization of check-in location (recorded as CPP), compared with the dummy location selection algorithm based on multiobjective optimization (recorded as enhanced DLS) [12] and the location privacy protection algorithm based on random selection method (recorded as RS) [7]. The performance of the algorithm is analyzed by comparison, and the influence of the parameters involved in the algorithm on the algorithm is evaluated. In the test, the value range of privacy protection anonymous parameter k is from 2 to 32.

The software and hardware environment of this experiment are as follows: (1) hardware environment: Intel Xeon 3.90 GHz CPU and 256 GB; (2) operating system platform:

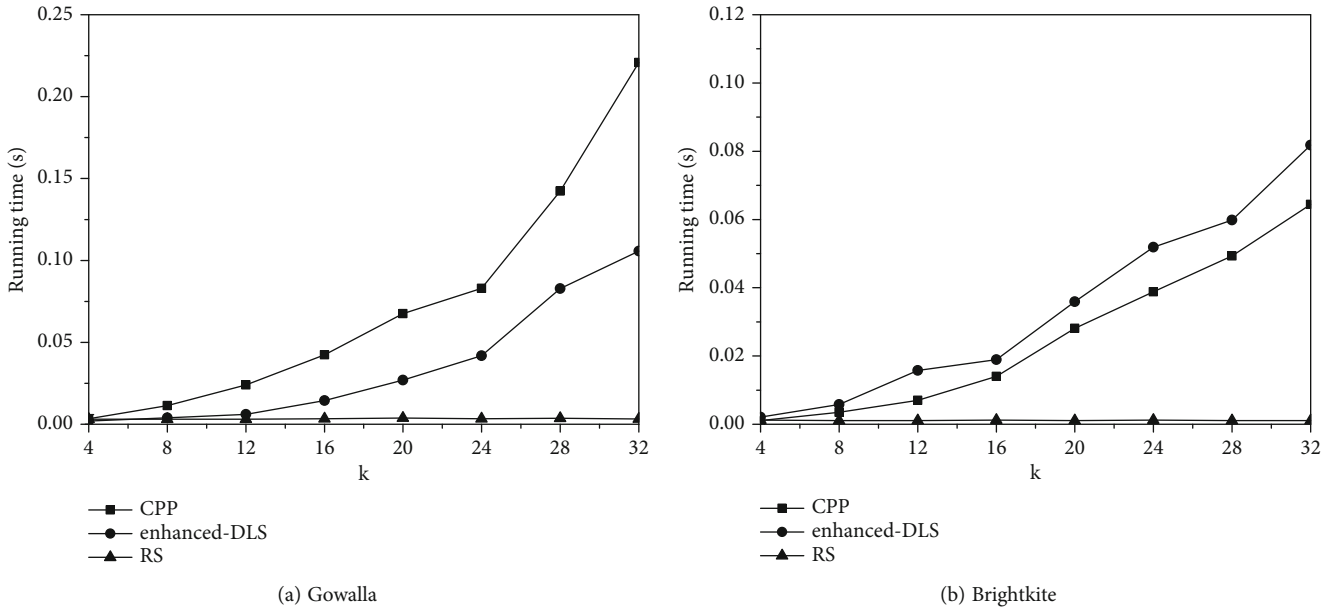


Figure 4: Change situation of running time.

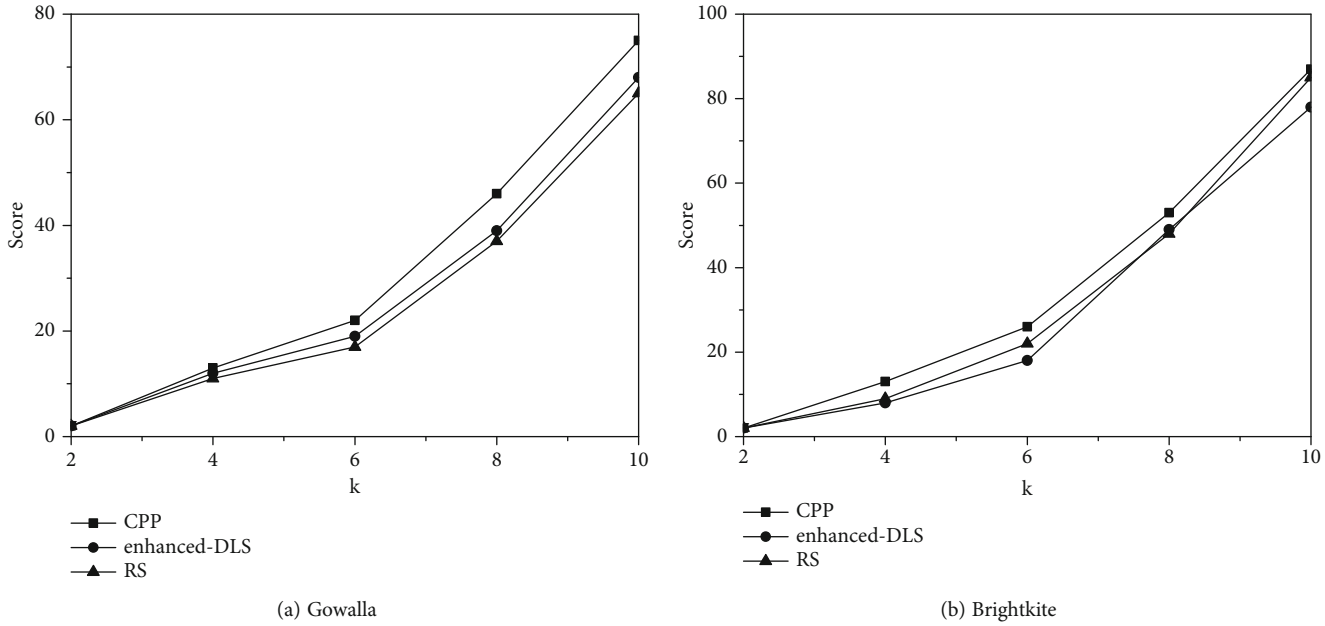
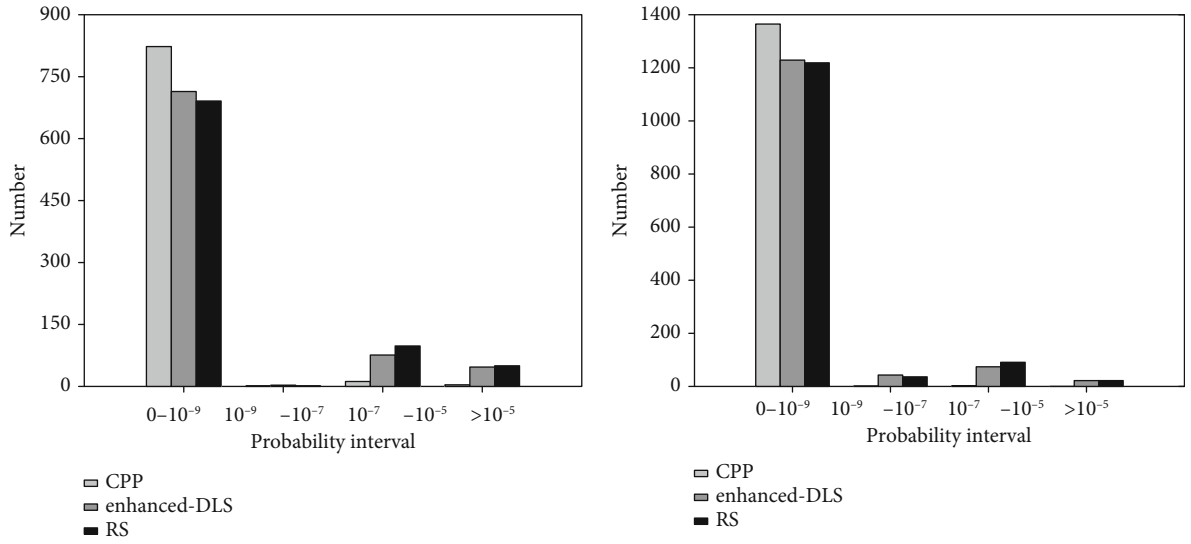


Figure 5: Change situation of score value.

Microsoft Windows 10; and (3) programming environment: Python language, Pycharm.

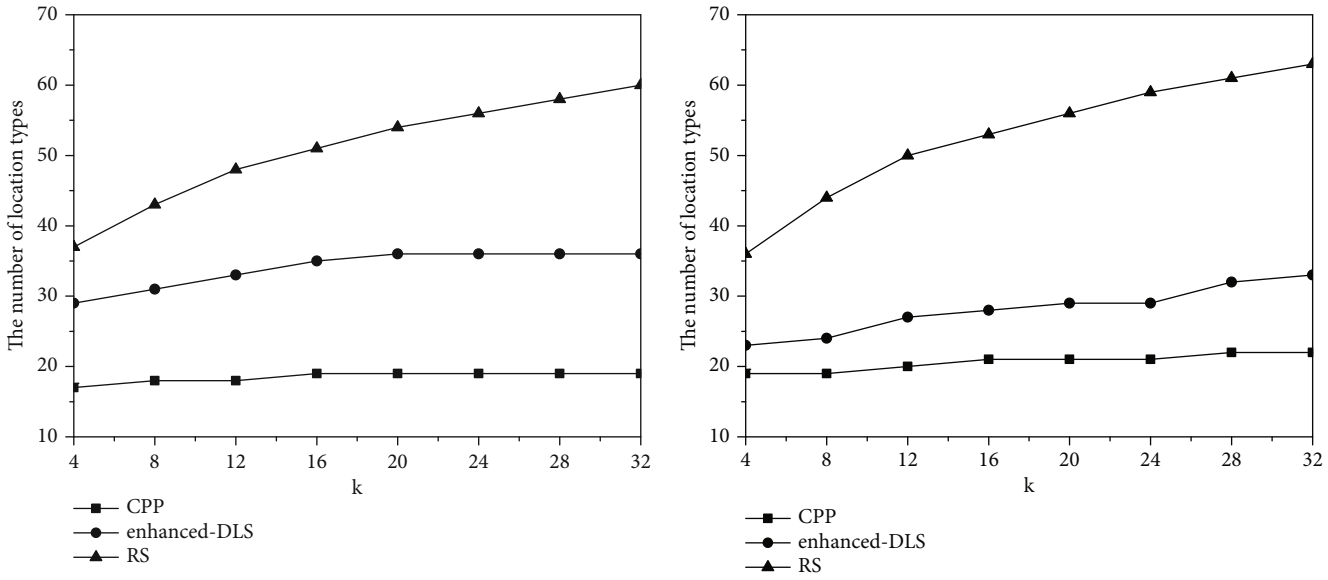
This section compares the performance of the CPP algorithm, enhanced DLS algorithm, and RS algorithm by analyzing the running times of the algorithm, the change of score value, and data availability. The following can be seen from Figures 4 and 5: (1) The running time of the three algorithms increases with the increase of the privacy protection threshold k. The running time of the CPP algorithm is between the RS algorithm and the enhanced DLS algorithm. The running time of the enhanced DLS algorithm changes significantly with the increase of k, and the running time of the RS algorithm is the smallest and tends to be stable. The enhanced DLS algorithm should consider the influence of the query probability and the entropy when selecting generalized locations, and the running time will be increased with the number of generalized locations, The CPP algorithm saves the running time by proposing a reasonable and effective location search algorithm. (2) The *score* value measures the degree of dispersion between locations and the distance between the anonymous trajectory and the original trajectory.

When the score value is larger, it means that the selected generalized location and sensitive check-in location are



(a) Gowalla (b) Brightkite

Figure 6: Distribution situation of probability change.



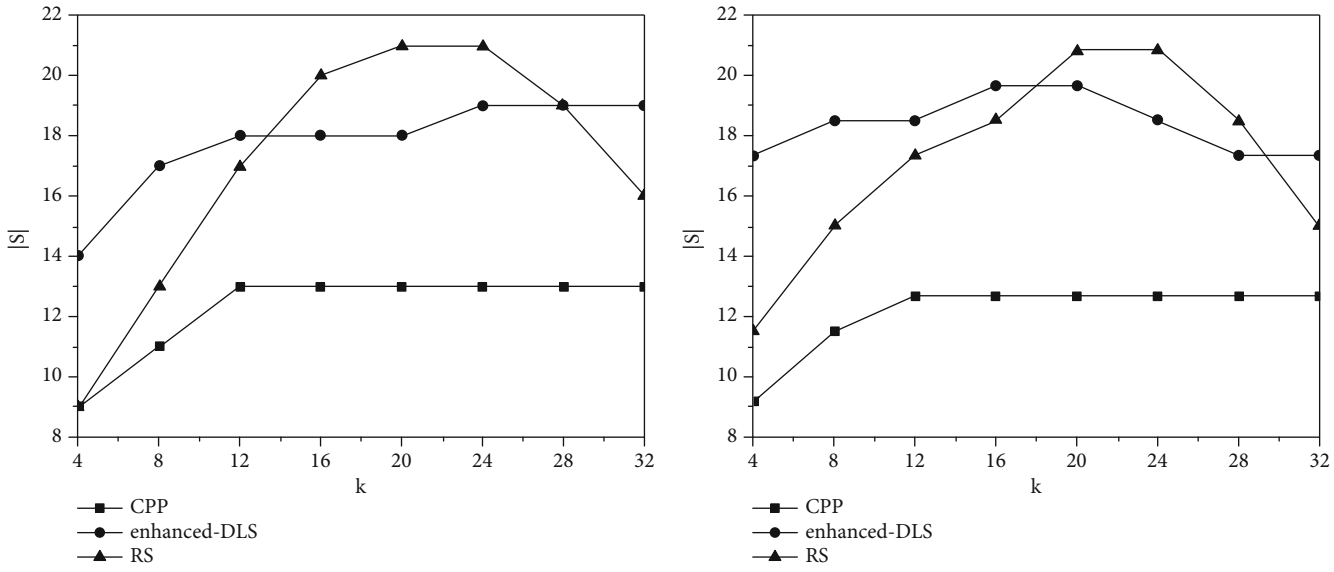
(a) Gowalla (b) Brightkite

Figure 7: Change situation of location types.

more dispersed, and the distance between the generalized trajectory and the real trajectory is closer. With the increase of the privacy protection threshold k, the score of the three algorithms increases significantly. The CPP algorithm has the better performance in score value because the CPP algorithm uses the heuristic rules to select each generalization location, and it ensures that each selected generalized location can keep the maximum score value. However, the random selection of each generalized location will lead to uncertainty, and the size of the *score* value to be unstable. When the order of magnitude of score value is too large, this paper uses the logarithm of *score* value to express it.

The availability of measurement data can be evaluated from three aspects: the change of user trajectory pattern, the change of access location type, and the change of access location points. The following can be seen from Figures 6–8:

(1) According to the position type transition probability difference before and after anonymity of the trajectory pattern matrix, it can be divided into four intervals: $0 \sim 10^{-9}, 10^{-9} \sim 10^{-7}, 10^{-7} \sim 10^{-5}, >10^{-5}$, counting the quantity distribution of each interval. The probability difference greater than 98% in the CPP algorithm falls in the $0{\sim}10^{-9}$ interval, and it shows that the change of location type transition probability is small, the similarity of the trajectory pattern matrix before and after anonymity is high, while the performance of enhanced DLS algorithm



(a) Gowalla

(b) Brightkite

Figure 8: Change situation of specific location.

and RS algorithm is lower than that of CPP algorithm. The similarity of the trajectory pattern matrix before and after anonymity is high, and the performance of the enhanced DLS algorithm and RS algorithm is lower than the CPP algorithm

(2) *The Changes in the Type of User Access Location*. The total number of original location types visited by a user is 38. It can be seen from the figure that the number of access location types of the CPP algorithm changes little with the increasing of the k, and the algorithm will not generate new location types. The access location type of the enhanced DLS algorithm increases gradually as the increasing value of the k, while the number of access location types of the RS algorithm changes significantly with the increasing of the k. Because the CPP algorithm recommends the location type for the user according to the user's trajectory pattern, the generalized location selected by the random method is uncertain, and the location type of the generalized location will exceed the range of the user's original access location type.

(3) For the user's access location change index, set the number of user visits to each access location before generalization as n_i, and after generalization, the number of user visits to each location becomes $n_i{}'$; the change in the number of visits for each location is defined as $\Delta n_i = |n_i - n_i{}'|$. Set a standard number threshold N and a location set S, and put the locations with the change of the number of visits greater than or equal to the standard number threshold into the set S, symbolized as $S = \{l_i | \Delta n_i \geq N\}$. The value of $|S|$ represents the number of position points in the set S. The smaller the value of $|S|$, the more stable the number of visits of the user to each access location after anonymity, and the better anonymity. It can be seen from Figure 8 that the CPP algorithm proposed in this paper has the smallest $|S|$ value, which is much lower than the other two algorithms, so the anonymous protection effect is the best. The RS algorithm is easy to generate new locations when selecting generalized locations, and the number of new locations is uncertain, so the $|S|$ value is larger

6. Conclusion

In this paper, a check-in privacy protection algorithm based on check-in location generalization for sensitive check-in protection is proposed for the first time and can be applied to blockchain transactions to solve the privacy protection problem of transaction users' sensitive information. Considering the user's trajectory pattern factor, the algorithm recommends the location type of the generalized check-in location for the user and selects generalized locations that can ensure the minimum change of trajectory pattern. Experimental research based on real check-in data sets shows that the CPP algorithm can effectively protect the sensitive check-ins in the trajectory, greatly reduce the probability of the attacker identifying the real sensitive check-ins, and maintain the high availability of the trajectory pattern data. This method is suitable for protecting the location in the area with dense geographical density. However, the k-anonymity method may not be implemented in areas with sparse geographical density. The solution to the above problem needs to be further studied.

Data Availability

All data included in this study are available upon request by contact with the corresponding author.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is partly supported by the National Natural Science Foundation of China (No. 61802268).

References

[1] L. Liu, J. Feng, Q. Pei et al., "Blockchain-enabled secure data sharing scheme in mobile-edge computing: an asynchronous advantage actor-critic learning approach," *IEEE Internet of Things Journal*, vol. 8, no. 4, pp. 2342–2353, 2021.

[2] L. Tan, Y. Keping, N. Shi, C. Yang, W. Wei, and L. Huimin, "Towards secure and privacy-preserving data sharing for COVID-19 medical records: a blockchain-empowered approach," *IEEE Transactions on Network Science and Engineering*, vol. 9, no. 1, pp. 271–281, 2022.

[3] Y. Sun, J. Liu, K. Yu, M. Alazab, and K. Lin, "PMRSS:privacy-preserving medical record searching scheme for intelligent diagnosis in IoT healthcare," *IEEE Transaction on Industrial Informatics*, vol. 18, no. 3, pp. 1981–1990, 2022.

[4] L. Liu, C. Chen, Q. Pei, S. Maharjan, and Y. Zhang, "Vehicular edge computing and networking: a survey," *Mobile Networks and Applications*, vol. 26, no. 3, pp. 1145–1168, 2021.

[5] J. Feng, L. Liu, Q. Pei, and K. Li, "Min-Max cost optimization for efficient hierarchical federated learning in wireless edge networks," *IEEE Transactions on Parallel and Distributed Systems*, p. 1, 2022.

[6] L. Zhao, Z. Li, A. Al-Dubai et al., "A novel prediction-based temporal graph routing algorithm for software defined vehicular networks," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–16, 2021.

[7] Z. Liang, T. Zheng, M. Lin, A. Hawbani, J. Shang, and C. Fan, "SPIDER: a social computing inspired predictive routing scheme for softwarized vehicular networks," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–12, 2021.

[8] L. Zhao, H. Li, N. Lin, M. Lin, C. Fan, and J. Shi, Eds., "Intelligent content caching strategy in autonomous driving toward 6G," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–11, 2021.

[9] L. Liu, M. Zhao, M. Yu, M. A. Jan, D. Lan, and A. Taherkordi, "Mobility-aware multi-hop task offloading for autonomous driving in vehicular edge computing and networks," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–14, 2022.

[10] M. Wang, Y. Lin, Q. Tian, and G. Si, "Transfer learning promotes 6G wireless communications: recent advances and future challenges," *IEEE Transactions on Reliability*, vol. 70, no. 2, pp. 790–807, 2021.

[11] L. Wang and X. F. Meng, "Location privacy preservation in big data era: a survey," *Journal of Software*, vol. 25, no. 4, pp. 693–712, 2014.

[12] L. Sweeney, "K-anonymity: a model for protecting privacy," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 5, pp. 557–570, 2002.

[13] C. Dwork, "Differential privacy," in *Proc of the 33rd International Colloquium on Automata, Languages and Programming*, pp. 1–12, Springer-Verlag, Berlin, 2006.

[14] Z. Feng, *Research on Location Privacy-Preserving Nearest Neighbor Query Based PIR*, Southeast University, 2017.

[15] Z. Feng and N. Wei-Wei, "Pseudo-random number encryption based on location privacy preserving nearest neighbor querying," *Journal of East China Normal University*, vol. 2015, no. 5, pp. 128–142, 2015.

[16] F. Deldar and M. Abadi, "PDP-SAG: personalized privacy protection in moving objects databases by combining differential privacy and sensitive attribute generalization," *IEEE Access*, vol. 7, pp. 85887–85902, 2019.

[17] H. Kido, Y. Yanagisawa, and T. Satoh, "An anonymous communication technique using dummies for location-based services," in *Proceedings of the 2005 International Conference on Pervasive Services*, pp. 88–97, Piscataway, 2015.

[18] M. Gruteser and D. Grunwald, "Anonymous usage of location-based services through spatial and temporal cloaking," in *Proceedings of the First International Conference on Mobile Systems, Applications, and Services*, pp. 31–42, San Francisco, CA, USA, 2003.

[19] M. Gruteser and X. Liu, "Protecting privacy in continuous location trajectorying applications," *IEEE Security and Privacy*, vol. 2, no. 2, pp. 28–34, 2004.

[20] B. Gedik and L. Ling, "Protecting location privacy with personalized k-anonymity: architecture and algorithms," *IEEE Transactions on Mobile Computing*, vol. 7, no. 1, pp. 1–18, 2008.

[21] W. Jie, W. Chunru, and M. A. Jianfeng, "Dummy location selection algorithm based on location semantics and query probability," *Journal on Communications*, vol. 41, no. 3, pp. 53–61, 2020.

[22] B. Niu, Q. Li, and X. Zhu, "Achieving k-anonymity in privacy-aware location-based services," in *IEEE INFOCOM 2014-IEEE Conference on Computer Communications*, Toronto, ON, Canada, 2014.

[23] H. Lu, C. S. Jensen, and M. L. Yiu, "PAD: privacy-area aware, dummy-based location privacy in mobile services," in *Acm International Workshop on Data Engineering for Wireless & Mobile Access*, pp. 16–23, Vancouver, Canada, 2008.

[24] P. Xiong, L. Zhang, and T. Zhu, "Reward-based spatial crowdsourcing with differential privacy preservation," *Enterprise Information Systems*, vol. 11, no. 10, pp. 1500–1517, 2017.

[25] X. Qiyuan, C. Zhenping, and F. Baochuan, "Hybrid location privacy protection based on differential privacy," *Computer Applications and Software*, vol. 36, no. 6, pp. 296–301, 2019.

[26] X. Ping, T. Zhu, and P. Lei, "Privacy Preserving in Location Data Release: A Differential Privacy Approach," in *Pacific Rim International Conference on Artificial Intelligence*, pp. 183–195, Springer, Cham, 2014.

[27] S. Papadopoulos, S. Bakiras, and D. Papadias, "Nearest neighbor search with strong location privacy," *Proceedings of the VLDB Endowment*, vol. 3, no. 1-2, pp. 619–629, 2010.

Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 8482415, 16 pages
https://doi.org/10.1155/2022/8482415

Research Article

A Data Management Model for Intelligent Water Project Construction Based on Blockchain

Zhoukai Wang[1,2] **Kening Wang**[3] **Yichuan Wang**[1,2] **and Zheng Wen**[4]

[1]*School of Computer Science and Engineering, Xi'an University of Technology, Xi'an, 710048, China*
[2]*Shaanxi Provincial Key Laboratory of Network Computing and Security Technology, Xi'an, 710048, China*
[3]*School of Automation and Information Engineering, Xi'an University of Technology, Xi'an, 710048, China*
[4]*School of Fundamental Science and Engineering, Waseda University, Tokyo 169-8050, Japan*

Correspondence should be addressed to Zhoukai Wang; zkwang@xaut.edu.cn

Received 7 December 2021; Accepted 16 February 2022; Published 9 March 2022

Academic Editor: Qingqi Pei

The engineering construction-related data is essential for evaluating and tracing project quality in industry 4.0. Specifically, the preservation of the information is of great significance to the safety of intelligent water projects. This paper proposes a blockchain-based data management model for intelligent water projects to achieve standardization management and long-term preservation of archives. Based on studying the concrete production process in water conservancy project construction, we first build a behavioral model and the corresponding role assignment strategy to describe the standardized production process. Then, a distributed blockchain data structure for storing the production-related files is designed according to the model and strategy. In addition, to provide trust repository and transfer on the construction data, an intelligent keyless signature based on edge computing is employed to manage the data's entry, modification, and approval. Finally, standardized and secure information is uploaded onto the blockchain to supervise intelligent water project construction quality and safety effectively. The experiments showed that the proposed model reduced the time and labor cost when generating the production data and ensured the security and traceability of the electronic archiving of the documents. Blockchain and intelligent keyless signatures jointly provide new data sharing and trading methods in intelligent water systems.

1. Introduction

In the water conservancy project management, archives have the characteristics of large numbers and comprehensive coverage, and they play an essential role in all aspects of engineering construction. With the increasing investment of water conservancy projects, the scale gradually grows, and the project gradually becomes complex. The management of water conservancy project archives also faces more and more problems, which restrict the development of water conservancy projects. On the other side, the traditional file management mode can no longer adapt to the rapidly developing economic needs, so the introduction of digital archives for water conservancy projects has become an inevitable trend [1, 2, 3]. However, because the construction of water conservancy projects requires the global deployment and management of various units and resources, making the digitization process of its archives difficult, the status of library management leading to the water conservation institutions requires acceleration transformation [4].

At present, digital archives of water conservancy projects have less relevant research in foreign countries, and the research in China is also in the initial stage [5, 6]. Although the new "Archives Law of the People's Republic of China" provides legal and policy guarantees for the informationization of construction files of water conservancy project construction, the relevant research and application are still focused on the initial stage of construction [7]. Other important aspects of water conservancy project construction, such as concrete production and mixing, and metal structure installation, still lack effective information management means [8]. In addition, the current digital file management

methods are relatively simple, with the drawbacks of poor antitampering and antirepudiation capabilities, and their application range is also limited. In total, the current digital file management methods cannot undertake the engineering construction works involving significant safety needs [9].

In response to the shortcomings of traditional file management methods, this paper introduces blockchain and keyless signature techniques [10], takes the concrete mixing process as the research object, and conducts research on data management in intelligent water conservancy construction. The main contributions of this paper are demonstrated as follows.

(1) By employing the smart keyless signatures, this paper established a paperless concrete production and operation management model to monitor the concrete mixing process and prevent data tampering during the process

(2) With the help of the consortium blockchain, this paper built up an intelligent document storage method to effectively supervise the progress, quality, and safety of concrete production and then explore general methods for encryption, storage, and traceability of production files

(3) Integrated with the corresponding model and method, this paper proposed a blockchain-based file management system for concrete mixing procedures and then implemented it in the Hanjiang to Weihe River Project to improve the production management capacity markedly

2. Motivation

Concrete mixing is a vital link in the construction of water conservancy projects, and many engineering archival documents are generated during the mixing process to record the concrete mixing details [11]. These files are crucial basic information for project quality control and problem tracing and are related to the whole life cycle safety of the project. However, the management of concrete production files has problems such as low informationization and insufficient security at present [12, 13]. Firstly, the current management method wastes paper. The volume of files related to concrete mixing production is enormous. The amount of grouting required for reservoir construction is usually more than 100,000 cubic meters, which will generate a massive amount of paper data that is difficult to store and manage. Secondly, the paper-based management method has less credibility. The manually dumped paper files are not standardized, and falsification of the paper files often occurs. Thirdly, the traceability of the paper files is feeble. Currently, the cataloging and archiving of concrete production files have not yet formed a strict and complete discipline and management system. Therefore, it is difficult to achieve practical traceability issues tracing. At last, the current management methods obtain insufficient security since the lack of security and confidentiality control measures for the massive paper files.

In response to the above problems, more and more researchers have devoted their efforts to studying the digital file management of water conservancy projects, especially for the informatization of the concrete mixing process, and preliminary research results have been achieved. The representative projects include the Jingtaichuan Dam in Gansu Province, the Daxing Water Conservancy Hub Project in Guizhou Province, and Chushandian Reservoir in Henan Province [14]. However, these research results still fail to completely solve the shortcomings of low antitampering ability and poor antirepudiation ability of digitized archives [15]. Digital archives are still exposed to risks, and they are difficult to effectively manage the concrete mixing procedure and ensure the procedure's safety.

The quality of concrete production and the management of related production documents are closely associated with the safety of people's lives and property. They have a high level of tamper-proof and repudiation-proof requirements [16]. Although the Chinese government has established the corresponding laws to push forward electronic signatures steadily, digital files are bound to be severe trust and security concerns when transmitted over the Internet and stored in centralized servers for long periods [17]. The higher the sensitivity of the data, the greater the risk of using high-tech means to "blacken" it. Apparently, there are substantial technical difficulties in achieving the highly informative management of concrete mixing files paperless. How to help the concrete mixing files and the corresponding data get rid of the security threat becomes a hot topic in the intelligent water conservancy field.

In 2009, blockchain technology was proposed to guarantee the security of data. Recently, applications based on blockchain have increasingly appeared in various fields of daily social life, such as finance, public services, culture and entertainment, data insurance, and general welfare [18–20]. However, the present archival research on the blockchain mainly focuses on the feasibility of document archive management and specific application methods [21]. Many scholars have proposed their application for standard archival management based on blockchains, such as museum archives, student archives, and medical information archives [22, 23]. Other scholars have also discussed the challenges and troubles that blockchain technology may face when applied in archival management [24, 25]. But there are only a few cases of the practical application of blockchain-based archive management in hydraulic engineering fields [26]. In summary, applying blockchain and related technologies in engineering construction archive management, especially to critical aspects such as concrete production, has received less attention from relevant studies domestic and abroad.

Based on the current research foundation in related fields, this paper takes the whole concrete production process as the research object, integrates blockchain technology with the specific needs of water conservancy projects, and improves the management quality of the electronic files in the concrete mixing process. Meanwhile, this paper also proposes a highly integrated information management system to guarantee the data security of each step in the concrete mixing procedure. The specific steps are as follows: first,

study the relationship between the different concrete production departments and establish a behavioral model describing the concrete production process; second, design and implement a distributed blockchain data structure for concrete production process management; third, use keyless signature technology to manage the type-in, modification, and approval process of the concrete production files; finally, all the files generated in the concrete production process are uploaded to the blockchain to achieve openness and transparency of the entire process, guaranteeing accurate traceability of production files and quick location of quality problems, thus effectively supervising the data quality and safety in the intelligent water conservancy projects construction.

3. Behavioral Model for the Concrete Production Process

3.1. Process Sorting and Role Assignment. To establish a behavioral model for the concrete production process, we first need to sort out the production process. As shown in Figure 1, the concrete production process is divided into raw material preparation and concrete production parts. Specifically, the raw material preparation part can be divided into the import and test subpart. In contrast, the concrete production part can be divided into the mix proportion design subpart, the concrete mixture subpart, and the concrete test subpart.

The raw materials for concrete production include cement, fly ash, admixtures, coarse aggregate, and fine aggregate. The first three materials are transported and supplied by the corresponding manufacturers, while the other materials can be produced by the mixing plant itself. As Figure 1 illustrates, in the material import stage, the quality and quantity reports are provided along with the entry of the purchased raw materials. When the raw materials are in storage, the laboratory of the mixing plant will sample and measure them and then record the report of the material test results in the ledger by computer. Besides, self-made raw materials like coarse and fine aggregates are also tested in detail and recorded by the laboratory of the mixing plant either. At last, all these raw material inspection reports are submitted to the supervision, and the supervision's approval allows the materials to participate in concrete production.

In the concrete production stage, the construction unit submits an application of concrete to the mixing plant. Moreover, the required concrete grade and performance requirements, the required quantity, and the use purpose are also informed to the mixing plant at the same time. After receiving the application, the laboratory personnel in the mixing plant will inspect the moisture content of sand and stone, check the exceeding and inferior grain in aggregate according to the relevant regulations, and then design the concrete mixing proportion. After the supervisor confirms the mixing ratio, the relevant mixing information is provided to the mixing plant. The mixing plant strictly follows the ratio, sets the raw material feeding value, and operates the mixing plant for concrete production. Besides, the raw material temperature and weighing information are recorded during the concrete mixing process according to the regulations. After the concrete mixture, samples are taken from the outlet of the mixing plant; then, the construction unit tests the samples' quality and forms the sample record and test report.

The role assignment could be set as follows by sorting the concrete production process. The main characters involved in the production process are the mixing plant, the laboratory of the mixing plant, the construction department of China Railway 12th Bureau (CR-12 in short), the laboratory of CR-12, the supervisor, and the third-party testing center. In specific, the mixing plant and its laboratory worked in the raw material preparation stage, while CR-12 and the corresponding laboratory worked in the concrete production stage. At last, the supervisor and the third-party testing center took part in every stage of the concrete production process to ensure the safe and reliable quality of the whole concrete production process.

3.2. Classification of Concrete Production Files. The second step of building the concrete production behavioral model is to classify all the files involved in the concrete production process according to their attributes. The files include the raw material performance testing records before concrete mixing, the concrete supply contact sheets, the descriptions on concrete mixing proportion, the records about the mixing process, the result of the concrete performance testing, forms related to each cycle errata, and summaries. The cooperation of these files is demonstrated as follows: The manufacturers supply the raw materials to the mixing plant for concrete production. After production, the mixing plant's laboratory samples the concrete and conducts a quality inspection. If the concrete meets the quality standards, it would be transported to the construction department of CR-12 by vehicles. After the additional tests conducted by the laboratory of CR-12, the construction department of CR-12 builds the water conservancy facilities with qualified concrete. At last, as a neutral third party, the supervisor keeps on inspecting the concrete by commissioning a third-party laboratory to sample and test the concrete at all stages during the production.

In total, after summarizing the files involved in the concrete production process, 50 categories of forms are obtained. There are a total of 29 forms related to raw materials, 1 contact sheet for material supply, 7 forms related to the concrete mixing process, 12 forms related to testing, and 1 form for erratum summary. The details are in Figure 2.

4. Distributed Blockchain Data Structure

4.1. General Framework Design. Based on the behavioral model, the distributed blockchain data structure can be constructed, and then, the preservation, categorization, and management of the concrete production-related archives can be achieved. The general framework design is illustrated in Figure 3. In Figure 3, the archives generated in concrete production are divided into temporal and spatial levels in the order of warehouse blocks, procedure blocks, branch

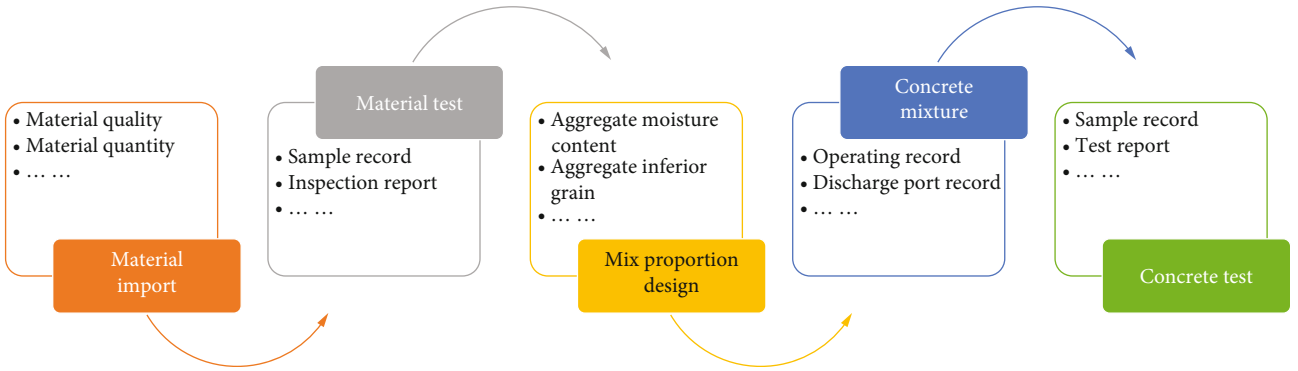


Figure 1: Schematic diagram of the concrete production process.

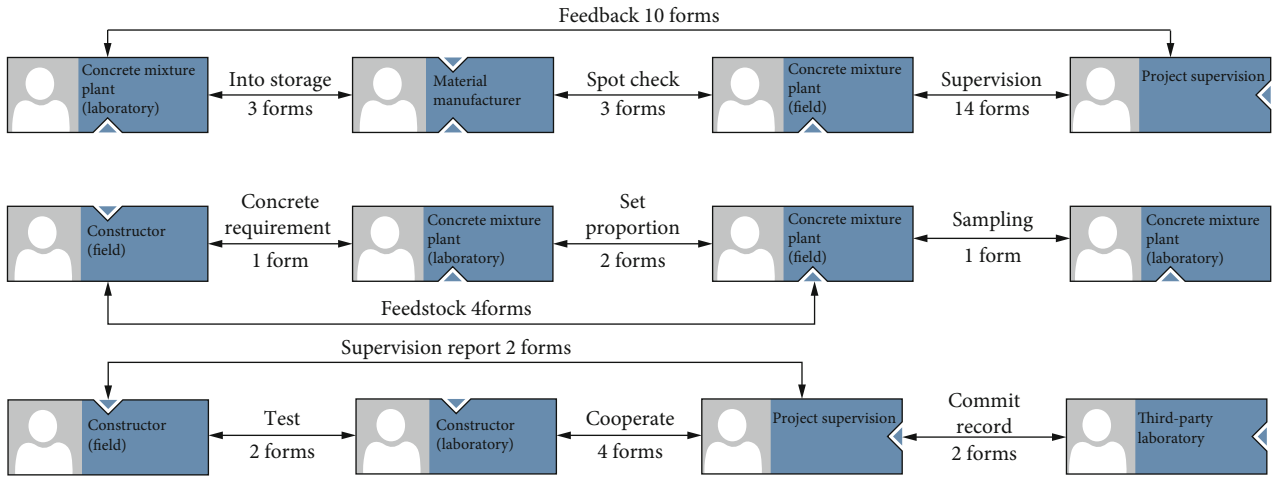


Figure 2: Classification and statistics of the concrete production files.

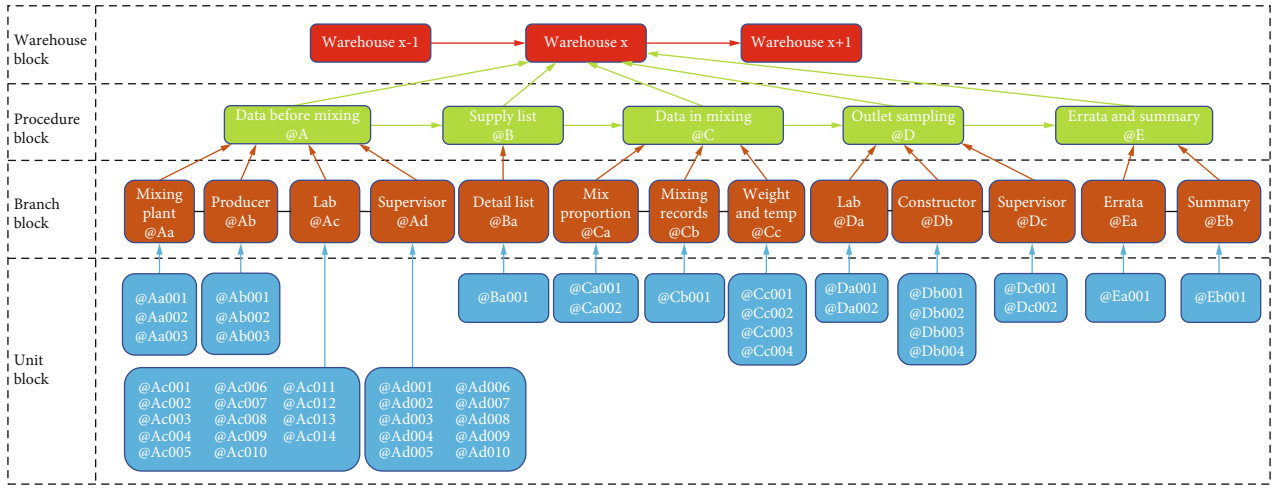


Figure 3: General framework of the distributed blockchain data structure.

blocks, and cell blocks. The structure in Figure 3 represents a comprehensive mixing procedure for a warehouse of concrete, and the warehouse is the fundamental quantity unit in the concrete production process. Inside the data structure, the subblock is composed of one or several distributed ledgers. The functions and properties of each subblock in the distributed blockchain are described below.

The top element in the distributed blockchain data structure is the warehouse block. The warehouse block contains all the files during the concrete mixing process. In the actual

environment, the whole construction procedure of the water conservancy project is often divided into unit projects, division projects, and cell projects. Furtherly, the cell projects are refined into a series of sequential subtasks, and the data and corresponding files generated in each subtask are formed as a warehouse block. Like the example in Figure 3, during the mixing process, the computers automatically record the production data for each tray of concrete, including the set and actual usage amount of the raw materials, the mixing time, the use of the concrete, and other detailed information. The warehouse blocks are made up of cyclic packets, and they are numbered sequentially from 0001 onwards in chronological order.

The procedure blocks are the blocks that indicate the specific flows of the concrete production. Note that the block could not be formed until the previous one is generated, and all blocks in the same layer are chained together in a tandem pattern. As shown in Figure 3, the concrete production process contains five procedure blocks: data before mixing, supply list, data in mixing, outlet sampling, errata, and summary.

The blocks in the branch block layer are the distributed ledgers created and categorized by different roles in the concrete mixing procedure. For example, @Aa,@Ab,@Ac,@Ad are the branch blocks under the same procedure block A in Figure 3; they represent the file collections in the mixing plant, the producer, the lab, and the supervisor, respectively.

The bottom layer in the distributed blockchain data structure is the unit block layer. In this layer, the unit blocks are the specific files, forms, images, or other media boundaries with archival requirements in branch blocks, marked with 001, 002, and so on. As shown in Figure 3, each unit block refers to one file created by a specific role.

Besides, the naming scheme of the proposed distributed blockchain data structure is as follows: Firstly, "@" and "#" in the front of each unit block number indicate if this block is shared or not. Secondly, the warehouse block is often divided into blocks for the raw material test, blocks for the concrete inspection, and the other blocks. Among them, the raw material inspection blocks record the samples and the test results of raw materials, such as 200~400 t a sampling unit of cement, 100~200 t a sampling unit of fly ash, and 50 t a sampling unit of admixture. Thirdly, if the unit block is shared, the provenance of the shared data should be indicated, and the indication method is to add the name of the warehouse block which contains the shared block. For instance, if the cement test report is "×××××Ac013," suppose that a new cement test report is generated in warehouse block "0020," then its number is "0020#Ac013." If the next five bunker blocks "0021," "0022," "0023," "0024," and "0025" need to quote the previous report rather than generating new cement inspection reports, then the quoted report is named as "0020@Ac013." But if the warehouse block "0026" generates a new cement inspection report, then the name of the report is "0026#Ac013."

4.2. Distributed Storage Architecture for Digital Archives. Based on the design of blockchain data structure, this paper classifies the files according to different production roles and then stores them in distribution. Specifically, the main characters participating in the concrete production procedure keep their own files locally. For instance, the construction department that initialized and transmitted the supply list would leave a copy of the list in the server of CR-12. Similarly, if the mixing plant initiates the batching notification form, then the form is stored in the computer of the mixing plant. The rules for the rest of the file storage locations are similar, except that the files that are shared by different branches should be stored by both the sending and the receiving units.

Moreover, the data-sharing scheme is another crucial part of distributed storage architecture, and it consists of two parts: the data sharing between files and inside files. The data sharing between files means keeping the same sections' consistency and accuracy in different files. There is a mapping or logical relationship between information in some files and information in the other files during transmission. Therefore, when creating such files, we first store this information in public memory and then automatically obtain the corresponding data with the same content on different files. For illustration, "construction site, strength grade, collapse level, and planned quantity" in the batching order are derived from the contents "construction site, seepage and frost resistance, collapse level and outlet temperature, and concrete supply order" in the supply list. Similarly, the "oversize content" and "undersize content" on the batching notice come from the same contents on the coarse aggregate test records. However, if the shared data is inconsistent, we will issue warning messages to senders and receivers. Then, the file is rejected by the receiver until the sender makes corrections. During the revision, the character who makes the file first checks if the inconsistency is indeed caused by himself then resolves this dispute by amending the filled-in content. Otherwise, the inconsistency is caused by the incorrect data in the system. Then, the dispute will be temporarily put on hold through the consensus mechanism and resolved through the errata at the end of this warehouse block.

The data-sharing scheme inside files means that the files are shared between different warehouse blocks. For illustration, in the raw material test stage, an inspection form for raw materials may cover more than one warehouse block; then, these blocks share the same inspection form. As mentioned above, the specific method to distinguish the shared and the unshared data uses "@" and "#" symbols as indicators.

5. Edge Computing Supported Intelligent Keyless Signature

During the concrete mixing, every authentic and valid file requires the principal's signature of every department, and the signature means the approval of the file content. This signature process is represented as the form-filling operation in the proposed model. However, there is a risk of tampering with the file during the filling process. The traditional approach is to introduce asymmetric encryption technology in the file approval process to ensure the security of

transmission and the file's integrity. But this technical approach has certain management risks because it involves the management of an individual's private key. Therefore, this paper employs a keyless signature technology based on edge computing to standardize the form-filling process and provide security for electronic files.

5.1. Hash Tree Construction Based on Edge Computing. The fundamental method for data security during the file transmission is to use the hash function to make a calculation on the file and then regard the calculation result as a digital fingerprint to prove the file's authenticity. In detail, the proposed management model uses the SHA-256 hash algorithm to calculate the file, generate a 256-bit hash value, perform a series of operations with the hash value, and build up a hash tree. The process is in Figure 4.

In Figure 4, x_1 to x_8 represent the hash values calculated with the SHA-256 algorithm, and these values are the input of the leaf nodes in the hash tree. $h()$ denotes the hash function, and the vertical line represents the join operation, but $h(x_1 \mid x_2) \neq h(x_2 \mid x_1)$. The hash tree introduces the hash function to fulfill zero-knowledge proof and ensure that the file is authentic. For example, suppose the initial data x_3 knows the hash values $\{x_4, x_{12}, x_{58}\}$ and their position markers $\{1,0,1\}$. In that case, the root value can be recreated, thus proving that x_3 is involved in calculating the generated root value. In total, based on hash chains, goals including a fast comparison on massive data, locating the modified data, and constructing zero-knowledge proofs, can be easily achieved. The hash chain computing process is shown in Figure 5.

Further, to secure data transfer and file integrity from the spatial dimension, a large number of hash trees need to be aggregated into Merkle trees simultaneously, and edge computing is the best way to achieve such goals. A Merkle tree consists of a root node, some intermediate nodes, and a set of leaf nodes. Each leaf node is labeled with the hash value of the digital file, while intermediate nodes other than the leaf nodes are marked with the cryptographic hash of their child node labels. Creating a complete Merkle tree requires recursively hashing a set of nodes and inserting the generated hash nodes into the tree until only one hash node remains, which is also called the Merkle root. The construction process of the Merkle tree is in Figure 6.

As shown in Figure 6, Merkle trees are created and destroyed once per second. These trees are composed of a hierarchical network of geographically independent distributed computing nodes. Each operates in an asynchronous aggregation fashion, generating a hash tree by receiving hash values from its subtrees transmitting the hash root values to multiple parents. The aggregation process is theoretically unbounded and runs on top of virtual machines or dedicated hardware. Moreover, in a keyless signature system with a multilayer aggregation hierarchy, the acceptable theoretical limit of the system is 2^{64} signatures per second.

5.2. The Intelligent Keyless Signature System. The keyless signature system based on Merkle trees is shown in Figure 7, and the specific tree construction process can be described as follows. Firstly, the department participating in the concrete mixing procedure submits the hash value (the blue dots in Figure 7) of the file to the customized keyless signature gateway. Secondly, the adjacent hash values are connected in series, and then, an additional hash operation on the concatenated values is performed again to calculate the result. Subsequently, the newly calculated hash value is submitted to the upper layer for serial hash operation until the Merkle tree's root is created. Finally, the keyless signature gateway returns a keyless signature to the department. The keyless signature contains the hash value submitted in the previous step and the sequence to regenerate the hash root value. This keyless signature is a hash chain composed of coordinates like the red dots in Figure 7. With this keyless signature system, the concrete construction department can ensure the spatial integrity of electronic data.

Except for guaranteeing the spatial integrity of the electronic files, the intelligent keyless signature system based on Merkle trees can also ensure the temporal reliability of the electronic files. The mechanism is illustrated as follows: First, the keyless signature system stores the hash root values in a shared database called the calendar database while creating and destroying every second. Specifically, since 0 : 00, 0 seconds on January 1, 1970, each second of hash values has been regarded as a leaf node, forming a particular type of permanent hash tree, also known as a Merkle forest. The calendar hashes are periodically aggregated to generate the integrity code's hash value. In a keyless signature system, the calendar database's integrity code is regularly issued in electronic and paper form in the world media, as shown in Figure 8 [27]. After the integrity code is released in the electronic or paper-based public media, the authenticity of all signatures can be evaluated by tracing back the integrity code, thus ensuring the temporal integrity of the data. [28].

5.3. Signing and Verification of Production Files. Signing and verifying the production files based on keyless signature are illustrated in Figure 9. As the description at the top of Figure 9, when a file is created and needs to be signed during the concrete production process, first, the signatories make a hash calculation on the file with the SHA-256 function and then submit the hash value to the distributed keyless signature server. From the one-way nature of the hash function, it is clear that the hash value is only the credential for applying a keyless signature, so the privacy of the original file is still kept. In the second step, the keyless signature server that receives the hash value performs a calculation through the hash chain and returns a keyless signature starting from the root node of the Merkle tree to the signatories as a response. In the third step, the keyless signature server timely releases the integrity code through newspapers or other forms. Note that the integrity code is preserved in the online calendar database after its release.

The verification of signed files usually occurs in the file approval stage. As shown at the bottom of Figure 9, when the validator receives a signed file from the previous signatory, in order to verify the authenticity of the data, first and foremost, the received file and its corresponding keyless signature should be aggregated to conduct a hash

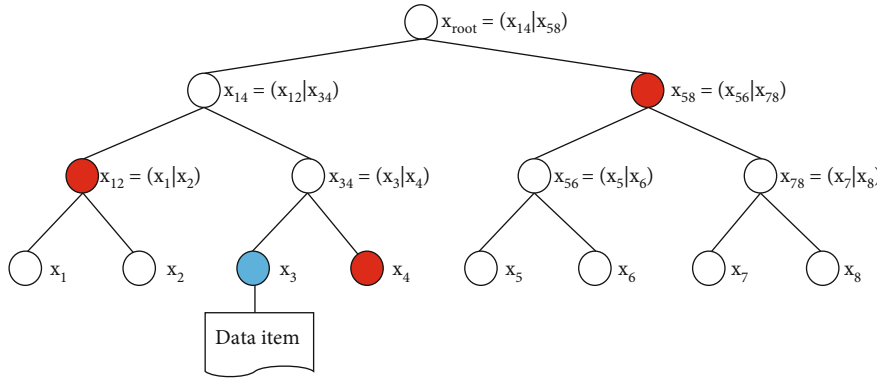


Figure 4: Schematic diagram on hash tree construction.

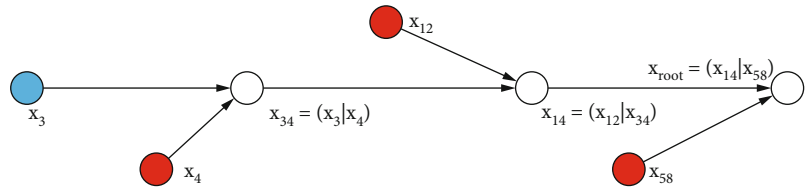


Figure 5: Schematic diagram on hash chain computing.

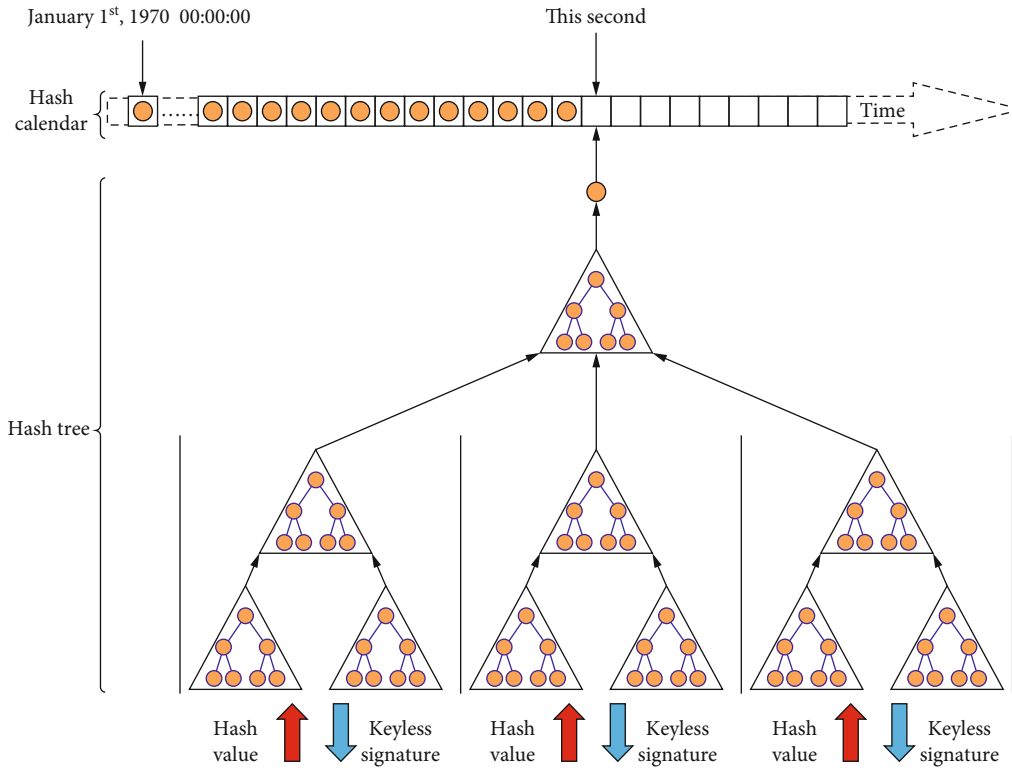


Figure 6: Parallel construction of the Merkle tree based on edge computing.

computation. Next, the integrity code associated with the signature file is figured out from the online database. Then, a comparison of the integrity code with the hash computation result is conducted subsequently. If the comparison result is consistent, it indicates that the signature data is accurate and trustworthy, the file transmission and approval process is in line with the standard requirements, and there is no tampering with the data. If the comparison result is

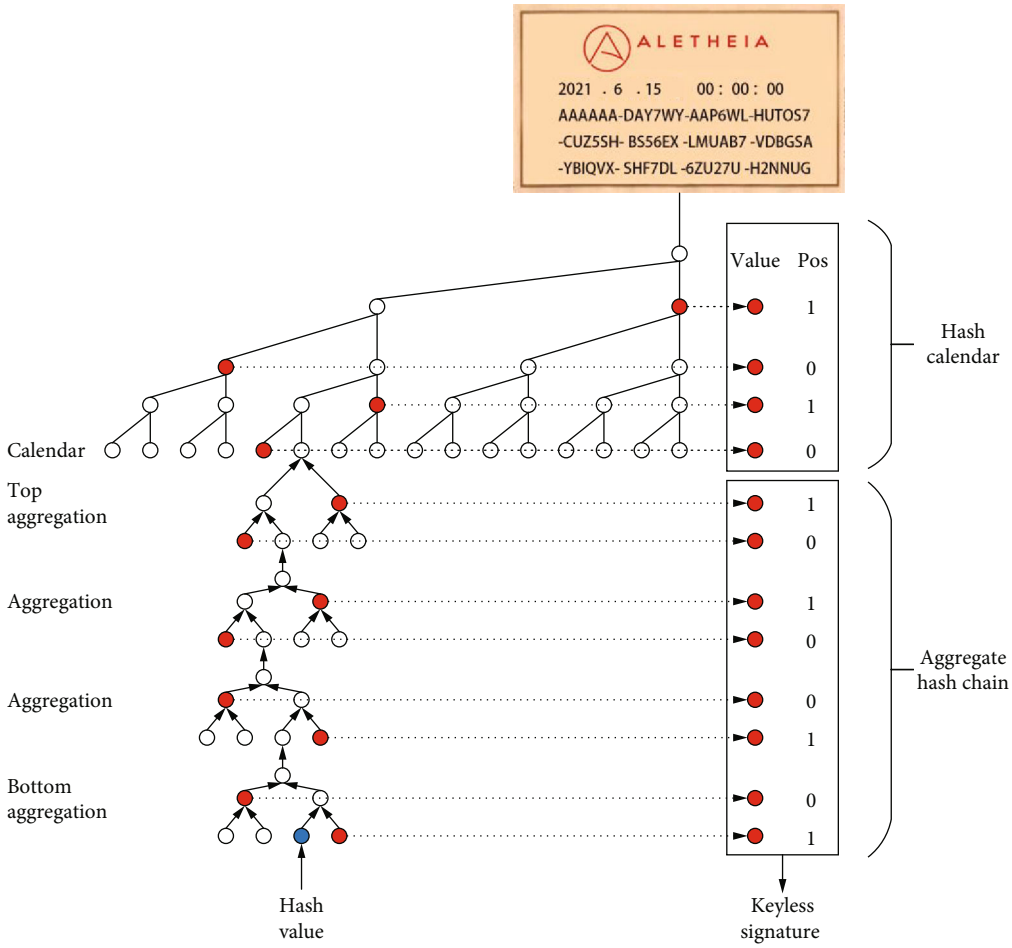


Figure 7: The keyless signature system legend.

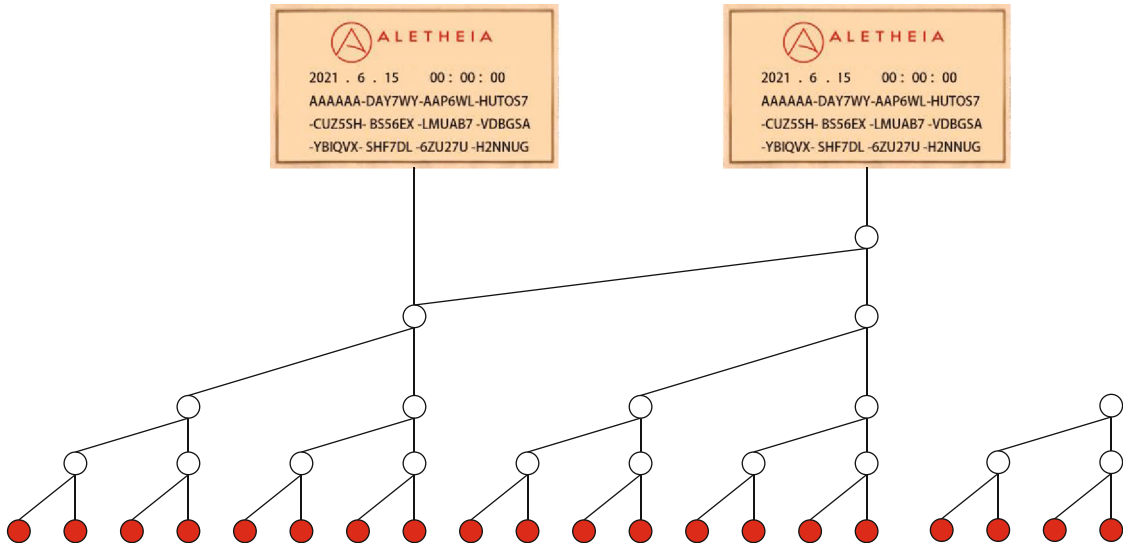


Figure 8: The Merkel forest structure.

inconsistent, it proves that the concrete production file management process is not standardized and data security risks have occurred.

In summary, the implementation of the keyless signature system could standardize the approval and writing process of files during the concrete production process, supervise

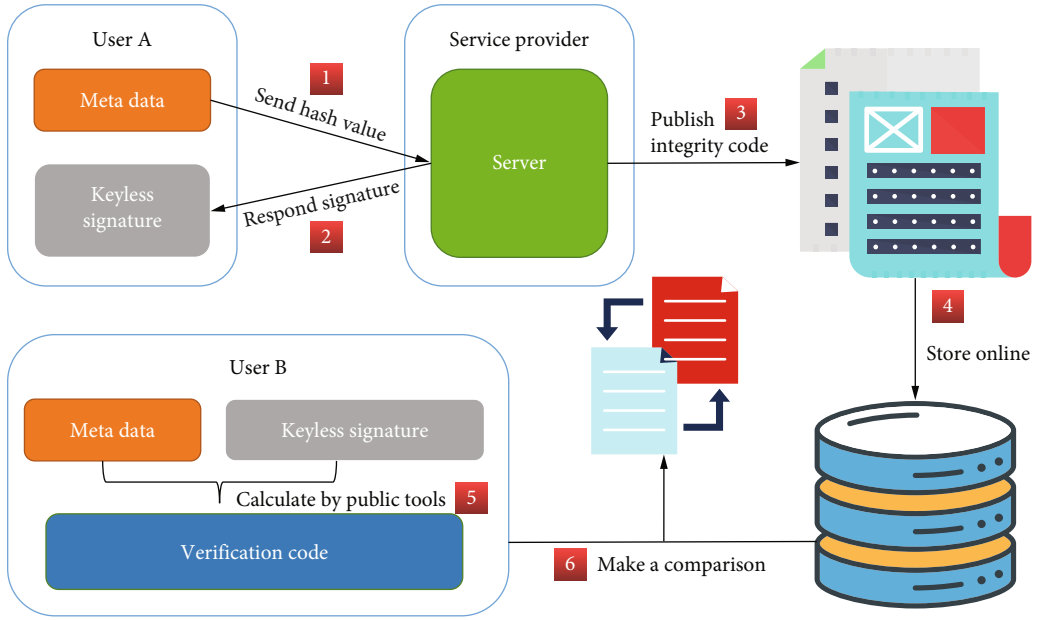


Figure 9: Data signing and verification process based on the keyless signature.

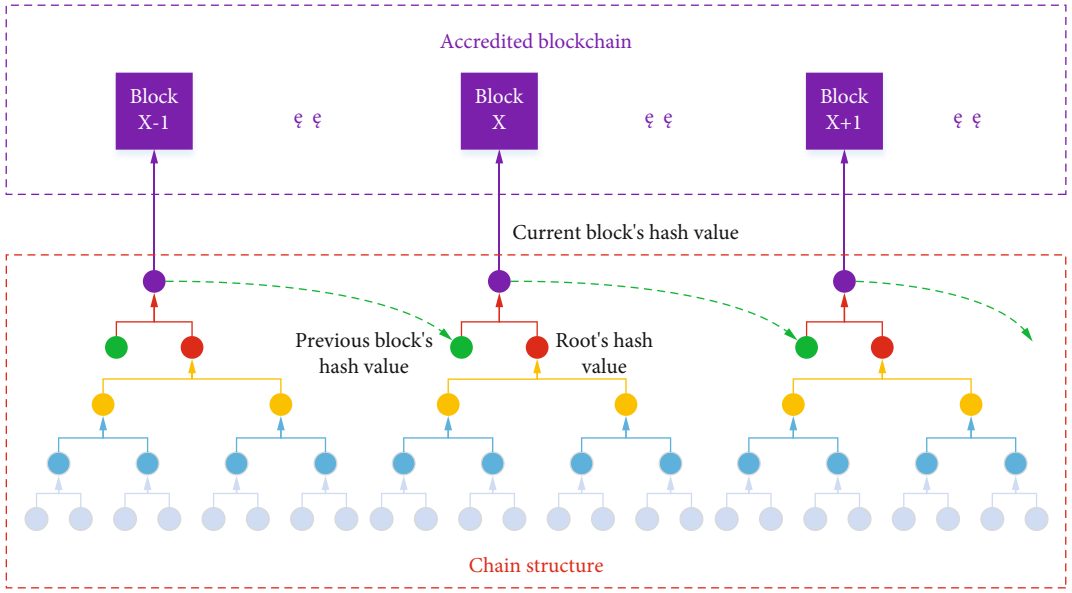


Figure 10: Schematic diagram of the chain structure.

every step of data generation, and eliminate irregular data recording and internal tampering, thus protecting the security of the concrete production file to a higher degree and for a long time.

6. Production File Management Based on Blockchains

6.1. Chain Structure Design. Based on the data structure and the keyless signature system, the chain structure of the concrete production file and the corresponding data on-chaining process are in Figure 10. When all the files involved in each concrete warehouse are collected, each file's hash value, also known as the unit block in the distributed blockchain data structure, is calculated separately. Then, a series of unit blocks aggregate two by two to form a binary tree, the root of which is called a procedure block. Thirdly, many procedure blocks polymerize to a compound as a Merkle tree, and the root is regarded as the warehouse block. Finally, by aggregating the current warehouse block with the previous warehouse block into a Merkle tree and storing the root of the tree on a trusted blockchain, the information security of the adjacent two warehouse blocks can be ensured.

As Figure 10 illustrates, compared with the traditional paper form files, the electronic files are more conducive to data search and analysis. Besides, electronic information

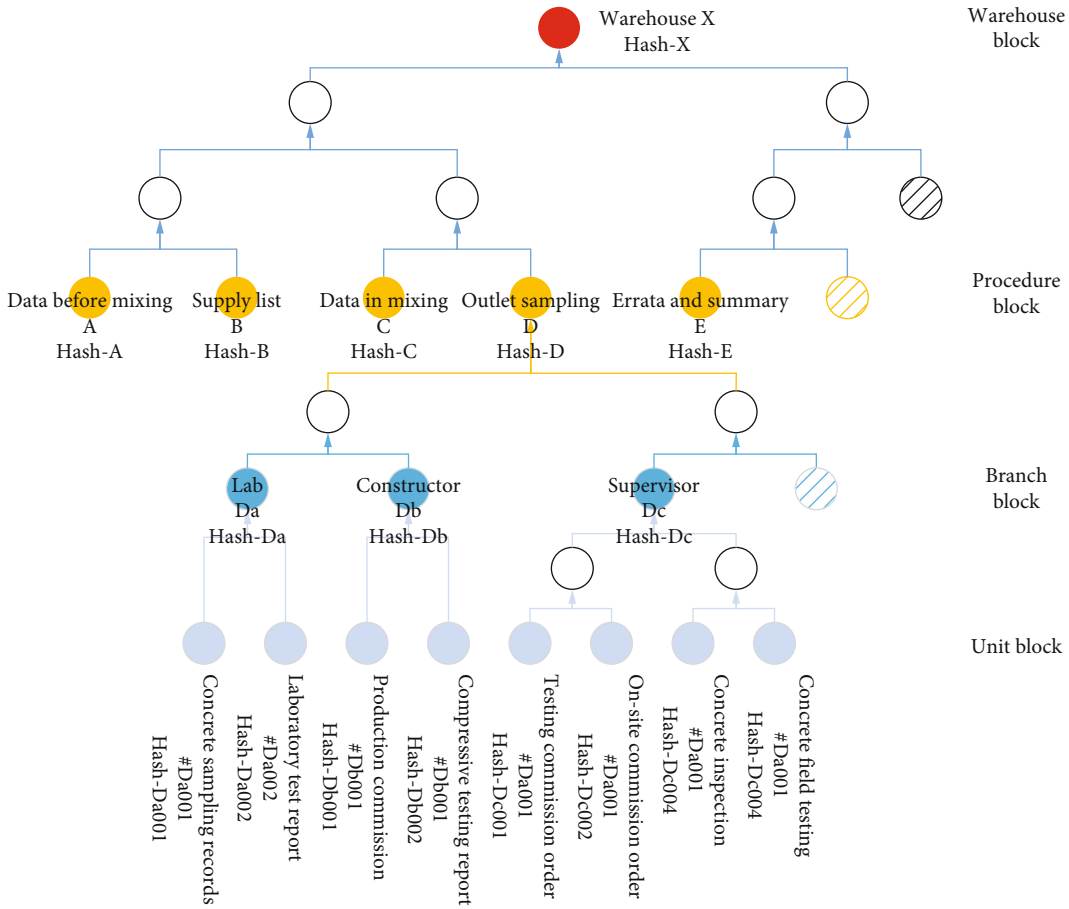


Figure 11: The organization of the warehouse block.

security and traceability can be improved markedly by the blockchain-based information deposition mechanism compared with the conventional centralized storage database.

6.2. Automatic Data On-Chaining Mechanism. Creating one warehouse block and uploading it onto blockchain means that the mixing plant finishes an entire concrete production task from batching, mixing to the end according to the instructions. A warehouse usually produces tens to hundreds of cubic meters of concrete. In the proposed model, the warehouse blocks are at the top level, and adjacent warehouse blocks are linked in tandem with time stamps. The organization of each warehouse is in Figure 11.

As shown in Figure 11, the blocks of each warehouse employ the Merkle tree structure to organize data, which is compatible with the signature generation mechanism in the keyless signature system. In Merkle trees, the two leaf nodes on each set of forks represent two files, and the files are paired two-by-two in the order of their generation time. In Figure 11, the file hash is regarded as a unit hash, and the branch hash is generated by two-by-two aggregation of all unit hashes. Furtherly, the procedure hash is composed of a two-by-two accumulation of branch hashes, and the warehouse hash is made up of pair-wise procedure hashes. Finally, the automatic data uploading is finished when all unit hashes are chained to form a warehouse hash.

Due to the structural characteristics of the Merkle tree, any changes in the underlying data will lead to changes in its parent nodes and eventually affect the changes in the Merkle root. So the Merkle tree has the advantages of efficient comparison of a large amount of data, fast location of modified data, and fast verification of incorrect data, which are all demonstrated explicitly in the proposed management model. For illustration, when two Merkle tree roots are the same, the data they represent must be the same, which makes data verification between different users possible. Besides, when the underlying data is changed, its location can be quickly detected by inspecting the corresponding branch. With this feature, the proposed model can easily fulfill fast querying of the information about the abnormal data. Last but not least, when it is necessary to prove the originality and authenticity of the data, only the hash summary of the data needs to be validated without knowing the exact content of the data.

6.3. Smart Contracts and Consensus Algorithm. The smart contracts in our blockchain management model are fulfilled by introducing various forms of notification measures such as emails and cell phone applets to inform users of pending matters and remind them of the approval delays during file flow. Beyond that, to ensure data consistency during the automatic data on-chaining process, our model adopts

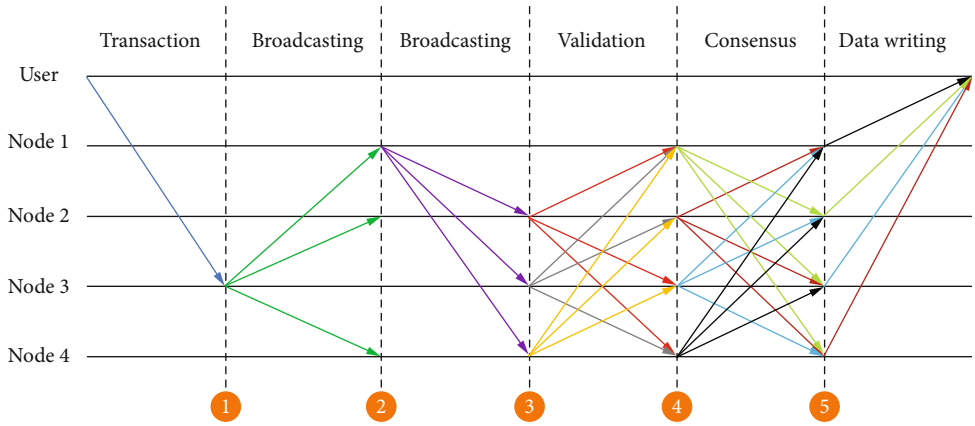


Figure 12: The implementation model of the consensus mechanism.

Byzantine Fault Tolerance (BFT) [29] as the consensus algorithm to synchronize the data to be recorded. The automatic concrete production data on-chaining mechanism based on the BFT consensus algorithm is shown in Figure 12, and it can be precisely divided into five steps, which are as follows:

(1) When the supervisor starts the approval of the concrete mixing order, the action will be considered a transaction, and the proposed model broadcasts this transaction to all blockchain nodes, including raw material providers and construction units

(2) After hash computation, the supervisor broadcasts the hash value of the transaction to all blockchain nodes

(3) Each blockchain node (participating construction unit) makes a hash after receiving the transaction and compares it with the supervisor's hash sequence

(4) After all nodes receive the message that more than half of the comparisons are approved, the transaction is deemed to be established

(5) The transaction is recorded into the block

6.4. Validation and Abnormal Block Tracking. The validation and tracking of the production files can also be fulfilled by the blockchain. As Figure 13 shows, the process is to verify the file's integrity and record the location of the files that failed the verification. Specifically, the information of the abnormal file is retrieved from the database; then, the values of the file and the corresponding warehouse are recalculated according to the calculation rules at the time of uploading. Subsequently, the newly calculated warehouse values are compared with the corresponding uploaded warehouse values on the blockchain in sequence according to the warehouse organization order.

Suppose the comparison of the hash values is consistent. In that case, all the electronic files in the warehouse are safe and secure. It has not been tampered with, so it is unnecessary to continue comparing the detailed information of this warehouse. But if inconsistency happens, the files contained in that warehouse are lost or tampered with, so it is necessary to continue to compare the hash value of each file in that warehouse. The processes of file hash matching and warehouse hash matching are the same; the newly calculated file hash is compared with the file hash recorded on the blockchain. The file that contains inconsistent hash comparison results is recorded. Thus, the traceability of the problematic blocks can be achieved.

7. Model Application

7.1. Overall Architecture. In this paper, a concrete production management system based on the proposed model has been developed and implemented in the Hanjiang to Weihe River Project in Shaanxi Province to verify the model's practicality and security. The system adopts a B-S architecture, and all users can log in and use it directly through a browser. Figure 14 shows the overall architecture.

The concrete production information management system mainly manages data related to concrete production in the water conservancy project construction, including standardized management of file filling, unified management of data archiving, and automatic uploading of production files. The management system consists of user management, menu management, process management, parameter management, authority management, and log management. Through the network interface provided by the management system, different construction units in the concrete production system automatically import or manually enter various information about concrete production and create electronic files. After that, the file, branch, procedure, and warehouse hash are generated sequentially, and then, they are organized to the tree structure according to the distributed blockchain data structure.

The generated hash values are uploaded to a credible blockchain for deposition. The information interaction between the blockchain and the information management platforms is fulfilled through port calls. In our information management system, the blockchain is the consortium blockchain called the Blockchain-based Service Network. This blockchain was jointly initiated by the State Information Center, China Mobile Communications Corporation, China UnionPay Corporation, and Beijing Red Date

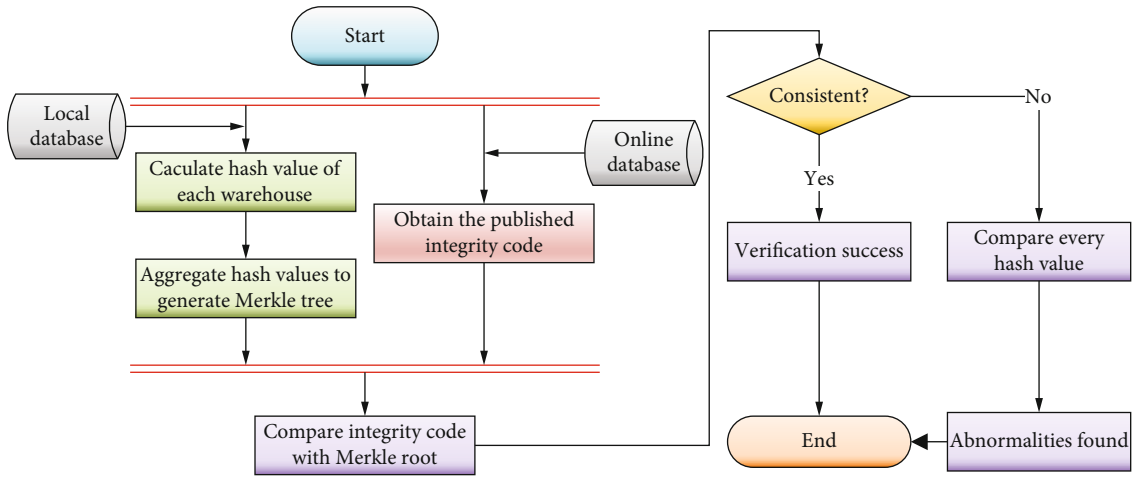


Figure 13: Schematic diagram of the abnormal block tracking flow.

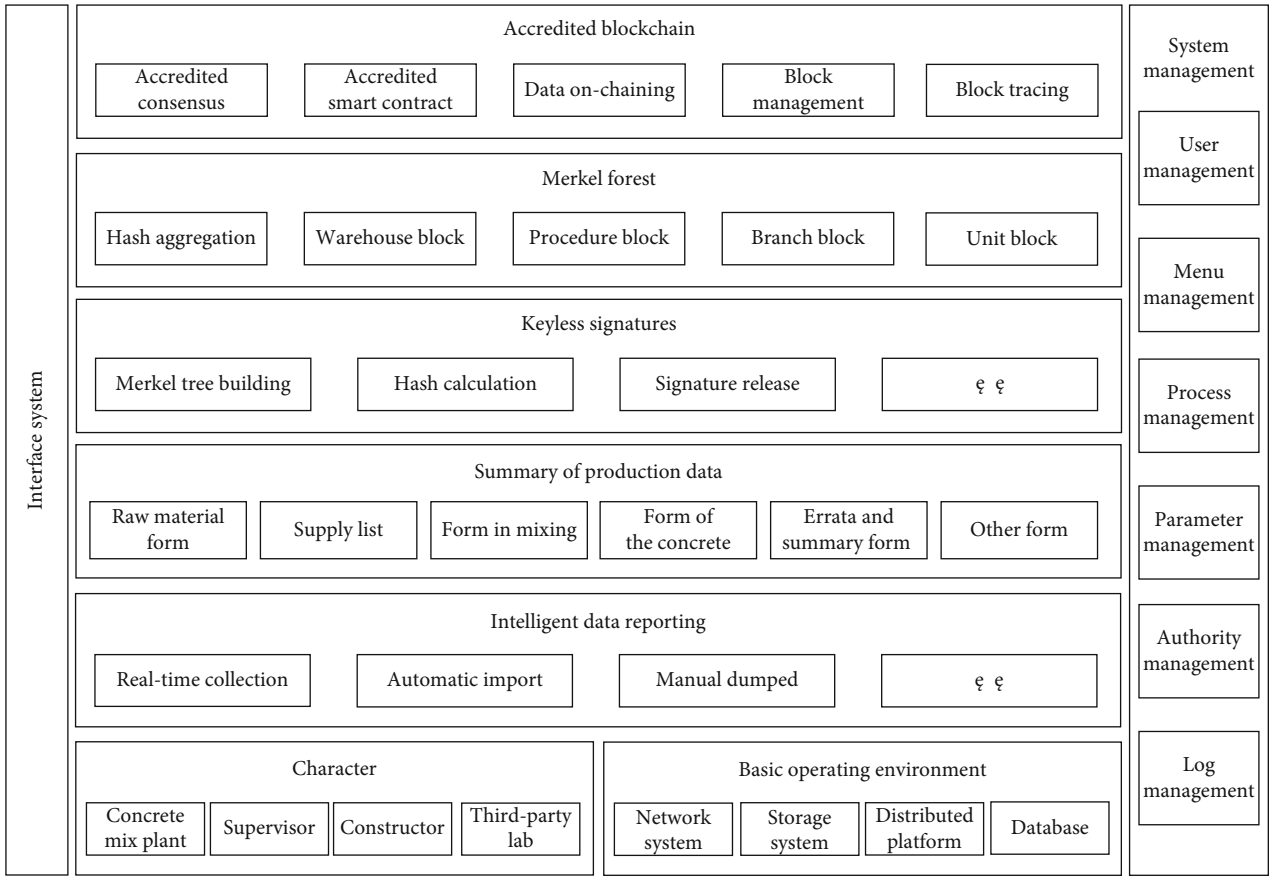


Figure 14: Architecture of concrete production information management system.

Technology Corporation [30]. Besides, this consortium blockchain provides the storage, verification, and traceability of hash values and facilitates historical data security verification.

In practice, the system was implemented in the Hanjiang to Weihe River Project to collect and organize the concrete production-related files in 2020. The total concrete production volume in the project in 2020 was about 170,000 square meters, which generated about 16,000 related paper forms in total. At present, we have entered and uploaded some of the files, including 18,000 square meters of concrete related to more than 3,500 forms, and stored these records on the consortium blockchain. The data server and the application server configurations in our system are the same: both are

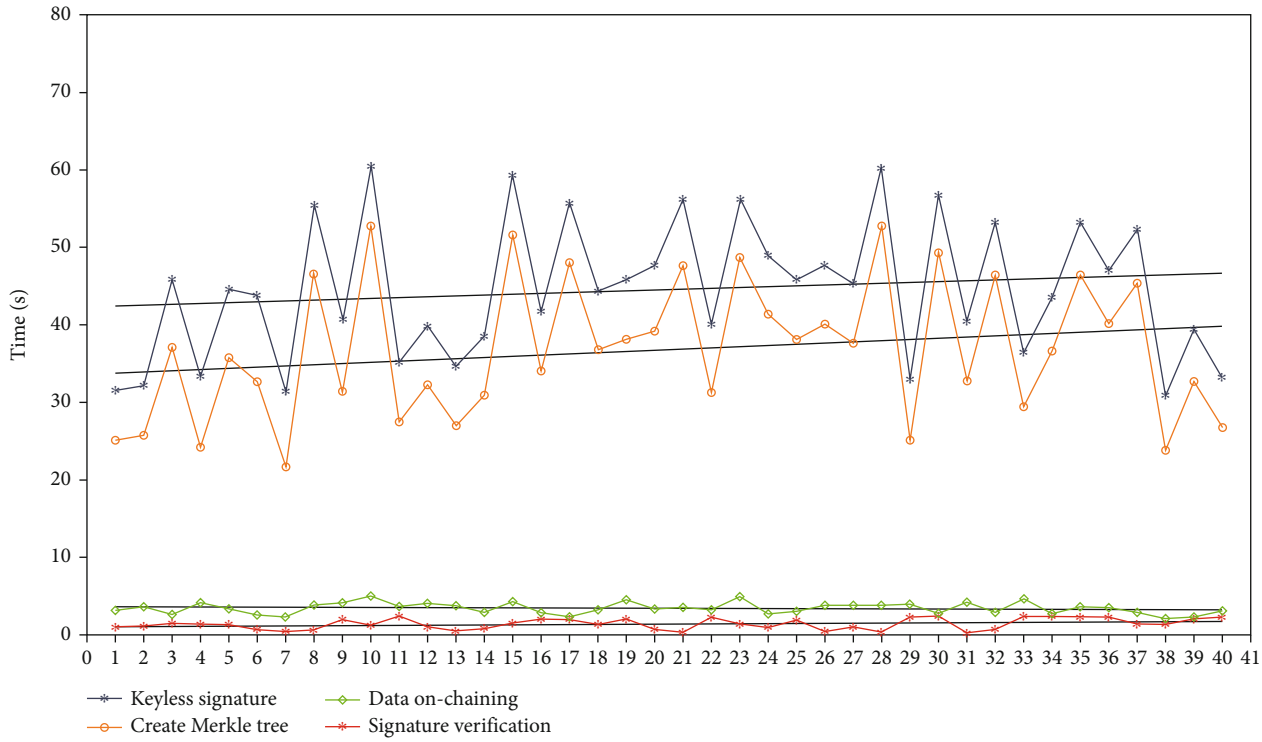


Figure 15: Comparison of system time consumption by phase.

dual-core and quad-threaded, with 8 G memory and 500 G storage space, which meet the minimum requirements for civil engineers [31].

7.2. Experiment and Analysis. The experiment for testing the system performance is designed as follows: one warehouse block is selected randomly as the experiment object from the actual concrete production process. The target warehouse block contains 40 files, including supply contact sheets, production notification sheets, quality inspection sheets, and errata summary sheets, recording a complete concrete production process. The file creation and transmission are standardized with the keyless signature. After the files are filled and verified, they are uploaded on the blockchain for permanent storage. Besides, the time for the Merkle tree construction, the keyless signature creation, the signature verification, and files' chaining are recorded separately. The specific time spent on the four steps of the 40 concrete production files is shown in Figure 15.

As shown in Figure 15, the overall trend of keyless signature generation time per file raises as the production data increases. By fitting the linear regression model, it can be seen that the slope of the total keyless signature time is about 0.109. For each integrated keyless signature registration, when the size of the Merkle tree increases, the keyless signature generation time of the following file will also increase by about 0.109 s.

Secondly, the average time consumption for data on-chaining is about 3.39 s, with a slope of -0.009. This erratic fluctuation is caused by blockchain instability and network fluctuations.

Thirdly, the time for the Merkle tree generation also shows an increasing trend correlation with the keyless signature generation time because the keyless signature is based on the combination of hash values from the root to the leaf sequence of the Merkle tree and its corresponding sequence coordinates. The slopes of Merkle tree creation and keyless signature generation are similar by linear fitting, which indicates that the creation time of the Merkle tree is the main factor that increases the generation time of keyless signature.

Fourthly, the average time to verify the on-chain data is about 1.38 s. The slope of the linear fit function is 0.019, indicating that the verification is swift for on-chain data. The verification efficiency is mainly affected by the structure of the warehouse block.

Besides, we also conduct the tests on keyless signature sizes. As shown in Figure 16, the keyless signature size of each file is about 157 kb, and its storage cost is less than 1 penny. The keyless signature storage cost of the whole warehouse is less than 0.1 yuan, and this cost is almost negligible compared with the benefits of data security.

Finally, we use the number of transactions processed per second as the criterion for system throughput to evaluate the entire performance. The throughput of the relevant smart contracts is calculated for different concurrent requests. The number of concurrent requests is set from 100 to 1000, and 10 experiments are conducted in sequence. At last, the average values are taken as the experimental results. The throughput of the smart contracts is in Figure 17.

In Figure 17, the throughput of the write operation (data on-chaining) is overall lower than that of the read operation (signature verification). In other words, the write operation

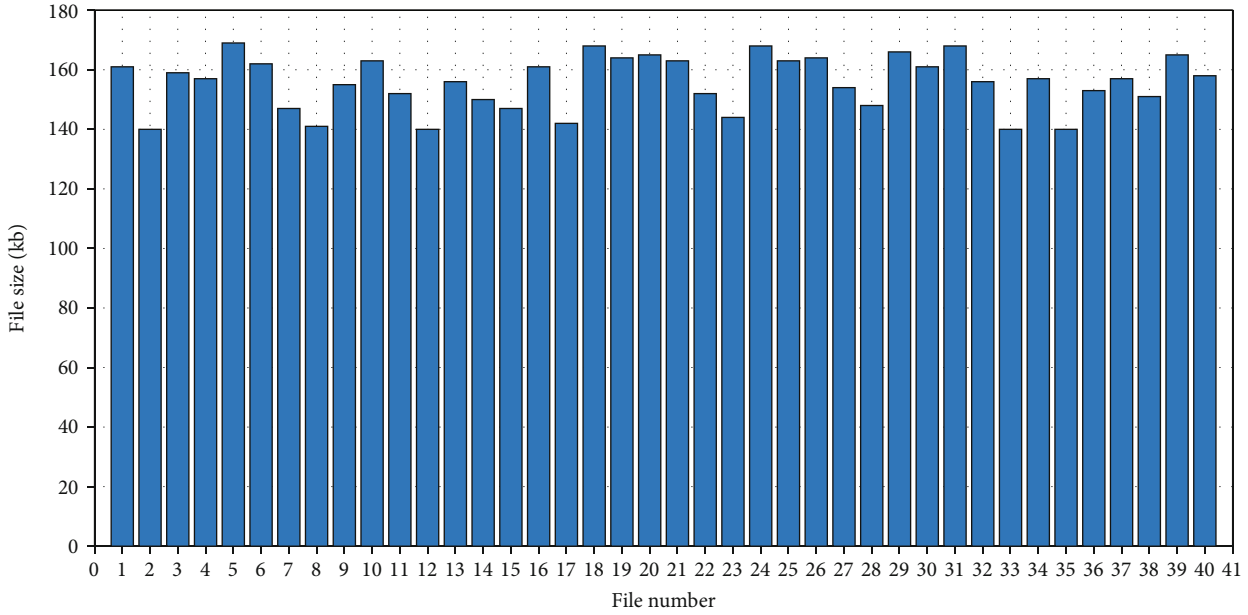


Figure 16: Storage space consumed by keyless signatures.

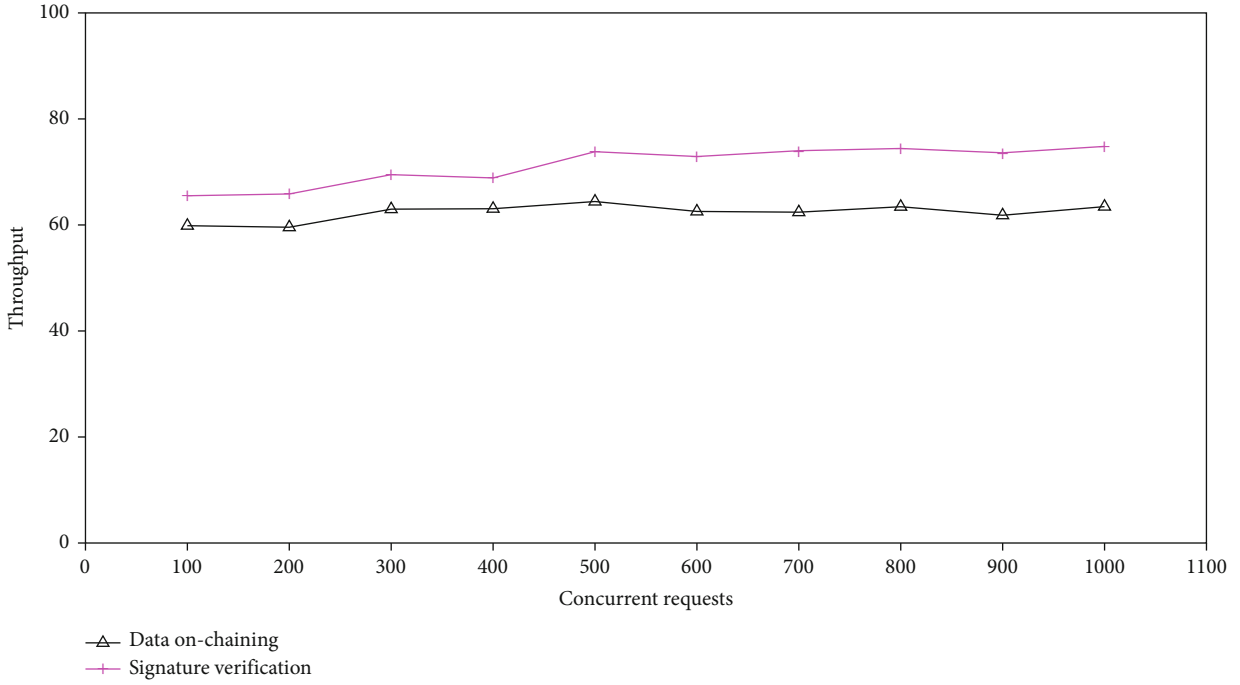


Figure 17: Throughput of the smart contract with the concurrent requests.

tends to be more time-consuming than the read operation. It is because the write operation needs to hash the data and generate a historical version of the old data to ensure the traceability of the blockchain. On the other hand, read operation only needs to search and validate data based on index positions, thus taking less time. Besides, the system's throughput increases with the number of concurrent requests. However, when the number of simultaneous requests reaches a certain value, the growth trend slows down slightly. By calculation, the throughput stays 71 for the smart contract with data on-chaining. In contrast, the throughput for a signature verification smart contract is about 62.

8. Conclusions

The digital archiving of engineering construction files in intelligent water projects is of great significance. The blockchain can provide security verification and integrity check for electronic files, which guarantees the security of archive

informatization and contributes to the realization of electronic archiving of files. This paper proposes a comprehensive data management model for smart water system construction based on blockchain and edge intelligence and then implements it in the Hanjiang to Weihe River Project in Shaanxi Province. Firstly, the behavioral model for the concrete production process is summarized, and the corresponding roles that participate in the process are abstracted out simultaneously. Secondly, the intelligent keyless signature based on parallel edge computing is introduced to ensure data security. The proposed model uses the Merkle tree to construct a chained file structure and standardizes the data entering, uploading, and checking procedure by the consensus mechanism. In the case study, we have created a blockchain of 3,500 blocks according to the decentralization requirement. In total, the proposed model and the corresponding system have already taken a big step forward in saving workforce and material resources and improving the security and traceability of construction archives markedly. We believe that through a more extensive scope of application and continuous improvement, the management of archives in civil engineering, especially in smart water projects, will eventually achieve the goal of digitalization.

Data Availability

The experiment data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

All authors declare no conflict of interest in this paper.

Acknowledgments

This research work is supported by the National Natural Science Funds of China (62072368), Basic Research in Natural Science and Enterprise Joint Fund of Shaanxi (2021JLM-58), and Special Scientific Research Project of Education Department of Shaanxi (21JK0781).

References

[1] N. Nizamuddin, K. Salah, and M. A. Azad, "Decentralized document version control using ethereum blockchain and IPFS," *Computers & Electrical Engineering*, vol. 76, pp. 183–197, 2019.

[2] C. Feng, B. Liu, Z. Guo, K. Yu, Z. Qin, and K. K. R. Choo, "Blockchain-based cross-domain authentication for intelligent 5G-enabled Internet of drones," *IEEE Internet of Things Journal*, 2021.

[3] W. Wang, H. Xu, M. Alazab, T. R. Gadekallu, Z. Han, and C. Su, "Blockchain-based reliable and efficient certificateless signature for IIoT devices," *IEEE Transactions on Industrial Informatics*, p. 1, 2021.

[4] Y. Zhang, W. Luo, and F. Yu, "Construction of chinese smart water conservancy platform based on the blockchain: technology integration and innovation application," *Sustainability*, vol. 12, no. 20, p. 8306, 2020.

[5] J. Song, Z. Han, W. Wang, J. Chen, and Y. Liu, "A new secure arrangement for privacy-preserving data collection," *Computer Standards & Interfaces*, vol. 80, article 103582, 2022.

[6] C. Feng, B. Liu, and K. Yu, "Blockchain-empowered decentralized horizontal federated learning for 5G-enabled UAVs," *IEEE Transactions on Industrial Informatics*, 2022.

[7] T. Bui, D. Cooper, and J. Collomosse, "Tamper-proofing video with hierarchical attention autoencoder hashing on blockchain," *IEEE Transactions on Multimedia*, vol. 22, no. 11, pp. 2858–2872, 2020.

[8] B. Zhong, H. Wu, and L. Ding, "Hyperledger fabric-based consortium blockchain for construction quality information management," *Frontiers of Engineering Management*, vol. 7, no. 4, pp. 512–527, 2020.

[9] L. Zhang, M. Peng, and W. Wang, "Secure and efficient data storage and sharing scheme for blockchain-based mobile-edge computing," *Transactions on Emerging Telecommunications Technologies*, no. article e4315, 2021.

[10] G. Nagasubramanian, R. K. Sakthivel, and R. Patan, "Securing e-health records using keyless signature infrastructure blockchain technology in the cloud," *Neural Computing and Applications*, vol. 32, no. 3, pp. 639–647, 2020.

[11] J. Zhang, Y. Huang, and Y. Wang, "Multi-objective optimization of concrete mixture proportions using machine learning and metaheuristic algorithms," *Construction and Building Materials*, vol. 253, article 119208, 2020.

[12] Y. Gong, L. Zhang, and R. Liu, "Nonlinear MIMO for industrial Internet of Things in cyber-physical systems," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 8, pp. 5533–5541, 2021.

[13] C. Feng, K. Yu, M. Aloqaily, M. Alazab, Z. Lv, and S. Mumtaz, "Attribute-based encryption with parallel outsourced decryption for edge intelligent IoV," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 11, pp. 13784–13795, 2020.

[14] K. A. Nguyen, R. A. Stewart, H. Zhang, O. Sahin, and N. Siriwardene, "Re-engineering traditional urban water management practices with smart metering and informatics," *Environmental Modelling & Software*, vol. 101, pp. 256–267, 2018.

[15] J. Feng, L. Liu, and Q. Pei, "Min-Max cost optimization for efficient hierarchical federated learning in wireless edge networks," *IEEE Transactions on Parallel and Distributed Systems*, 2022.

[16] H. Li, K. Yu, B. Liu, C. Feng, Z. Qin, and G. Srivastava, "An efficient ciphertext-policy weighted attribute-based encryption for the Internet of health things," *IEEE Journal of Biomedical and Health Informatics*, vol. PP, 2021.

[17] K. Yu, M. Arifuzzaman, and Z. Wen, "A key management scheme for secure communications of information centric advanced metering infrastructure in smart grid," *IEEE transactions on instrumentation and measurement*, vol. 64, no. 8, pp. 2072–2085, 2015.

[18] L. Tan, K. Yu, and N. Shi, "Towards secure and privacy-preserving data sharing for covid-19 medical records: a blockchain-empowered approach," *IEEE Transactions on Network Science and Engineering*, 2022.

[19] L. Zhao, J. Li, and A. Al-Dubai, "Routing schemes in software-defined vehicular networks: design, open issues and challenges," *IEEE Intelligent Transportation Systems Magazine*, vol. 13, no. 4, pp. 217–226, 2020.

[20] J. Feng, F. R. Yu, and Q. Pei, "Cooperative computation offloading and resource allocation for blockchain-enabled mobile-edge computing: a deep reinforcement learning approach," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6214–6228, 2019.

[21] L. Tan, K. Yu, F. Ming, X. Chen, and G. Srivastava, "Secure and resilient artificial intelligence of things: a HoneyNet approach for threat detection and situational awareness," *IEEE Consumer Electronics Magazine*, p. 1, 2021.

[22] K. Yu, L. Tan, and L. Lin, "Deep-learning-empowered breast cancer auxiliary diagnosis for 5GB remote E-health," *IEEE Wireless Communications*, vol. 28, no. 3, pp. 54–61, 2021.

[23] L. Liu, C. Chen, and Q. Pei, "Vehicular edge computing and networking: a survey," *Mobile Networks and Applications*, vol. 26, no. 3, pp. 1145–1168, 2021.

[24] S. Hakak, W. Z. Khan, and G. A. Gilkar, "Securing smart cities through blockchain technology: architecture, requirements, and challenges," *IEEE Network*, vol. 34, no. 1, pp. 8–14, 2020.

[25] K. Yu, Z. Guo, and Y. Shen, "Secure artificial intelligence of things for implicit group recommendations," *IEEE Internet of Things Journal*, 2022.

[26] T. Alladi, V. Chamola, and R. M. Parizi, "Blockchain applications for industry 4.0 and industrial IoT: a review," *IEEE Access*, vol. 7, pp. 176935–176951, 2019.

[27] H. Huang, J. Lin, and B. Zheng, "When blockchain meets distributed file systems: an overview, challenges, and open issues," *IEEE Access*, vol. 8, pp. 50574–50586, 2020.

[28] L. Liu, J. Feng, and Q. Pei, "Blockchain-enabled secure data sharing scheme in mobile-edge computing: an asynchronous advantage actor–critic learning approach," *IEEE Internet of Things Journal*, vol. 8, no. 4, pp. 2342–2353, 2020.

[29] H. Xiong, C. Jin, M. Alazab et al., "On the design of blockchain-based ECDSA with fault-tolerant batch verication protocol for blockchain-enabled IoMT," *IEEE Journal of Biomedical and Health Informatics*, vol. PP, p. 1, 2021.

[30] J. Ma, S. Zhang, and H. Li, "Sparse Bayesian learning for the time-varying massive MIMO channels: acquisition and tracking," *IEEE Transactions on Communications*, vol. 67, no. 3, pp. 1925–1938, 2018.

[31] L. Liu, M. Zhao, M. Yu, M. A. Jan, D. Lan, and A. Taherkordi, "Mobility-aware multi-hop task offloading for autonomous driving in vehicular edge computing and networks," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–14, 2022.

Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 5730196, 8 pages
https://doi.org/10.1155/2022/5730196

Research Article

HeteroFL Blockchain Approach-Based Security for Cognitive Internet of Things

Shivani Wadhwa,[1] Shalli Rani,[1] Gagandeep Kaur,[2] Deepika Koundal,[3] Atef Zaguia,[4] and Wegayehu Enbeyle[5]

[1]*Chitkara University Institute of Engineering and Technology, Rajpura 140401, India*
[2]*Department of Computer Science, Punjabi University, Patiala 147001, India*
[3]*Department of Systemics, School of Computer Science, University of Petroleum and Energy Studies, Dehradun, India*
[4]*Department of Computer Science, College of Computers and Information Technology, Taif University, P.O. BOX 11099, Taif 21944, Saudi Arabia*
[5]*Department of Statistics, Mizan-Tepi University, Tepi, Ethiopia*

Correspondence should be addressed to Wegayehu Enbeyle; wegu0202@gmail.com

Received 21 October 2021; Revised 22 December 2021; Accepted 12 January 2022; Published 7 March 2022

Academic Editor: Celimuge Wu

Cognitive learning is progressively prospering in the field of Internet of Things (IoT). With the advancement in IoT, data generation rate has also increased, whereas issues like performance, attacks on the data, security of the data, and inadequate data resources are yet to be resolved. Recent studies are mostly focusing on the security of the data which can be handled by blockchain. Blockchain technology records the learned data into the block which is generated after completing proper consensus mechanism. In this paper, Hetero Federated Learning approach is used to apply cognitive learning on data produced by Internet of Thing devices. Security on cognitiveIoT data is provided by blockchain using Proof of Work consensus mechanism. By applying blockchain over heteroFL approach, we have conducted various simulations to check the performance of our proposed framework. Parameters taken into consideration during performance evaluation are effect of number of blocks on memory utilization and impact of data sample size on accuracy according to different learning rates.

1. Introduction

Previous Google chief director, Eric Schmidt made this striking IoT forecast: "The Internet will vanish. There will be numerous IP addresses, such countless gadgets, sensors, things that you are wearing, things that you are cooperating with, that you will not detect it. It will be important for your essence constantly." In-numerable efforts have been done from academia community, network providers, service providers, and various standard developing organizations, to provoke the growth of IoT devices [1]. It is expected that greater than 64 B devices based on IoT will exist worldwide by 2025. Most focused areas of research include networking, security, computations, communication, and energy harvesting but without cognitive ability, i.e., without brain, IoT seems awkward [2]. Entitling high level intelligence into the IoT gives rise to Cognitive Internet of Things [3].

Cognitive Internet of Things (CIoT) is a new paradigm, where physical or virtual things are connected with least human interruption. The communication with the things occurs by utilizing the approach of understanding. Understanding can be done from the actual climate, sensed data, and social communities. They store the gained connotation and additional data gathered in form of data sets and adjust to changes by means of computation and resource efficient algorithms for decision making. CIoT assists in bringing together physical world with the social world in an intelligent manner. It includes smart learning, smart resource allocation, spectrum sensing, capturing high precision data, smart service provisioning, and information

processing. Figure 1 illustrates the smart features that can be incorporated in the CIoT.

Smart data being produced by smart devices may suffer from security attacks like denial of service (DoS), physical attacks, malware, and malicious data injection [19]. Traditional machine learning approaches may not provide prevention from such attacks. Federated learning is a recent approach which learns from the dispersed data by incorporating collaborative models that are embedded in the local nodes. This approach learns iteratively till it reaches the threshold value set by the global model. Hetero Federated learning (HeteroFL) models are designed for devices that need different computation requirements and communication abilities.

With the rapid development of new technologies, the data generation rate has also increased. As the number of devices is increasing, the data production rate will also increase. It is of utmost importance to provide security to the users who are relying on the data produced by IoT devices. Lot of work is done in the field of research of IoT security [4]. However, few areas of research in IoT security are still unexplored. Meanwhile, as an arising innovation, blockchain innovation steadily stirs consideration of the scholarly community and industry. Blockchain innovation depends on a decentralized shared organization, based on cryptography, time-managed information of all events, well-defined consensus mechanisms, and with proper traceability and check of information to be stored.

Cognitive computing is assisting a lot in making IoT become smarter by providing human intelligence to the systems. However, privacy leakage of heterogeneous clients of IoT is not addressed by most of the technologies developed so far. In our proposed framework, cognitive computing is done by using heterogeneous federated learning to serve the needs of heterogeneous IoT clients. However, poisonous attacks can also be done on federated learning which will degrade the performance of the system [5]. Hence, integration of blockchain is done to protect the system from attacks and make the system more secure.

The structure of this paper is organized as follows. In section 2, related work is discussed. Then, the proposed cognitive learning through hetero federated learning and privacy through blockchain is presented in section 3. Section 4 discusses the performance evaluation parameters. Conclusion and future work are mentioned in section 5.

2. Related Work

In last few years, cognitive computing in Internet of Things has gained momentum in different ways. Various technologies collaborated with this are federated learning and blockchain. To guarantee the intelligent sensing of data, the Quality of Information Coverage (QIC) fulfillment metric is utilized to decide how gathered information tests can fulfill CIoT necessities. Experiments conducted in this model proved the accuracy of the QIC algorithm [7]. Hierarchical architecture is proposed for the heterogeneous IoT system based on blockchain [8]. Content caching

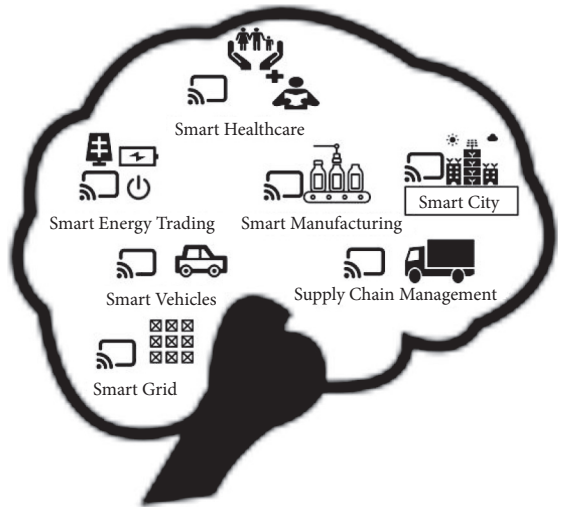


Figure 1: Cognitive internet of things.

architecture is proposed for the interaction between different vehicles and Road Side Units. Blockchain technology is also adopted to provide trust among the users which are connected to each other [9]. Edge networks are used to facilitate blockchain in vehicular decentralized environments. The selected edge nodes perform the task of maintaining blockchain. Selection of edge nodes is done by considering the velocity, distribution, and link of the vehicles. The proposed method provides improved performance in block dissemination for the implementation of blockchain in vehicular decentralized environments [10]. Authors highlighted that future research area will be to improve protocols of vehicular IoT which can support blockchain and also to frame efficient blockchain which can satisfy the essential requirements for the vehicular IoT [11]. Multichannel blockchain architecture is proposed for Internet of Vehicles where optimization of all channels is done by using vehicle density as well as based on requirements of applications. The proposed method improves the latency, transaction success ratio, and through put for varying number of vehicles [12]. The decentralized model for huge data based on cognitive processing, federated learning, and blockchain together is fostered. Blockchain empowered federated learning assists fast assembly with high-level verifications and selection of members [13]. To maximize the throughput, transmission scheduling for CIoT based on Q-learning approach is proposed. A Markov choice interaction-based model is detailed to portray the state transformation of the framework [14]. Cognitive computing technologies with IoT provide solutions to many existing challenges like big sensory data, efficient computation at CIoT edge, and various data sources [15]. Energy and spectrum efficiency are considered as very important parameters in CIoT. Metric is identified which provides the characteristics of network design space [18]. To increase the network utilization and throughput, the hybrid model is proposed for energy constraint devices and data aggregation of IoT devices. Deep Reinforcement Learning is applied with the

Double Q-learning algorithm to provide optimization using multiobjective ant colony optimization (MOACO) and greedy method techniques [19]. Internet of Multimedia Things based services are provided by the proposed algorithm, i.e., cognitive-based middleware for private data mashup (CMPM). Privacy issues are taken into consideration to provide proper environmental monitoring of data [20]. Table 1 presents the work done by different researchers in the field of cognitive IoT.

Although many studies are focusing on the integration of blockchain in IoT, blockchain in cognitive learning, and cognitive learning in IoT, there arises a need to solve the problem of heterogeneous IoT clients by incorporating cognitive learning and blockchain which can provide improvement in accuracy, learning rate, and latency.

Due to limited research carried out for security of IoT data produced by heterogeneous clients using HeteroFL, the research in this area is in its infancy. Therefore, this paper focusses on providing security to heteroFL-based cognitive learned data using blockchain.

3. Blockchain-Based Privacy on Cognitive Learned Data

Different applications of IoT involve variety of clients because of connectivity of heterogeneous devices. As different clients possess varying computation and capabilities of communicating with each other, they are assigned different complexity levels. First, the learning takes place on their respective local model and then the aggregation of all parameters of local models gives rise to parameter of the single global model. In our proposed framework, the cognitive model uses heterogeneous federated learning for training of the IoT data and then the trained data are secured by using blockchain.

3.1. Cognitive Learning Based on HeteroFL. Mobile devices, Gaming, and IoT devices generate huge amount of data. Based on cognitive computing, models can be made to store the data and then train the models locally. Federated learning (FL) is an approach of machine learning where parameters of local models are trained and their aggregation produces the global model which is independent of raw data. Generally, local models and global model share the same architecture. However, there can be various scenarios where miscellaneous types of local models will exist with the wide range of computing complexities. To meet the requirements of heterogeneous clients of IoT devices, another unified learning system named HeteroFL is used to outfit the entirely different computations and their communication abilities [6].

The process of training global model is done from local data $\{x1, \ldots, xn\}$ available at heterogeneous IoT devices. Local model parameters are expressed as $\{w1, \ldots, wn\}$. The model averaging of the local parameters is done to find the global parameter wg. This process is done in various iterations, and wg calculated at ith iteration is passed on to the local parameters of $(i+1)th$ iteration.

For effective cognitive learning to take place, size of network can be modulated by changing width of the network. This can help in decreasing the local parameters, whereas architecture of local and global parameters remains in the same model class. This also improves the stability of aggregation in the global model. In heteroFL, selection of global parameters is done based on the size of input channel (ig), output channel (og), and computation complexity level (c). Figure 2 shows the federated learning approach with various complexity levels of computations on the data produced by heterogeneous IoT devices. Calculation of shrinkage ratio plays very important role for hidden layers. Equations of shrinkage ratio of output channel are expressed as mentioned in the following equation:

$$s1 = \left(\frac{o_l^{c+1}}{og}\right)^{1/c}. \tag{1}$$

Formula for shrinkage ratio of input channel is shown in the following equation:

$$s2 = \left(\frac{i_l^{c+1}}{ig}\right)^{1/c}. \tag{2}$$

For simplification, let $s1 = s2 = s$.

Shrinkage ratio of local model parameter is mentioned in the following equation:

$$SR = w_l^c = wg * s^{2(c-1)}. \tag{3}$$

According to the calculated potential of the local model parameter, global model parameters can be constructed based on allocated subsets. The concept of set difference is mostly used in the calculations of the global parameter. Figure 3 shows different regions according to set differences. According to Figure 3, total clients $m = 6$ are shown. Here, 3 clients are of complexity level 3 (represented by m_3 in red region), 2 clients are of complexity level 2 (represented by m_2 in yellow region), and 1 client is of complexity level 1 (represented by m_1 in blue region).

Aggregation of smallest local model parameter (red region) is done as follows:

$$w_l^3 = \frac{\left(w_1^3 + w_2^3 + w_3^3\right)}{3}. \tag{4}$$

Calculation of subset of yellow region is done as follows:

$$w_1^2 - w_1^3 = \frac{1}{(m - m_3)} * \sum_{m=1}^{m-m_3} w_1^2 - w_1^3. \tag{5}$$

Calculation of subset of blue region is done as follows:

$$w_1^1 - w_1^2 = \frac{1}{(m - m_2 - m_3)} * \sum_{m=1}^{(m-m_2-m_3)} w_1^1 - w_1^2. \tag{6}$$

Calculation of subset of global model parameter is done as follows:

Table 1: Related work in CognitiveIot and Blockchain.

Ref. no	Parameters	Technology used	Application	Proposed model	Future scope
[7]	Information density, price, collection of data samples	Non-cooperative game	Intelligent sensing intelligence system	Quality of information coverage algorithm	Considering privacy for the growth of internet of things
[16]	Smart contract	Cognitive engine for machine translation, blockchain, intrusion detection	Shopping center	Cognitive recommender system	Applicability of proposed framework for web of things
[9]	Cache hit rate and robustness	Caching strategy, deep learning and machine learning algorithms	Internet of vehicles	Blockchain and cognitive-engine-enabled content caching strategy	—
[17]	Average delay, processing time	Convolutional neural networks (CNN), smart contract, machine learning algorithms	Sharing economy services in mega smart cities	MEC-based sharing service economy system, which includes the blockchain	Testing different sharing economy cases at a bigger level
[18]	System utility, no. of average packet loss	Markov decision process	Wireless data	Q-learning algorithm and stacked autoencoders deep learning model	Process to create more relays
[18]	Energy and spectrum efficiency	Dynamics of spectrum sharing and energy harvesting	Solar energy harvesting	Cloud enabled CIoT platform	—
[19]	Energy and throughput	Deep reinforcement learning and double Q-learning algorithm	—	Multiobjective ant colony optimization (MOACO)	Considering security parameters

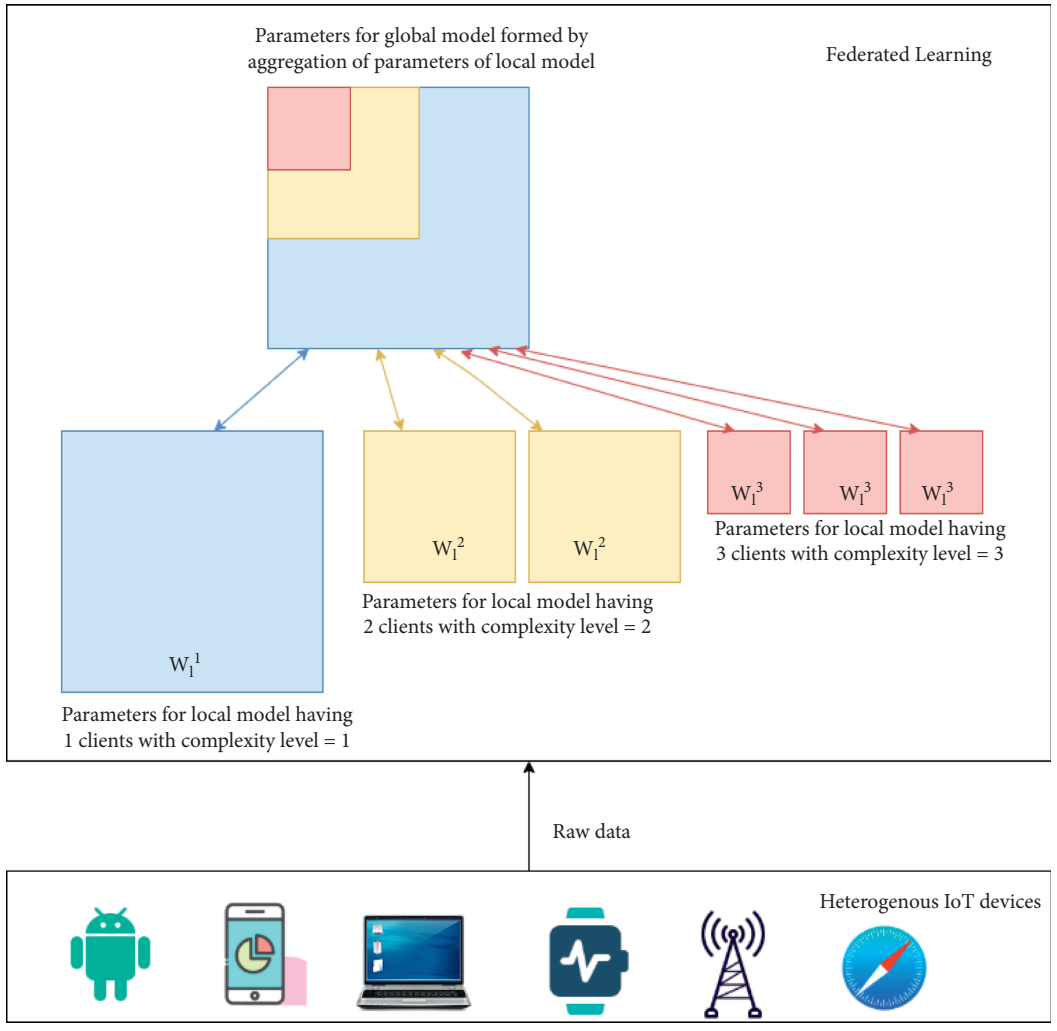


Figure 2: Federated learning from data produced by IoT devices.

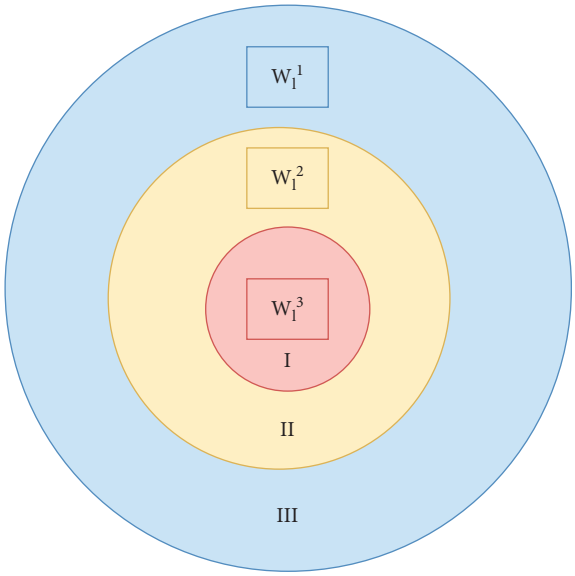


Figure 3: Venn diagram of different complexity levels.

$$w_g = w_1^3 \cup \left(w_1^2 - w_1^3\right) \cup \left(w_1^1 - w_1^2\right). \tag{7}$$

Aggregation of all those clients is done whose parameter is the part of the parameter matrix. Hence, model having intermediate complexities contains the parameters completely aggregated with the bigger models whereas moderately with the smaller models. Also, aggregation is more in case of smaller local models which can benefit global model. The data stored at the global model become the learning part of the cognitive learning. Algorithm 1 and Algorithm 2 are mainly designed for applying cognitive learning using heteroFL and sending client updates.

3.2. Blockchain on Cognitive Learned Data. Blockchain assumes a significant part to protect the performance of cognitive learned data. As these data will be used for making further decisions, it is very important to provide privacy to these data. Important characteristics of blockchain-like immutability, tamper-resistant, decentralized, pseudonymous identity, and so on are the contributing factors of security and privacy.

This subsection explains the role of blockchain to the data stored in a global model. Here, blockchain guarantees the security of the global model parameters by storing the learned data into the blocks [21]. Here, blockchain guarantees the security of the global model parameters by storing the learned data into the blocks. The framework of the blockchain of our proposed model is the same as that of the basic blockchain. Figure 4 explains the task of process of computing block. All blocks consist of previous hash, a hash of the current block, timestamp, nonce, and data field which contains cognitive learned data. The first block is the genesis block whose previous hash field contains all zeroes. All blocks are cryptographically linked to each other through the hash of the previous block. A very minute change in any one of the fields of the block can change the hash of the entire block. Applicability of the consensus algorithm on the block completes the process of verification and validation of the block and then appends the block to the distributed blockchain. The Proof of Work (PoW) consensus algorithm is used in our approach. The miners keep on trying to create the random nonce until they reach the constraints of the target nonce [22]. Once a miner gets the desired nonce, miner obtains the authority of broadcasting the block as a new block to the distributed blockchain. All miners will append the new block to their blockchain, which makes the blockchain consistent. Algorithm 3 explains the procedure followed by miners to compute block by completing the task of nonce calculation.

4. Implementation and Performance Evaluation

Amazon AWS platform has been used which includes different types of 1000 nodes. Some of these nodes are SPV nodes and few are full nodes. Nodes are configured with a Linux Virtual Machine. The privacy of the learned data is tested on the testing environment. Testing was done in 4 sets by changing the number of clients according to different complexity levels. Accuracy, latency, and block generation rate are evaluated to check its performance. The SHA-256 algorithm is used to compute the ID of IoT devices added in this framework. Computed ID is 16 bytes long. All IoT devices are assigned public key and private key for interaction with the other devices and providing secure signatures.

The IoT devices connected to the framework are known by 16 bytes ID. HeteroFL approach is applied for successful cognitive learning to take place. HeteroFL technique minimizes the computation as well as communication complexity of complete process. Training of local models is done in lesser number when compared with the global model. Private Ethereum platform is used for performing blockchain computations. Core i7-8565U CPU 1.80 GHz, 1992 Mhz, 4 Core(s), and 8 Logical Processor(s) are used for implementation. The performance of our proposed scheme has been evaluated for different parameters such as accuracy and memory utilization [23].

Memory utilization of different sizes of blocks is considered, i.e., 10 transactions per block, 20 transactions per block, and 30 transactions per block. Memory utilization mainly depends on the size of basic information of block excluding transactions data and size of the transactions. The data produced by the global models are stored in the blocks. However, to evaluate the appropriate number of transactions to be stored in the block, evaluation of this parameter is done. Figure 5 presents the reduction in memory utilization with the increase in the transactions per block. Experiment proves that less number of transactions per block will consume less memory.

The performance is also evaluated at different learning rates, i.e., 0.005, 0.05, and 0.5. Good accuracy is observed at large data sample sizes also as shown in Figure 6. It is clear from the graph that initially learning is done linearly in all three cases and then it becomes constant, but the best

(i) **Variables**: $w1, wg, s, p, m$
(ii) **Procedure input**: data generated by heterogeneous IoT devices, X_i, and local clients with the data, x_i
(iii) Begin: Initialization of global model parameter, wg, and local clients with the data, x_i
(iv) **for** each round of data production $t = 0, 1, 2, \ldots$ **do**
(v) S ← random set of active clients
(vi) **for** all clients, $k \in S$
(vii) Compute complexity level 'c' based on local information
(viii) Compute output channel shrinkage ratio ($s1$), input channel shrinkage
(ix) ratio ($s2$) and hidden shrinkage ratio (SR)
(x) **End**
(xi) **for** all complexity levels, c **do**
(xii) Compute global model parameter using aggregation of local model parameter
(xiii) **end**
(xiv) **end**
(xv) End

Algorithm 1: Algorithm for cognitive learning using HeteroFL.

(i) Variables: T, l, η
(ii) Procedure input: wk, Xk
(iii) Begin: $B_k \leftarrow$ splitting of local data X_k into various batches of size, T
(iv) After regular interval of time, t do
(v) for batch $bk \in T$ do
(vi) for all clients, $k \in S$
(vii) $l_k \leftarrow \eta \Delta l(Wk, bk)$
(viii) $Wk \leftarrow Wk - lk$
(ix) ratio ($s2$) and hidden shrinkage ratio (SR)
(x) End
(xi) End

Algorithm 2: Blockchain protected cognitive learned data.

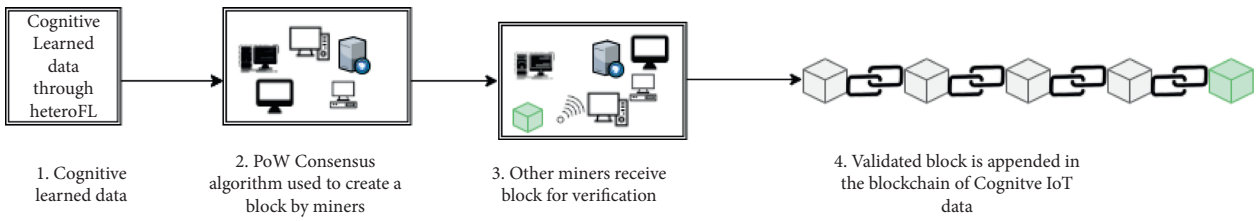


Figure 4: Blockchain protected cognitive data.

(i) Variables: b_i
(ii) Procedure input: cognitive learned data, Y_i, and Miners, M_i
(iii) Begin: Fetch cognitive learned data, Y_i
(iv) while Y_i do
(v) Upload Y_i of fixed size to Miners M_i
(vi) If M_i finds the nonce
(vii) Block b_i of that data is created
(viii) b_i is appended to all local ledgers
(ix) Computation done by other miners is dropped
(x) end
(xi) Winning miner M_i gets the incentive
(xii) Compute global model parameter using aggregation of local model parameter
(xiii) end
(xiv) End

Algorithm 3: Blockchain protected cognitive learned data.

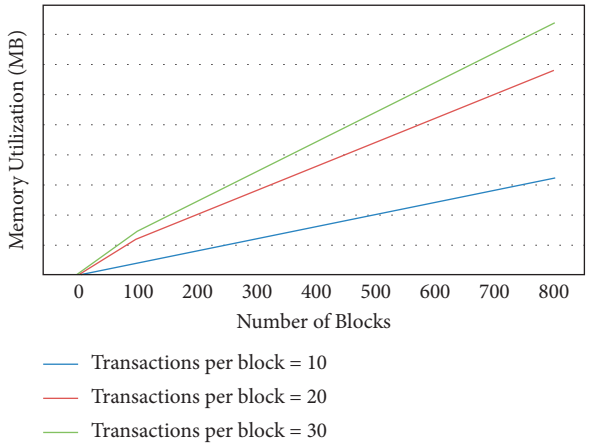


Figure 5: Effect of number of blocks on memory utilization.

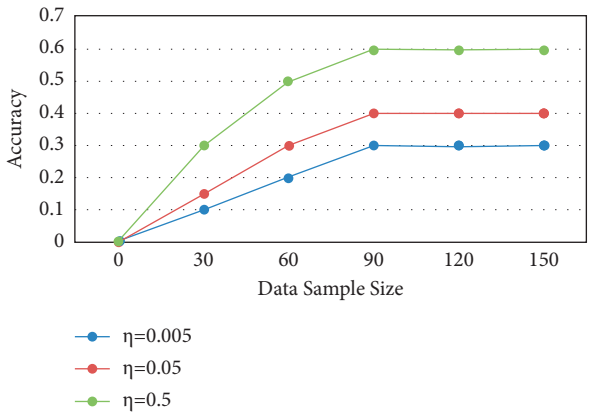


Figure 6: Impact of data sample size on accuracy according to different learning rates.

accuracy is observed in case of high learning rate. Similar graph is expected in case of larger data sample sizes (in thousands). This also guarantees high scalability.

Figure 7 presents execution time taken for the creation of blocks with 10, 20, and 30 transactions per block using one thread. Figure 8 presents execution time taken for the creation of blocks with 10, 20, and 30 transactions per block using two threads. Figure 9 presents execution time taken for the creation of blocks with 10, 20, and 30 transactions per block using four threads. Figure 10 presents execution time taken for the creation of blocks with 10, 20, and 30 transactions per block using eight threads. As the count of blocks rises with rise in number of transactions per block, the execution time also increases whereas the increase in number of threads reduces the execution time. Figure 9 shows very less execution time as the number of threads is four and the number of cores of our system is also four. Evaluation of this parameter proves that there is a dependency on the system's configuration for the execution time of a block. Less execution time will ultimately improve the performance of the network by updating the blocks in a blockchain very quickly, which will make the system consistent with the more recent learned data in its ledger.

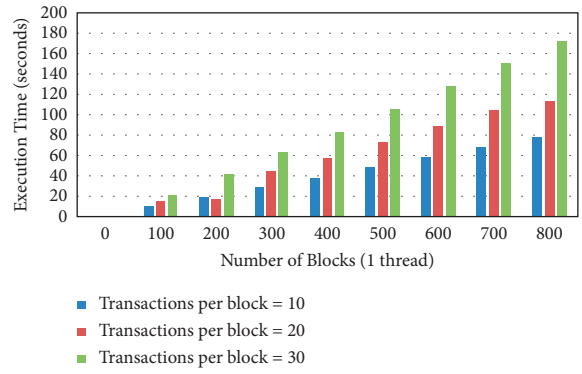


Figure 7: Execution time with 1-thread.

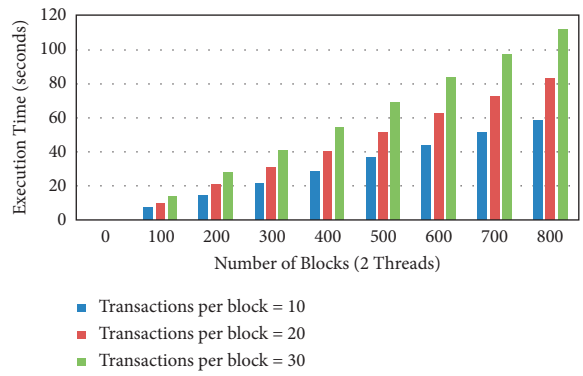


Figure 8: Execution time with 2-threads.

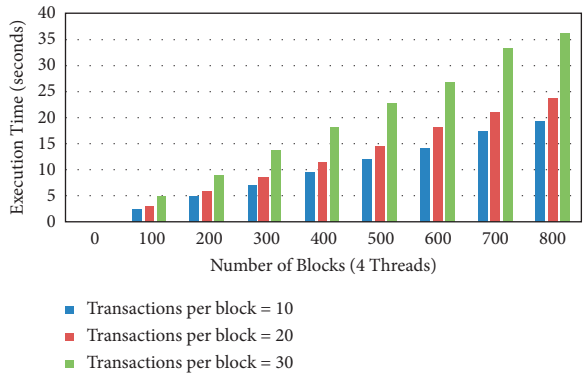


Figure 9: Execution time with 4-threads.

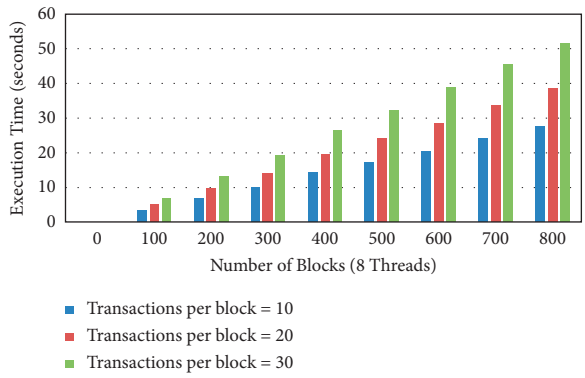


Figure 10: Execution time with 8-threads.

5. Conclusion

We propose heteroFL for cognitive learning to take place from the raw data produced by IoT devices. In this approach, local models are trained by exploiting their full capabilities, and then their aggregation is done to infer an individual global model. HeteroFL takes less number of iterations to produce best results. Blockchain is employed to provide the privacy to the learned data. The PoW consensus algorithm is used to verify and validate a block. From the experiments, accuracy at different learning rates and memory utilization at different number of transactions per block are computed. This approach achieves good results for heterogeneous clients of IoT devices. In future, multimodal learning can be used for addressing heterogeneous learning.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

Acknowledgments

This work was supported by Taif University Researchers Supporting Project Number (TURSP-2020/114), Taif University, Taif, Saudi Arabia.

References

[1] P. Datta and B. Sharma, "A survey on IoT architectures, protocols, security and smart city based applications," in *Proceedings of the 2017 8th in- Ternational Conference on Computing, Communication and Networking Technologies (ICCCNT)*, pp. 1–5, Delhi, India, 2017 July.

[2] R. Chhabra, S. Verma, and C. R. Krishna, "A survey on driver behavior detection techniques for intelligent transportation systems," in *Proceedings of the 2017 7th International Conference on Cloud Computing, Data Science Engineering-Confluence*, pp. 36–41, Noida, India, 2017 January.

[3] F. Li, K. Y. Lam, X. Li, Z. Sheng, J. Hua, and L. Wang, "Advances and emerging challenges in cognitive internet-of-things," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 8, pp. 5489–5496, 2019.

[4] Y. Qian, Y. Jiang, J. Chen et al., "Towards decentralized IoT security enhancement: a blockchain approach," *Computers & Electrical Engineering*, vol. 72, pp. 266–273, 2018.

[5] S. R. Pokhrel and J. Choi, "Federated learning with blockchain for autonomous vehicles: analysis and design challenges," *IEEE Transactions on Communications*, vol. 68, no. 8, pp. 4734–4746, 2020.

[6] E. Diao, J. Ding, and V. Tarokh, "HeteroFL: computation and communication efficient federated learning for heterogeneous clients," arXiv preprint arXiv:2010.01264, 2020.

[7] Y. Liu, A. Liu, T. Wang, X. Liu, and N. N. Xiong, "An intelligent incentive mechanism for coverage of data collection in cognitive Internet of Things," *Future Generation Computer Systems*, vol. 100, pp. 701–714, 2019.

[8] L. Tseng, L. Wong, S. Otoum, M. Aloqaily, and J. B. Othman, "Blockchain for managing heterogeneous internet of things: a perspective architecture," *IEEE network*, vol. 34, no. 1, pp. 16–23, 2020.

[9] Y. Qian, Y. Jiang, L. Hu, M. S. Hossain, M. Alrashoud, and M. Al-Hammadi, "Blockchain-based privacy-aware content caching in cognitive internet of vehicles," *IEEE Network*, vol. 34, no. 2, pp. 46–51, 2020.

[10] S. Buda, C. Wu, W. Bao et al., "Empowering blockchain in vehicular environments with decentralized edges," *IEEE Access*, vol. 8, pp. 202032–202041, 2020.

[11] C. Peng, C. Wu, L. Gao, J. Zhang, K.-L. Alvin Yau, and Y. Ji, "Blockchain for vehicular internet of things: recent advances and open issues," *Sensors*, vol. 20, no. 18, Article ID 5079, 2020.

[12] L. Gao, C. Wu, T. Yoshinaga, X. Chen, and Y. Ji, "Multi-channel blockchain scheme for internet of vehicles," *IEEE Open Journal of the Computer Society*, vol. 2, pp. 192–203, 2021.

[13] Y. Qu, S. R. Pokhrel, S. Garg, L. Gao, and Y. Xiang, "A blockchained federated learning framework for cognitive computing in industry 4.0 networks," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 4, pp. 2964–2973, 2020.

[14] J. Zhu, Y. Song, D. Jiang, and H. Song, "A new deep-Q-learning- based transmission scheduling mechanism for the cognitive Internet of Things," *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 2375–2385, 2017.

[15] Y. Zhang, X. Ma, J. Zhang, M. S. Hossain, G. Muhammad, and S. U. Amin, "Edge intelligence in the cognitive internet of things: improving sensitivity and interactivity," *IEEE Network*, vol. 33, no. 3, pp. 58–64, 2019.

[16] A. M. Saghiri, M. Vahdati, K. Gholizadeh, M. R. Meybodi, M. Dehghan, and H. Rashidi, "A framework for cognitive Internet of Things based on blockchain," in *Proceedings of the 2018 4th International Conference on Web Research (ICWR)*, pp. 138–143, Tehran, Iran, 2018 April.

[17] M. A. Rahman, M. M. Rashid, M. S. Hossain, E. Hassanain, M. F. Alhamid, and M. Guizani, "Blockchain and IoT-based cognitive edge framework for sharing economy services in a smart city," *IEEE Access*, vol. 7, pp. 18611–18621, 2019.

[18] A. Afzal, S. A. R. Zaidi, M. Z. Shakir et al., "The cognitive internet of things: a unified perspective," *Mobile Networks and Applications*, vol. 20, no. 1, pp. 72–85, 2015.

[19] S. Vimal, M. Khari, R. G. Crespo, L. Kalaivani, N. Dey, and M. Kaliappan, "Energy enhancement using Multiobjective Ant colony optimization with Double Q learning algorithm for IoT based cognitive radio networks," *Computer Communications*, vol. 154, pp. 481–490, 2020.

[20] A. M. Elmisery, M. Sertovic, and B. B. Gupta, "Cognitive privacy middleware for deep learning mashup in environmental IoT," *IEEE Access*, vol. 6, pp. 8029–8041, 2017.

[21] J. Ren, J. Li, H. Liu, and T. Qin, "Task offloading strategy with emergency handling and blockchain security in SDN-empowered and fog-assisted healthcare IoT," *Tsinghua Science and Technology*, vol. 27, no. 4, pp. 760–776, 2021.

[22] L. Li, P. Shi, X. Fu, P. Chen, T. Zhong, and J. Kong, "Three dimensional Tradeoffs for consensus algorithms: A review," *IEEE Transactions on Network and Service Management*, 2021.

[23] S. A. Kumar and J. Vassileva, "User acceptance of usable blockchain-based research data sharing system: an extended TAM-based study," in *Proceedings of the 2019 First IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA)*, IEEE, Los Angeles, CA, USA, 2019.

Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 8952219, 16 pages
https://doi.org/10.1155/2021/8952219

Research Article

A Privacy-Preserving Reinforcement Learning Approach for Dynamic Treatment Regimes on Health Data

Xiaoqiang Sun,[1,2] **Zhiwei Sun**,[3] **Ting Wang**,[3] **Jie Feng**,[4,5] **Jiakai Wei**,[6] **and Guangwu Hu**[1]

[1]*School of Computer, Shenzhen Institute of Information Technology, Shenzhen 518172, China*
[2]*Guangdong Key Laboratory of Intelligent Information Processing, College of Electronics and Information Engineering, Shenzhen University, Shenzhen 518060, China*
[3]*School of Artificial Intelligence, Shenzhen Polytechnic, Shenzhen 518055, China*
[4]*Guangzhou Institute of Technology, Xidian University, Guangzhou 510555, China*
[5]*Shaanxi Key Laboratory of Information Communication Network and Security, Xi'an University of Posts & Telecommunications, Xi'an, Shaanxi 710121, China*
[6]*Department of Neonatology, Xi'an Children's Hospital, Xi'an Jiaotong University, Xi'an 710003, China*

Correspondence should be addressed to Zhiwei Sun; smeker@szpt.edu.cn

Received 13 September 2021; Accepted 22 October 2021; Published 23 November 2021

Academic Editor: Celimuge Wu

Based on the clinical states of the patient, dynamic treatment regime technology can provide various therapeutic methods, which is helpful for medical treatment policymaking. Reinforcement learning is an important approach for developing this technology. In order to implement the reinforcement learning algorithm efficiently, the computation of health data is usually outsourced to the untrustworthy cloud server. However, it may leak, falsify, or delete private health data. Encryption is a common method for solving this problem. But the cloud server is difficult to calculate encrypted health data. In this paper, based on Cheon et al.'s approximate homomorphic encryption scheme, we first propose secure computation protocols for implementing comparison, maximum, exponentiation, and division. Next, we design a homomorphic reciprocal of square root protocol firstly, which only needs one approximate computation. Based on the proposed secure computation protocols, we design a secure asynchronous advantage actor-critic reinforcement learning algorithm for the first time. Then, it is used to implement a secure treatment decision-making algorithm. Simulation results show that our secure computation protocols and algorithms are feasible.

1. Introduction

As a recent healthcare tendency, personalized medicine [1] enables the patient to obtain early diagnoses, risk estimation, optimal treatments with low costs by using molecular and cellular analysis technologies, diagnosis results, genetic information, etc. Personalized medicine is usually implemented by the dynamic treatment regime technology [2, 3], which can provide various therapeutic methods according to the time-varying clinical states of the patient. This technology is particularly suitable for coping with complex chronic illnesses, such as diabetes, mental diseases, alcohol dependence, and human immunodeficiency virus infection, which have various stages.

Reinforcement learning [4], which is implemented by trial-and-error and interaction with the dynamic environment, is an important method for developing dynamic treatment regimes, industry automation, vehicular networks [5, 6], and other scenarios [7–12]. Meanwhile, with the developing technologies of internet of things and cloud computing, dynamic treatment regimes that are based on reinforcement learning are becoming increasingly attractive. For example, wearable devices are helpful for monitoring the patient's health data, which include heart rates and blood sugar levels. Next, collected health data are stored on the cloud. Then, the reinforcement learning algorithm can be implemented on these health data for making treatment decisions.

Unfortunately, because of the patient's limited computation ability, health data are usually outsourced to the cloud server for implementing the reinforcement learning algorithm. Because the cloud server may be untrusted, it is likely that health data will be illegally accessed, forged, tampered, or discarded in the process of transmission and computation. In addition, it may be harmful for personal privacy, economic interests, and even the security of human life. For example, as a billing service company, American Medical Collection Agency was intruded in 2019 [13]. This attack affects the health data of about 12 million patients. Besides, the parent firm of this company has filed for bankruptcy. Furthermore, unlike financial data or other types of human-generated data [14], health data are permanent biological data. They cannot be modified or wiped to avoid the damage, which is caused by health data disclosure.

In order to protect health data, we can encrypt health data by using a traditional encryption algorithm. Unfortunately, the reinforcement learning algorithm cannot be executed on the encrypted health data easily and flexibly. Homomorphic encryption [15] supports the operations on the ciphertext. Hence, the cloud server can run the reinforcement learning algorithm on the encrypted health data perfectly by using homomorphic encryption without leaking patient privacy. Finally, the encrypted computation result is returned to the patient. The computation result can be obtained by using the patient's secret key.

In this paper, we endeavor to study the security of health data in the above realistic scenario and focus on the secure implementation of the asynchronous advantage actor-critic (A3C) reinforcement learning algorithm. Taking into account the privacy and computation of health data on the untrusted cloud servers, we adopt homomorphic encryption as the main encryption primitive to carry out our research. Eventually, we make the following three contributions:

(1) Because the efficiency of Cheon et al.'s approximate homomorphic encryption scheme [16] is better than that of fully homomorphic encryption (FHE), we use it to design secure computation protocols, namely, homomorphic comparison protocol, homomorphic maximum protocol, homomorphic exponential protocol, and homomorphic division protocol. Based on these protocols, we first design the homomorphic reciprocal of square root protocol, which needs only one approximate computation

(2) Based on the proposed secure computation protocols, we design the secure A3C reinforcement learning algorithm for the first time. Then, we use it to implement a secure treatment decision-making algorithm

(3) Finally, we simulate the proposed secure computation protocols and algorithms on the personal computer's virtual machine. Then, we demonstrate the efficiency of our secure computation algorithms according to the thorough analysis

The layout of this paper is as follows. Section 2 analyzes related work about homomorphic encryption and secure computation of encrypted health data. Preliminaries are presented in Section 3. Section 4 shows related work about secure dynamic treatment regimes on health data. Building blocks are discussed in Section 5. Section 6 describes the proposed privacy-preserving A3C reinforcement learning algorithm and treatment decision-making algorithm. Performance results are shown and analyzed in Section 7. Finally, this paper is concluded in Section 8.

2. Related Work

In this section, we introduce related work about reinforcement learning, homomorphic encryption, and the computation of encrypted health data, which are described as follows.

Reinforcement learning can be mainly classified as value-based algorithms, policy-based algorithms, and actor-critic algorithms. Value-based algorithms usually compute the optimum cumulative reward and give a suggested policy. As a typical value-based algorithm, Q-learning is used for estimating the utility of the individual pair that consists of a state and an action. Q-learning has been applied for path planning [17, 18] in vehicular networks. Policy-based algorithms can evaluate the optimum policy directly. Williams [19] proposed a policy-based algorithm REINFORCE. Actor-critic algorithms combine the advantages of value-based algorithms and policy-based algorithms. The A3C reinforcement learning algorithm [20] is an actor-critic algorithm. It can work in discrete action spaces as well as continuous action spaces [21].

The concept of homomorphic encryption begins from privacy homomorphism [15]. According to the types of supported homomorphic operations, homomorphic encryption can be divided into partial homomorphic encryption (PHE), somewhat homomorphic encryption (SWHE), and FHE. PHE only supports homomorphic addition or homomorphic multiplication. SWHE is the basis of FHE. SWHE supports finite homomorphic addition and homomorphic multiplication. FHE supports arbitrary homomorphic addition and homomorphic multiplication.

In 2009, Gentry [22] designed the first FHE scheme, which is based on ideal lattices. Since then, homomorphic encryption has become a research hotspot. Next, in order to improve the efficiency of homomorphic operations, Gentry et al. [23] first constructed the FHE scheme, which is based on the approximate eigenvector method. In this scheme, the ciphertext noise increases linearly after each homomorphic multiplication. Although homomorphic multiplication of this scheme is efficient, it does not support the technique of single instruction multiple data (SIMD) [24]. Then, based on the learning with errors over rings (RLWE) [25] assumption and relinearization technique [24], Brakerski et al. [24] designed a FHE scheme, which supports the SIMD technique. However, this scheme does not support approximate homomorphic operations. Hence, based on Brakerski et al.'s scheme [24], Cheon et al. [16] proposed an improved homomorphic encryption scheme.

In terms of the computation of encrypted health data by using homomorphic encryption, there exist following several schemes. Khedr and Gulak [26] first proposed an optimized

homomorphic encryption scheme, which is based on Gentry's scheme [23]. Then, the proposed scheme is applied for secure medical computations, which include comparison, Pearson goodness-of-fit test, and logistic regression. Sun et al. [27] implemented secure average heart rate, long QT syndrome detection, and chi-square tests by using Dowlin et al.'s FHE scheme [28]. Based on Boneh et al.'s homomorphic encryption scheme [29], Poon et al. [30] implemented the secure Fisher's exact test algorithm, which is often used to guarantee the statistical stability of genetic analysis. Raisaro et al. [31] used homomorphic encryption to explore genomic cohorts securely in a real scenario. In 2019, based on the distributed two trapdoors public-key algorithm [32] and Q-learning algorithm, Liu et al. [2] constructed a secure reinforcement learning model, which is helpful for making treatment decisions dynamically. Based on Fan's SWHE scheme [33], Jiang et al. [34] performed secure and efficient feature point detection and image matching for retinal images of diabetic retinopathy.

However, most of the above schemes are based on PHE, which only supports homomorphic addition or homomorphic multiplication. PHE may not support homomorphic multiplication. If homomorphic multiplication is required in some schemes, excessive rounds of interactions are needed. FHE can avoid this problem. But FHE confronts the problem of the efficiency of homomorphic operations. Furthermore, the Q-learning algorithm is usually used as the reinforcement learning algorithm in the above schemes. There does not exist an approach that can implement the A3C reinforcement learning algorithm securely.

3. Preliminaries

In this section, we begin with basic notations and definition of Cheon et al.'s approximate homomorphic encryption scheme. Then, we give the introduction of the A3C algorithm.

3.1. Basic Notations. Let $[z] = (z - 1/2, z + 1/2]$, where z is a real number. Let $[z]_p = z - [z/p] \cdot p \in (-p/2, p/2]$, where p denotes an integer.

Let $R = \mathbb{Z}/\langle \Phi_m(x) \rangle$ denote the ring modulo $\Phi_m(x)$, where λ is the security parameter, m is a positive integer, and $\Phi_m(x)$ is the mth cyclotomic polynomial. $R_q = \mathbb{Z}_q[x]/\langle \Phi_m(x) \rangle$ represents the ring modulo q and $\Phi_m(x)$, where q is the prime modulus, $q \geq 2$.

As for an integer $h > 0$, the distribution $\mathcal{HWT}(h)$ is selected from $\{0, \pm 1\}$ randomly with the Hamming weight h. As for a rational number $\sigma > 0$, the distribution $\mathcal{DG}(\sigma^2)$ outputs a vector, which coefficients are selected from the discrete Gaussian distribution with the variance σ^2. As for a rational number $0 \leq \rho \leq 1$, the distribution $\mathcal{ZO}(\rho)$ is chosen from $\{0, \pm 1\}$ randomly, where $\rho/2$ is the probability that ± 1 is selected and $1 - \rho$ is the probability that 0 is selected.

3.2. Learning with Errors over Rings. In 2010, Lyubashevsky et al. [25] first proposed the RLWE assumption, which is described as follows.

Definition 1 (RLWE). The $\mathrm{RLWE}_{\lambda,q,\chi}$ assumption is to distinguish two distributions, namely, $(a, a \cdot s + e) \in R_q \times R_q$ and $(a, c) \in \mathrm{Unif}(R_q \times R_q)$, where $a \in R_q$ and $s \in R_q$, e is an error term, and Unif represents uniform random. Lyubashevsky et al. [25] proved that the security of RLWE assumption relies on ideal lattices.

3.3. Cheon et al.'s Homomorphic Encryption Scheme. In this subsection, we introduce Cheon et al.'s approximate homomorphic encryption scheme AHE = (KeyGen, Enc, Add, Sub, Mul, Dec, ReScale, Ecd, Dcd) [16] as follows:

(i) AHE.KeyGen$(1^\lambda, p, L)$: given the security parameter λ, an integer p, and a level L, this algorithm first sets $q_l = p^l \cdot q_0$, where q_0 is a fixed integer, $l = L, \cdots, 1$. It selects a power-of-two integer $M = M(\lambda, q_L)$, an integer $P = P(\lambda, q_L)$, and a rational number $\sigma = \sigma(\lambda, q_L)$. Next, it chooses a vector s from $\mathcal{HWT}(h)$. The secret key sk is set as $(1, s)$. A ring element a is sampled from R_{q_L}. An error term e is sampled from $\mathcal{DG}(\sigma^2)$. The public key pk is set as $(b, a) \in R_{q_L}^2$, where $b = -a \cdot s + e (\mathrm{mod}\ q_L)$. Then, a ring element a' is sampled from $R_{P \cdot q_L}$. An error term e' is sampled from $\mathcal{DG}(\sigma^2)$. The evaluation key evk is set as $(b', a') \in R_{P \cdot q_L}^2$, where $b' = -a' \cdot s + e' (\mathrm{mod}\ P \cdot q_L)$

(ii) AHE.Enc(pk, m): in order to encrypt a plaintext m, this algorithm samples an integer v from $\mathcal{ZO}(0.5)$. In addition, it chooses two error terms e_0 and e_1 from $\mathcal{DG}(\sigma^2)$. m is encrypted as the ciphertext $c = v \cdot \mathrm{pk} + (m + e_0, e_1) (\mathrm{mod}\ q_L)$

(iii) AHE.Dec(sk, c): in this algorithm, $c = (b, a)$ is decrypted as $b + a \cdot s (\mathrm{mod}\ q_l)$

(iv) AHE.Add(c', c''): in this algorithm, input parameters include two ciphertexts $c' = ([c'_0]_{q_l}, [c'_1]_{q_l})$ and $c'' = ([c''_0]_{q_l}, [c''_1]_{q_l})$, which are under the same secret key. Then, the additive ciphertext $c_{\mathrm{add}} = ([c_0]_{q_l} + [c''_0]_{q_l}, [c_1]_{q_l} + [c''_1]_{q_l})$

(v) AHE.Mul(evk, c', c''): in this algorithm, input parameters include evk, two ciphertexts $c' = ([c'_0]_{q_l}, [c'_1]_{q_l})$ and $c'' = ([c''_0]_{q_l}, [c''_1]_{q_l})$, where c' and c'' are under the same secret key. Then, the ciphertext $c_{\mathrm{temp}} = (c_0, c_1, c_2) = ([c'_0 \cdot c''_0]_{q_l}, [c'_0 \cdot c''_1 + c'_1 \cdot c''_0]_{q_l}, [c'_1 \cdot c''_1]_{q_l})$. The multiplicative ciphertext $c_{\mathrm{mul}} = (c_0, c_1) + [P^{-1} \cdot c_2 \cdot \mathrm{evk}] (\mathrm{mod}\ q_l)$

(vi) AHE.ReScale$_{l \longrightarrow l'}(c)$: as for a ciphertext $c \in R_{q_l}^2$ at the level l, the new ciphertext $c' = [(q_{l'}/q_l)c] (\mathrm{mod}\ q'_l)$

(vii) AHE.Ecd(z, Δ): as for a vector $z = (z_0, z_1, \cdots, z_{N/2-1}) \in \mathbb{C}^{N/2}$ and a scaled factor $\Delta > 0$, this algorithm outputs $z' = [\sigma^{-1}(\Delta \cdot z)] \in R$, where σ^{-1} is the inverse operation of a canonical embedding map $\sigma(\cdot)$

(viii) AHE.Dcd(z', Δ): as for $z' \in R$, this algorithm outputs $z = \Delta^{-1} \cdot \sigma(m) \in \mathbb{C}^{N/2}$

In Cheon et al.'s scheme, the decryption noise should be bounded by $8\sqrt{2} \cdot \sigma \cdot N + 6\sigma \cdot \sqrt{N} + 16\sigma \cdot \sqrt{h \cdot N}$ for the correctness of decryption. In addition, the noise of the rescaling ciphertext is at most $\sqrt{N/3} \cdot (3 + 8\sqrt{h})$. Furthermore, the noise of the multiplicative ciphertext should be less than $P^{-1} \cdot q_l \cdot 8\sigma \cdot N/\sqrt{3} + \sqrt{N/3} \cdot (3 + 8\sqrt{h})$. The details about the analysis of Cheon et al.'s scheme can be found in [16].

3.4. Asynchronous Advantage Actor-Critic Reinforcement Learning Algorithm. In 2016, Mnih et al. [20] proposed the asynchronous advantage actor-critic reinforcement learning algorithm, which is based on combining the value-based method and the policy-based method. One advantage of the A3C algorithm is that it can work in discrete action spaces as well as continuous action spaces. In addition, in order to improve the learning efficiency of the A3C algorithm, multiple asynchronous actor-learners, which can interact with the environment and acquire various independent exploration policies, are running in parallel. The details of the A3C algorithm are described as follows.

In the A3C algorithm, there is a policy function $\pi(a_t \mid s_t ; \theta)$ and a value function $V(s_t ; \theta_v)$, where a_t denotes an action at the time step t, s_t denotes a state at the time step t, and θ and θ_v are two parameters. In addition, $V(s_t ; \theta_v)$ and $\pi(a_t \mid s_t ; \theta)$ will be updated $t_{\max}$ times, where $t_{\max}$ denotes the maximum step. $V(s_t ; \theta_v)$ and $\pi(a_t \mid s_t ; \theta)$ are usually approximated by a single convolutional neural network. Specifically, $V(s_t ; \theta_v)$ is based on a linear layer. $\pi(a_t \mid s_t ; \theta)$ is relied on a softmax layer. Namely, $V(s_t ; \theta_v) = x(s_t) \cdot \theta_v$, where $x(s_t)$ is a function which is related to s_t. $\pi(a_t \mid s_t ; \theta) = e^{f(a_t|s_t)\cdot\theta} / \sum_{j=0}^{t_{\max}} e^{f(a_j|s_j)\cdot\theta}$, a_j is an action at the time step j, and $f(a_j \mid s_j)$ is a function which is related to a_j and s_j.

Furthermore, the A3C algorithm uses two loss functions, namely, policy loss function and value loss function, which are described as follows. On the one hand, the policy loss function

$$f_\pi(\theta) = \ln \pi(a_t \mid s_t ; \theta) \cdot (R - V(s_t ; \theta_v)) + \beta \cdot H(\pi(s_t ; \theta)), \tag{1}$$

where R is the reward and the parameter k depends on the state. In addition, the upper bound of k is $t_{\max}$. r_{t+i} is the immediate reward. The discount factor $\gamma \in (0, 1]$. The entropy function $H(\pi(s_t ; \theta))$ can be set as $-\Sigma_{i=0}^{k} f(a_t \mid s_t) \cdot \theta \cdot \ln \pi(s_t ; \theta)$. The hyperparameter β can adjust the intensity of the entropy regulation term. Then, we can conclude that

$$\begin{aligned} f_\pi(\theta) = {} & \left(f(a_t \mid s_t) - \Sigma_{j=0}^{t_{\max}} f(a_j \mid s_j)\right) \cdot \theta \cdot (R - x(s_t) \cdot \theta_v) \\ & + \beta \cdot \left(-\Sigma_{i=0}^{k} f(a_t \mid s_t) \cdot \left(f(a_t \mid s_t) - \Sigma_{j=0}^{t_{\max}} f(a_j \mid s_j)\right) \cdot \theta^2\right). \end{aligned} \tag{2}$$

Hence, the differentiation of $f_\pi(\theta)$ with respect to θ is

$$\begin{aligned} \frac{\partial f_\pi(\theta)}{\partial \theta} = {} & \left(f(a_t \mid s_t) - \Sigma_{j=0}^{t_{\max}} f(a_j \mid s_j)\right) \\ & \cdot (R - x(s_t) \cdot \theta_v) + 2\beta \cdot \theta \cdot f(a_t \mid s_t) \\ & \cdot \left(f(a_t \mid s_t) - \Sigma_{j=0}^{t_{\max}} f(a_j \mid s_j)\right). \end{aligned} \tag{3}$$

On the other hand, the value loss function

$$f_v(\theta_v) = (R - V(s_t ; \theta_v))^2 = (R - x(s_t) \cdot \theta_v)^2. \tag{4}$$

Hence, the differentiation of $f_v(\theta_v)$ with respect to θ_v is

$$\frac{\partial f_v(\theta_v)}{\partial \theta_v} = 2(R - x(s_t) \cdot \theta_v)\left(\frac{\partial R}{\partial \theta_v} - x(s_t)\right). \tag{5}$$

Based on the above two loss functions and corresponding differentiation, the A3C reinforcement learning algorithm is defined in Algorithm 1, which is described as follows. Algorithm 1 requires input parameters θ, θ_v, θ', θ'_v, T, t, $t_{\max}$, $T_{\max}$, t_g, η, W, and α, where the definition of these parameters are shown in Table 1. In order to implement Algorithm 1, we first set $T = 0$, $t = 1$. If $T < T_{\max}$ and $\omega \in [1, W]$, we implement the iteration, which is shown as follows. Global gradients $d\theta$ and $d\theta_v$ are set as 0. θ' and θ'_v are synchronized as θ and θ_v, respectively. We set $t_0 = t$ and obtain the system state $\mathcal{S}_t \in \mathcal{S}$, where $\mathcal{S}$ is a state set, $\mathcal{S} = (\mathcal{S}_0, \cdots, \mathcal{S}_{\varphi-1})$, and φ is the number of states. Next, we repeat a subalgorithm until $t - t_0 \neq t_{\max}$. In this subalgorithm, the action $\mathcal{A}_t \in \mathcal{A}$ is obtained by using $\pi(\mathcal{A}_t \mid \mathcal{S}_t ; \theta')$, where $\mathcal{A}$ is an action set, $\mathcal{A} = (\mathcal{A}_0, \cdots, \mathcal{A}_{\chi-1})$, and χ is the number of actions. We execute $\mathcal{A}_t$, get the reward R_t, and observe the next state $\mathcal{S}_{t+1}$, where R_t is set as $\Sigma_{i=0}^{k-1} \gamma^i \cdot r_{t+i} + \gamma^k \cdot V(\mathcal{S}_{t+k} ; \theta'_v) = \Sigma_{i=0}^{k-1} \gamma^i \cdot r_{t+i} + \gamma^k \cdot x(\mathcal{S}_{t+k}) \cdot \theta'_v$. In addition, we set $t = t + 1$. After the implementation of the above subalgorithm, we observe whether $t \% t_g$ equals to 0. If $\mathcal{S}_t$ is terminal, we set $R = 0$. If $\mathcal{S}_t$ is nonterminal $R = V(\mathcal{S}_t ; \theta'_v) = x(\mathcal{S}_t) \cdot \theta'_v$. Then, we repeat a subalgorithm from $i = t - 1$ to $i = t_0$. In this subalgorithm, R is set as $R_t + \gamma \cdot R$, namely, $R = \Sigma_{i=0}^{k-1} \gamma^i \cdot r_{t+i} + \gamma^k \cdot x(\mathcal{S}_t) \cdot \theta'_v + \gamma \cdot R$. We compute

$$\begin{aligned} \frac{\partial f_\pi(\theta')}{\partial \theta'} = {} & \left(f(\mathcal{A}_t \mid \mathcal{S}_t) - \sum_{j=0}^{t_{\max}} f(\mathcal{A}_t \mid \mathcal{S}_t)\right) \\ & \cdot (R - x(\mathcal{S}_t) \cdot \theta_v) + 2\beta \cdot \theta' \cdot f(\mathcal{A}_t \mid \mathcal{S}_t) \\ & \cdot \left(f(\mathcal{A}_t \mid \mathcal{S}_t) - \sum_{j=0}^{t_{\max}} f(\mathcal{A}_t \mid \mathcal{S}_t)\right), \end{aligned} \tag{6}$$

$A3C(\theta, \theta_v, \theta', \theta'_v, T, t, t_{\max}, T_{\max}, t_g, \eta, W, \alpha)$:
Input: θ, θ_v, θ', θ'_v, T, t, $t_{\max}$, $T_{\max}$, t_g, η, W, α.
Output: θ, θ_v.
Set $T = 0$, $t = 1$.
While $T < T_{\max}$ do.
  For $w = 1$ to W do.
    Set $d\theta = 0$, $d\theta_v = 0$.
    Synchronize $\theta' = \theta$, $\theta'_v = \theta_v$.
    Set $t_0 = t$ and get $\mathcal{S}_t$.
    Repeat.
      Get $\mathcal{A}_t$ according to $\pi(\mathcal{A}_t \mid \mathcal{S}_t; \theta')$.
      Execute $\mathcal{A}_t$, get R_t and observe $\mathcal{S}_{t+1}$.
      $t = t + 1$.
    Until $t - t_0 = t_{\max}$
    If $\mathcal{S}_t$ is terminal, $R = 0$.
    If $\mathcal{S}_t$ is non-terminal, $R = x(\mathcal{S}_t) \cdot \theta'_v$.
    For $i = t - 1$ to t_0 do
      $R = R_t + \gamma \cdot R$.
      Compute $\partial f_\pi(\theta')/\partial\theta' = (f(\mathcal{A}_t \mid \mathcal{S}_t) - \sum_{j=0}^{t_{\max}} f(\mathcal{A}_t \mid \mathcal{S}_t))$ $(R - x(\mathcal{S}_t) \cdot \theta_v) + 2\beta \cdot \theta' f(\mathcal{A}_t \mid \mathcal{S}_t) \cdot (f(\mathcal{A}_t \mid \mathcal{S}_t) - \sum_{j=0}^{t_{\max}}$ $f(\mathcal{A}_t \mid \mathcal{S}_t))$.
      Compute $d\theta = d\theta + (\partial f_\pi(\theta')/\partial\theta')$.
      Compute $\partial f_v(\theta'_v)/\partial\theta'_v = 2(R - x(\mathcal{S}_t) \cdot \theta'_v) \cdot (\partial R/\partial\theta'_v)$.
      Compute $d\theta_v = d\theta_v + (\partial f_\pi(\theta'_v)/\partial\theta'_v)$.
    End for.
    Compute $g = \alpha \cdot g + (1 - \alpha)(d\theta)^2$, $g_v = \alpha \cdot g_v + (1 - \alpha)(d\theta_v)^2$.
    Compute $\theta = \theta - \eta(d\theta/\sqrt{g + \varepsilon})$, $\theta_v = \theta_v - \eta(d\theta_v/\sqrt{g_v + \varepsilon})$.
  End for
End while

Algorithm 1: A3C reinforcement learning algorithm.

Table 1: Notations.

Symbol	Description
θ, θ_v	Shared parameter vectors in the global network
θ', θ'_v	Thread-specific parameter vectors in the local network
T	Global counter
t	Local step counter
$t_{\max}$, $T_{\max}$	Upper bounds
t_g	An integer
η	Learning rate
W	Number of agents
α	Momentum

where $\partial f_\pi(\theta')/\partial\theta'$ is the differentiation of $f_\pi(\theta')$ with respect to θ'. $d\theta$ is set as $d\theta + \partial f_\pi(\theta')/\partial\theta'$. We compute

$$\frac{\partial f_v\left(\theta'_v\right)}{\partial\theta'_v} = 2\left(R - x(\mathcal{S}_t) \cdot \theta'_v\right) \cdot \frac{\partial R}{\partial\theta'_v}, \tag{7}$$

where $\partial f_v(\theta'_v)/\partial\theta'_v$ is the differentiation of $f_v(\theta'_v)$ with respect to θ'_v and $\partial R/\partial\theta'_v$ is the differentiation of R with respect to θ'_v. Finally, θ and θ_v can be updated by using equations $\theta = \theta - \eta(d\theta/\sqrt{g + \varepsilon})$ and $\theta_v = \theta_v - \eta(d\theta_v/\sqrt{g_v + \varepsilon})$, respectively, where $g = \alpha \cdot g + (1 - \alpha)(d\theta)^2$ and $g_v = \alpha \cdot g_v + (1 - \alpha)(d\theta_v)^2$.

4. Secure Dynamic Treatment Regimes on Health Data

4.1. System Model. As shown in Figure 1, the system model of secure dynamic treatment regimes on health data consists of four parts, namely, undiagnosed patient, key generation center, cloud servers, and historical data owners, which are described as follows:

(i) The undiagnosed patient's current state is collected by using wearable devices, which integrate modules of physiological sensors, weak computation, and communication. Wearable devices include smart bracelet, smart glasses, sleep monitoring sensors, and smart watch. They can collect a variety of health data, such as body temperature, heart rate, blood sugar, and blood volume index. Then, these health

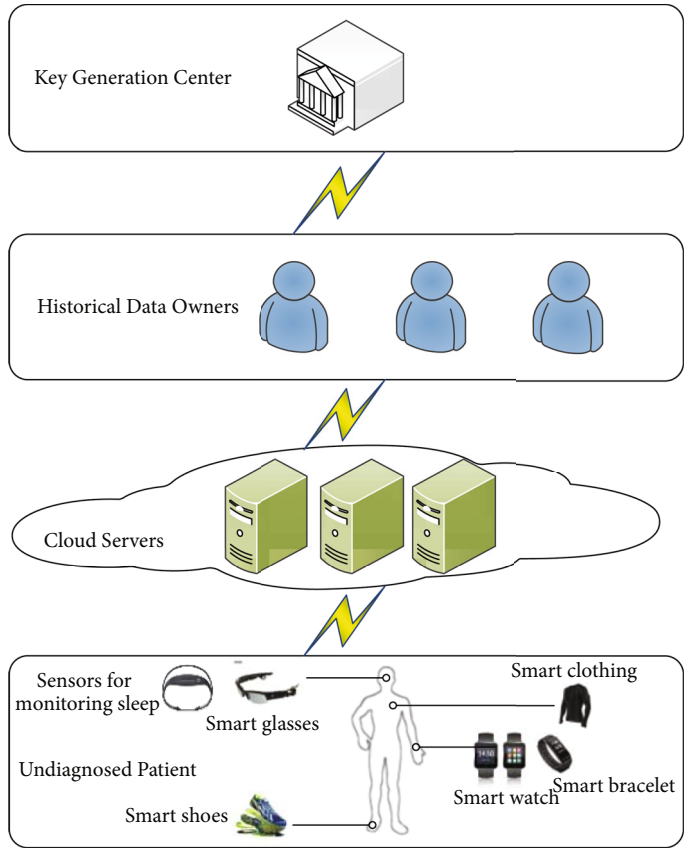


Figure 1: The model of secure dynamic treatment regimes on health data.

data are transmitted to cloud servers for computation. Based on the returned computation result, the patient can obtain the diagnosis

(ii) Key generation center is an indispensable and independent entity, which is trusted by the other entities in this system model. It is responsible for distributing and managing all the public keys and private keys of Cheon et al.'s homomorphic encryption scheme for wearable devices and undiagnosed patients via a secure channel

(iii) Cloud servers have the powerful data storage space. Hence, they store and manage ciphertexts, which come from undiagnosed patients and wearable devices. Additionally, they can perform some computations on these ciphertexts

(iv) Historical data owners have a sequence of medical data and their corresponding decision results. These encrypted data are transmitted and stored on cloud servers. Cloud servers can compute these ciphertexts for training the reinforcement learning model

4.2. Attack Model. In this paper, we suppose that the entities in the system model are honest-but-curious. Namely, the entities strictly follow the designed protocols. But they are interested in acquiring medical data of other entities. We suppose that there is an adversary $\mathbf{A}_1^*$ in the attack model. The goal of $\mathbf{A}_1^*$ is to guess the plaintexts of the challenge historical data owners' ciphertexts or the challenge wearable devices' ciphertexts.

In order to acquire the ciphertexts of historical data owners and wearable devices, middle ciphertext results during the execution of privacy-preserving A3C reinforcement learning algorithm and treatment decision-making algorithm (Section 6), $\mathbf{A}_1^*$ eavesdrops on the communication links among the entities in the system model. However, these ciphertexts are based on Cheon et al.'s approximate homomorphic encryption scheme [16]. Hence, $\mathbf{A}_1^*$ cannot decrypt these ciphertexts without knowing their secret keys. It can be guaranteed by using the semantic security of Cheon et al.'s scheme. In addition, the key generation center distributes key pairs to historical data owners and wearable devices in a secure way. Furthermore, due to the lack of private keys of these ciphertexts, $\mathbf{A}_1^*$ cannot generate evaluation keys. Hence, $\mathbf{A}_1^*$ cannot transform these ciphertexts into some domains that $\mathbf{A}_1^*$ can decrypt. Besides, $\mathbf{A}_1^*$ cannot get useful information by adding or multiplying a plaintext with these ciphertexts. In a conclusion, the proposed model is secure.

4.3. System Setup and Overview. Our secure model of dynamic treatment regimes consists of two phases, which are described as follows.

(i) Training dataset outsourcing and initialization: historical data owners initialize input parameters θ, θ_v,

learning rate η, and discount factor γ. The state set $\mathcal{S} = (\mathcal{S}_0, \cdots, \mathcal{S}_{\varphi-1})$ and action set $\mathcal{A} = (\mathcal{A}_0, \cdots, \mathcal{A}_{\chi-1})$ are encrypted as $c_{\mathcal{S}} = (c_{\mathcal{S}_0}, \cdots, c_{\mathcal{S}_{\varphi-1}})$ and $c_{\mathcal{A}} = (c_{\mathcal{A}_0}, \cdots, c_{\mathcal{A}_{\chi-1}})$. Then, historical data owners send $c_{\mathcal{S}}$, $c_{\mathcal{A}}$, η, γ, and other parameters to cloud servers for storage and computation, where $j = 0, \cdots, n-1$, and n is the number of historical data owners

(ii) Outsourced sequential treatment decision making: in order to achieve sequential treatment decision making, the undiagnosed patient's current state x, which comes from wearable devices, is encrypted as c_x. Then, c_x is transmitted to cloud servers for treatment decision making. Based on our privacy-preserving A3C reinforcement learning algorithm and treatment decision-making algorithm, cloud servers output the encrypted treatment decision c_a. The undiagnosed patient decrypts c_a to obtain the treatment decision a by using his own secret key

5. Building Blocks

5.1. Encoding Rational Number. In order to implement the privacy-preserving A3C reinforcement learning algorithm, we need to encrypt health data. Health data are usually rational numbers. However, most of the homomorphic encryption schemes only support homomorphic operations over integers. They cannot cope with rational numbers. Hence, in this paper, we take Cheon et al.'s encoding technique [16], which can encode a rational number. Then, the rational number can be converted to a ring element just like using the integer encoding technique. We can use Cheon et al.'s scheme [16] to encrypt the converted result.

5.2. Homomorphic Comparison Protocol. In order to implement comparison for our secure computation algorithms, we design a homomorphic comparison protocol by using Cheon et al.'s scheme [16] and Sun et al.'s method [35]. We suppose that the user owns plaintexts m_0 and m_1. Then, the user uses Cheon et al.'s scheme to encrypt these plaintexts. The ciphertexts are c_0 and c_1, respectively. The user owns the secret key sk. The cloud server is responsible for storing the ciphertexts. As shown in Algorithm 2, the cloud server first computes the ciphertext $c_b = t + c_0 - c_1$, where t is the plaintext modulus. Next, the cloud server transmits c_b to the user. The user uses sk to decrypt c_b. The decryption result is b. If $b > t$, $m_0 > m_1$. If $b = t$, $m_0 = m_1$. If $b < t$, $m_0 < m_1$. For example, we suppose that $t = 2$, $m_0 = 0$, $m_1 = 1$, and then $c_b = 2 + c_0 - c_1$, where c_0 and c_1 are ciphertexts of 0 and 1, respectively. The decryption result of c_b equals to 1. Hence, $m_0 < m_1$.

5.3. Homomorphic Maximum Protocol. In order to compute the encrypted index of the largest plaintext, we design a homomorphic maximum protocol by using the above protocol and Sun et al.'s method [35]. We suppose that the user owns $m_0, \cdots, m_{k-1}$, where k is the number of plaintexts. The user uses Cheon et al.'s scheme [16] to encrypt these plaintexts. The ciphertexts are $c_0, \cdots, c_{k-1}$, respectively. The

comp(c_0, c_1):
Input: c_0 and c_1.
Output: c_b.
1. Compute $c_b = t + c_0 - c_1$.
2. Return c_b to the user.
3. If $b > t$, $m_0 > m_1$.
   If $b = t$, $m_0 = m_1$.
   If $b < t$, $m_0 < m_1$.

Algorithm 2: Homomorphic comparison protocol.

argmax$(c_0, c_1, \cdots, c_{k-1})$:
Input: $c_0, c_1, \cdots, c_{k-1}$.
Output: $c_{\max}$.
1. Set $c_{\max} = c_0$.
2. For $i = 1$ to $k - 1$ do.
3. $c_b = \text{comp}(c_{\max}, c_i)$.
4. Decrypt c_b to get b.
5. If $b = 1$, $c_{\max} = c_i$.
6. $i = i + 1$.
End for

Algorithm 3: Homomorphic maximum protocol.

user owns the secret key sk. As shown in Algorithm 3, the cloud server computes $c_b = \text{comp}(c_{\max}, c_i)$, where $c_{\max}$ is initialized as c_0. If the decryption result $b < t$, $c_{\max} = c_i$, $i = i + 1$. The cloud server continues to compare $c_{\max}$ and c_i until $i > k - 1$. Finally, the user can obtain the index of the largest plaintext by decrypting $c_{\max}$. For example, there exist ciphertexts c_2, c_3, and c_4, whose plaintexts are 2, 3, and 4, respectively. We set $c_{\max} = c_2$. Next, we first compare $c_{\max}$ and c_3. $c_{\max}$ is updated as c_3. Then, we compare $c_{\max}$ and c_4. $c_{\max}$ is updated as c_4. After the decryption of $c_{\max}$, the user gets the maximum result 4.

5.4. Homomorphic Exponential Protocol. In this section, based on the Taylor series, we begin to describe the homomorphic exponential protocol. We suppose that the user owns the plaintext m. Next, it is encrypted as c_m by using Cheon et al.'s homomorphic encryption scheme. Only the user has the secret key. Then, c_m is stored on the cloud server. In the homomorphic exponential protocol (Algorithm 4), the cloud server first computes the ciphertext $c_{e^m} = 1 + c_m + c_m^2/2! + c_m^3/3! + \cdots + c_m^n/n!$ without decryption, where n denotes an integer. The precision of e^m increases with the increasing of n. Then, c_{e^m} is returned to the user. The user gets the exponential result e^m by using his secret key. For example, we can set $m = 4$, $n = 3$, and then $c_{e^2} = 1 + c_4 + c_4^2/2! + c_4^3/3!$, where c_{e^2} and c_4 are ciphertexts of e^2 and 4, respectively. After the decryption of c_{e^2}, the user gets the exponential result e^2.

5.5. Homomorphic Division Protocol. In this section, we begin to describe the homomorphic division protocol. We suppose that the user owns plaintexts m_0, m_1, and m_2. Then, they are encrypted as c_{m_0}, c_{m_1}, and c_{m_2} by using Cheon et al.'s

exp (c_m, n):
Input: c_m, n.
Output: c_{e^m}.
1. Compute $c_{e^m} = 1 + c_m + \cdots + (c_m^n/n!)$.
2. Return c_{e^m} to the user.

Algorithm 4: Homomorphic exponential protocol.

div $(c_{m_0}, c_{m_1}, c_{m_2})$:
Input: c_{m_0}, c_{m_1} and c_{m_2}.
Output: The ciphertext c_{div}.
1. Compute $c_{\text{add}} = c_{m_0} + c_{m_1}$.
2. Return c_{add} to the user.
3. Decrypt c_{add} to obtain add.
4. Calculate rev = 1/add.
5. rev is encrypted as c_{rev}.
6. c_{rev} is transmitted to the cloud server.
7. Calculate $c_{\text{div}} = c_{m_2} \cdot c_{\text{rev}}$.
8. c_{div} is returned to the user.

Algorithm 5: Homomorphic division protocol.

homomorphic encryption scheme. Only the user has the secret key. Then, c_{m_0}, c_{m_1}, and c_{m_2} are transmitted to the cloud server. In order to output the ciphertext c_{div} of the plaintext $m_2/(m_0 + m_1)$, we design the homomorphic division protocol (Algorithm 5), which is described as follows. The cloud server first computes the ciphertext $c_{\text{add}} = c_{m_0} + c_{m_1}$ without decryption, where the plaintext of c_{add} is add = $m_0 + m_1$. Then, c_{add} is returned to the user. The user gets the plaintext add by using his secret key. The user calculates rev = 1/add. rev is encrypted as c_{rev} by using Cheon et al.'s scheme. c_{rev} is transmitted to the cloud server. The cloud server calculates the ciphertext $c_{\text{div}} = c_{m_2} \cdot c_{\text{rev}}$. Finally, c_{div} is returned to the user. After the decryption of c_{div}, the user gets the division result div = $m_2/(m_0 + m_1)$. For example, we set $m_0 = 6$, $m_1 = 1$, and $m_2 = 2$. Then, the cloud server calculates $c_{\text{add}} = c_{m_0} + c_{m_1} = c_3$, where the plaintext of c_3 is 3. c_3 is returned to the user. After the decryption of c_3, the user calculates rev = 1/3 = 0.33. The ciphertext c_{rev} is sent to the cloud server. The cloud server calculates $c_{\text{div}} = c_{m_2} \cdot c_{\text{rev}} = c_{0.33\times 6} = c_{1.98}$, where the plaintext of $c_{1.98}$ is 1.98.

5.6. Homomorphic Reciprocal of Square Root Protocol. In this section, we begin to describe the homomorphic reciprocal of square root protocol. The traditional method is to compute the ciphertext of the approximate square root firstly. Then, it computes the approximate reciprocal of square root homomorphically. However, two approximate computations will affect the precision of the final result. Hence, based on Lomont's fast inverse square root algorithm [36], we design a new homomorphic reciprocal of square root protocol (Algorithm 6), which only needs one approximate computation. In our protocol, we suppose that the user owns the floating number $m = (1 + m_0)2^{m_1 - 127}$, where $0 < m_0 < 1$ and

$1/\sqrt{c_m}$:
Input: The ciphertext c_m.
Output: $c_{1/\sqrt{m}}$.
1. Transmit c_m to the user.
2. Decrypt c_m to obtain $m = (1 + m_0)2^{m_1 - 127}$.
3. Convert m to obtain $m' = m_1 \cdot 2^{23} + m_0 \cdot 2^{23}$.
4. Encrypt m' as $c_m{}'$.
5. Transmit $c_m{}'$ to the cloud server.
6. Compute $c_{\text{temp}} = 3/2(127 - 0.045)2^{23} - 0.5c_m{}'$.
7. Transmit c_{temp} to the user.
8. Decrypt c_{temp} to obtain temp = $(\text{temp}_1 + \text{temp}_0) \cdot 2^{23}$.
9. Convert temp to temp$'$ = $(1 + \text{temp}_0)2^{\text{temp}_1 - 127}$.
10. Encrypt temp$'$ as $c_{\text{temp}}{}'$.
11. Transmit $c_{\text{temp}}{}'$ to the cloud server.
12. Compute $c_{1/\sqrt{m}} = 3/2c_{\text{temp}}{}' - 1/2c_m \cdot c_{\text{temp}'}^3$.
13. Return $c_{1/\sqrt{m}}$ to the user.

Algorithm 6: Homomorphic reciprocal of square root protocol.

$0 < m_1 < 255$. It is encrypted as c_m by using Cheon et al.'s homomorphic encryption scheme. Only the user has the secret key. c_m is stored on the cloud server. c_m is first transmitted to the user. The user decrypts c_m by his secret key. The decryption result m is converted to an integer $m' = m_1 \cdot 2^{23} + m_0 \cdot 2^{23}$. m' is encrypted as $c_m{}'$ by using Cheon et al.'s scheme. Next, $c_m{}'$ is transmitted to the cloud server. The cloud server computes the intermediate ciphertext

$$c_{\text{temp}} = \frac{3}{2}(127 - 0.045)2^{23} - 0.5c_{m'}, \tag{8}$$

where the plaintext of c_{temp} is the floating number temp. Then, the cloud server sends c_{temp} to the user. The user decrypts c_{temp} to obtain temp = $\text{temp}_1 \cdot 2^{23} + \text{temp}_0 \cdot 2^{23}$, where $0 < \text{temp}_0 < 1$ and $0 < \text{temp}_1 < 255$. temp is converted to an integer temp$'$ = $(1 + \text{temp}_0)2^{\text{temp}_1 - 127}$. temp$'$ is encrypted as $c_{\text{temp}}{}'$. $c_{\text{temp}}{}'$ is transmitted to the cloud server. The cloud server computes the ciphertext

$$c_{1/\sqrt{m}} = \frac{3}{2}c_{\text{temp}}{}' - \frac{1}{2}c_m \cdot c_{\text{temp}'}^3. \tag{9}$$

Then, $c_{1/\sqrt{m}}$ is returned to the user. The user gets the reciprocal of square root $1/\sqrt{m}$ by using his secret key. For example, we set $m = (1 + 0.25)^{124-127} = 0.156$. Then, m is converted to $m' = 62 \cdot 2^{23} + 0.125 \cdot 2^{23}$. The ciphertext $c_m{}'$ of m' is sent to the cloud server. The cloud server computes

$$c_{t\text{emp}} = \frac{3}{2}(127 - 0.045)2^{23} - 0.5c_{m'}. \tag{10}$$

The user decrypts c_{temp} to obtain temp = $128 \cdot 2^{23} + 0.3075 \cdot 2^{23}$. temp is converted to temp$'$ = $(1 + 0.3075)2^{128-127} = 2.615$. The ciphertext $c_{\text{temp}}{}'$ of temp$'$ is sent to

PA3C($\theta, \theta_v, \theta', \theta'_v, T, t, t_{\max}, T_{\max}, t_g, \eta, W, \alpha, c_{\mathcal{S}}, c_{\mathcal{A}}$):
Input: θ, θ_v, θ', θ'_v, T, t, $t_{\max}$, $T_{\max}$, t_g, η, W, α, $c_{\mathcal{S}}$, $c_{\mathcal{A}}$.
Output: θ, θ_v.
Set $T = 0$, $t = 1$.
While $T < T_{\max}$ do
  For $w = 1$ to W do
    Set $d\theta = 0$, $d\theta_v = 0$. Synchronize $\theta' = \theta$, $\theta'_v = \theta_v$.
    Set $t_0 = t$, get $c_{\mathcal{S}_t}$.
    Repeat
      Get $c_{\mathcal{A}_t}$ according to argmax$(c_{\pi_0}, \cdots, c_{\pi_{t_{\max}}})$.
      Execute $c_{\mathcal{A}_t}$, get c_{R_t}.
      Observe $c_{\mathcal{S}_{t+1}}$, $t = t + 1$.
    Until $t - t_0 = t_{\max}$
    If $c_{\mathcal{S}_t}$ is terminal, $c_R = 0$.
    If $c_{\mathcal{S}_t}$ is non-terminal, $c_R = x(c_{\mathcal{S}_t}) \cdot \theta'_v$.
    For $i = t - 1$ to t_0 do
      If $c_{\mathcal{S}_t}$ is terminal
        $c_R = \Sigma_{i=0}^{k-1} \gamma^i \cdot r_{t+i} + \gamma^k \cdot x(c_{\mathcal{S}_t}) \cdot \theta'_v$.
        Compute $c_{\partial f_\pi(\theta')/\partial\theta'}$, $c_{\partial f_v(\theta'_v)/\partial\theta'_v}$.
      End if
      If $c_{\mathcal{S}_t}$ is non-terminal
        $c_R = \Sigma_{i=0}^{k-1} \gamma^i \cdot r_{t+i} + \gamma^k \cdot x(c_{\mathcal{S}_t}) \cdot \theta'_v + \gamma \cdot x(c_{\mathcal{S}_t}) \cdot \theta'_v$.
        Compute $c_{\partial f_\pi(\theta')/\partial\theta'}$, $c_{\partial f_v(\theta'_v)/\partial\theta'_v}$.
      End if
      Compute $c_{d\theta} = c_{d\theta} + c_{\partial f_\pi(\theta')/\partial\theta'}$.
      Compute $c_{d\theta_v} = c_{d\theta_v} + c_{\partial f_\pi(\theta'_v)/\partial\theta'_v}$.
    End for
    $c'_{d\theta}$ and $c'_{d\theta_v}$ are returned to the cloud server.
    Compute $c_g = \alpha c_g + (1 - \alpha)(c_{d\theta})^2$, $c_{g_v} = \alpha c_{g_v} + (1 - \alpha)(c_{d\theta_v})^2$.
    c'_g and c'_{g_v} are sent to the cloud server.
    Set $c_g = c'_g$, $c_{g_v} = c'_{g_v}$.
    Compute $c_\theta = \theta - \eta(c'_{d\theta}/\sqrt{c_g + \varepsilon})$, $c_{\theta_v} = \theta_v - \eta(c'_{d\theta_v}/\sqrt{c_{g_v} + \varepsilon})$.
    c_θ and c_{θ_v} are returned to the user.
    Decrypt c_θ and c_{θ_v} to obtain θ and θ_v.
  End for
End while

Algorithm 7: Privacy-preserving A3C reinforcement learning algorithm.

the cloud server. The cloud server computes the ciphertext

$$c_{1/\sqrt{0.156}} = \frac{3}{2} c_{\text{temp}}{}' - \frac{1}{2} c_m \cdot c_{\text{temp}'}^3. \tag{11}$$

The user decrypts $c_{1/\sqrt{m}}$ to obtain $1/\sqrt{0.156} = 3/2 \cdot 2.615 - 1/2 \cdot 0.156 \cdot 2.615^3 \approx 2.528$.

6. Privacy-Preserving Computation Algorithms

6.1. Privacy-Preserving A3C Reinforcement Learning Algorithm. In this section, we begin to describe how to implement a privacy-preserving A3C reinforcement learning algorithm by using Cheon et al.'s approximate homomorphic encryption scheme [16]. As shown in Algorithm 7, we describe the privacy-preserving A3C reinforcement learning algorithm as follows. Algorithm 7 requires input parameters θ, θ_v, θ', θ'_v, T, t, $t_{\max}$, $T_{\max}$, t_g, η, W, α, $c_{\mathcal{S}}$, and $c_{\mathcal{A}}$. Set $T = 0$, $t = 1$. If $T < T_{\max}$ and $w \in [1, W]$, we implement the iteration, which is shown as follows. $d\theta$ and $d\theta_v$ are set as 0. θ' and θ'_v are synchronized as θ and θ_v, respectively. We set $t_0 = t$ and obtain the encrypted system state $c_{\mathcal{S}_t} \in c_{\mathcal{S}}$. Next, we repeat a subalgorithm until $t - t_0 \neq t_{\max}$. In this subalgorithm, the encrypted action $c_{\mathcal{A}_t} \in c_{\mathcal{A}}$ is obtained based on argmax$(c_{\pi_0}, c_{\pi_1}, \cdots, c_{\pi_{t_{\max}}})$, where $c_{\pi_j} = e^{f(c_{\mathcal{A}_j} | c_{\mathcal{S}_j}) \cdot \theta'}$, $j = 0, 1, \cdots, t_{\max}$. We execute $c_{\mathcal{A}_t}$ and get the encrypted reward

$$c_{R_t} = \Sigma_{i=0}^{k-1} \gamma^i \cdot r_{t+i} + \gamma^k \cdot x\left(c_{\mathcal{S}_{t+k}}\right) \cdot \theta_v. \tag{12}$$

We observe the next encrypted state $c_{\mathcal{S}_{t+1}} \in c_{\mathcal{S}}$. In addition,

we set $t = t + 1$. After the implementation of the above subalgorithm, we observe whether $c_{\mathcal{S}_t}$ is terminal. Then, we repeat a subalgorithm from $i = t - 1$ to $i = t_0$. In this subalgorithm, if $c_{\mathcal{S}_t}$ is terminal, c_R is set as

$$c_R = \Sigma_{i=0}^{k-1}\gamma^i \cdot r_{t+i} + \gamma^k \cdot c_{x(\mathcal{S}_t)} \cdot \theta'_v. \tag{13}$$

The cloud server computes

$$\begin{aligned} c_{\frac{\partial f_\pi(\theta')}{\partial \theta'}} = &\left(f\left(c_{\mathcal{A}_t} \mid c_{\mathcal{S}_t}\right) - \Sigma_{j=0}^{t_{\max}} f\left(c_{\mathcal{A}_t} \mid c_{\mathcal{S}_t}\right)\right) \\ &\cdot \left(\Sigma_{i=0}^{k-1}\gamma^i \cdot r_{t+i} + \gamma^k \cdot x\left(c_{\mathcal{S}_t}\right) \cdot \theta'_v - x\left(c_{\mathcal{S}_t}\right) \cdot c_{\theta_v}\right) \\ &+ 2\beta \cdot c_{\theta}{}' \cdot f\left(c_{\mathcal{A}_t} \mid c_{\mathcal{S}_t}\right) \\ &\cdot \left(f\left(c_{\mathcal{A}_t} \mid c_{\mathcal{S}_t}\right) - \Sigma_{j=0}^{t_{\max}} f\left(c_{\mathcal{A}_j} \mid c_{\mathcal{S}_j}\right)\right). \end{aligned} \tag{14}$$

The cloud server computes

$$\begin{aligned} c_{\frac{\partial f_v(\theta'_v)}{\partial \theta'_v}} = &2\left(\Sigma_{i=0}^{k-1}\gamma^i \cdot r_{t+i} + \gamma^k \cdot x\left(c_{\mathcal{S}_t}\right) \cdot c_{\theta'_v} - x\left(c_{\mathcal{S}_t}\right) c_{\theta'_v}\right) \\ &\cdot \left(\gamma^k \cdot x\left(c_{\mathcal{S}_t}\right) - x\left(c_{\mathcal{S}_t}\right)\right). \end{aligned} \tag{15}$$

If $c_{\mathcal{S}_t}$ is nonterminal, c_R is set as

$$c_{R_t} + \gamma c_R = \Sigma_{i=0}^{k-1}\gamma^i \cdot r_{t+i} + \gamma^k \cdot x\left(c_{\mathcal{S}_t}\right) \cdot c_{\theta'_v} + \gamma \cdot x\left(c_{\mathcal{S}_t}\right) \cdot \theta'_v. \tag{16}$$

The cloud server computes

$$\begin{aligned} c_{\frac{\partial f_\pi(\theta')}{\partial \theta'}} = &\left(f\left(c_{\mathcal{A}_t} \mid c_{\mathcal{S}_t}\right) - \Sigma_{j=0}^{t_{\max}} f\left(c_{\mathcal{A}_t} \mid c_{\mathcal{S}_j}\right)\right) \\ &\cdot \left(\Sigma_{i=0}^{k-1}\gamma^i \cdot r_{t+i} + \gamma^k \cdot x\left(c_{\mathcal{S}_t}\right) \cdot \theta'_v + \gamma \cdot x\left(c_{\mathcal{S}_t}\right)\right. \\ &\left. \cdot \theta'_v - x\left(c_{\mathcal{S}_t}\right) \cdot \theta_v\right) + 2\beta \cdot \theta' f\left(c_{\mathcal{A}_t} \mid c_{\mathcal{S}_t}\right) \\ &\cdot \left(f\left(c_{\mathcal{A}_t} \mid c_{\mathcal{S}_t}\right) - \Sigma_{j=0}^{t_{\max}} f\left(c_{\mathcal{A}_t} \mid c_{\mathcal{S}_j}\right)\right), \end{aligned} \tag{17}$$

where $c_{\partial f_\pi(\theta')}{}'/\partial\theta'$ is the ciphertext of $\partial f_\pi(\theta')/\partial\theta'$. The cloud server computes

$$\begin{aligned} c_{\frac{\partial f_v(\theta'_v)}{\partial \theta'_v}} = &2\left(\Sigma_{i=0}^{k-1}\gamma^i \cdot r_{t+i} + \gamma^k \cdot x\left(c_{\mathcal{S}_t}\right) \cdot \theta'_v + \gamma \cdot x\left(c_{\mathcal{S}_t}\right)\right. \\ &\left. \cdot \theta'_v - x\left(c_{\mathcal{S}_t}\right)\theta'_v\right) \cdot \left(\gamma^k \cdot x\left(c_{\mathcal{S}_t}\right) + \gamma \cdot x\left(c_{\mathcal{S}_t}\right) - x\left(c_{\mathcal{S}_t}\right)\right), \end{aligned} \tag{18}$$

where $c_{\partial f_v(\theta'_v)/\partial\theta'_v}$ is the ciphertext of $\partial f_v(\theta'_v)/\partial\theta'_v$. Then, $c_{d\theta}$ is set as $c_{d\theta} + c_{\partial f_\pi(\theta')}{}'/\partial\theta'$. $c_{d\theta_v}$ is set as $c_{d\theta_v} + c_{\partial f_\pi(\theta'_v)/\partial\theta'_v}$. The cloud server computes $c_g = \alpha c_g + (1-\alpha)(c_{d\theta})^2$ and $c_{g_v} = \alpha c_{g_v} + (1-\alpha)(c_{d\theta_v})^2$. The cloud server sends $c_{d\theta}$ and $c_{d\theta_v}$ to the user. The user decrypts $c_{d\theta}$ and $c_{d\theta_v}$ to obtain $d\theta$ and $d\theta_v$. $d\theta$ and $d\theta_v$ are encrypted as $c'_{d\theta}$ and $c'_{d\theta_v}$. $c'_{d\theta}$ and $c'_{d\theta_v}$ are returned to the cloud server. In order to reduce the depth of homomorphic multiplication for the calculation of c_g and c_{g_v}, the cloud server sends c_g and c_{g_v} to the user. The user decrypts c_g and c_{g_v} to obtain g and g_v. g and g_v are encrypted as c'_g and c'_{g_v}. The user sends c'_g and c'_{g_v} to the cloud server. The cloud server sets $c_g = c'_g$ and $c_{g_v} = c'_{g_v}$. Finally, based on the above homomorphic reciprocal of square root protocol, θ and θ_v can be updated by using equations $c_\theta = \theta - \eta(c'_{d\theta}/\sqrt{c_g + \varepsilon})$ and $c'_{\theta_v} = \theta_v - \eta(c_{d\theta_v}/\sqrt{c_{g_v} + \varepsilon})$, respectively. After the execution of Algorithm 7, we can get the encrypted optimized parameters c_θ and c_{θ_v}, which are returned to the user. The user decrypts c_θ and c_{θ_v} to obtain θ and θ_v, which can be used the implementation of secure treatment decision-making algorithm.

Table 2: Symbols and initial values.

Symbol	Initial value
θ, θ_v	0.5
θ', θ'_v	0.5
$T_{\max}$, $t_{\max}$	1
t_0	1
k	1
γ	0.6
r_1, r_2	0.5
β	0.1
W	1
g, g_v	0
α	0.5
ε	0.01
η	0.1

In order to better understand Algorithm 7, we give an example, which is described as follows. In this example, as shown in Table 2, we set the initial values of related parameters. We suppose that $(c_{\mathcal{S}_0},\cdots,c_{\mathcal{S}_4})$ are ciphertexts of $\mathcal{S} = (\mathcal{S}_0,\cdots,\mathcal{S}_4) = (0.1,0.1,0.15,0.3,0.35)$, respectively. $(c_{\mathcal{A}_0}, \cdots,c_{\mathcal{A}_4})$ are ciphertexts of $\mathcal{A} = (\mathcal{A}_0,\cdots,\mathcal{A}_4) = (0.1,0.1,0.15,0.3,0.35)$, respectively. For the convenience of computation, we let $x(c_{\mathcal{S}_j}) = c_{\mathcal{S}_j}$, $f(c_{\mathcal{A}_j} \mid c_{\mathcal{S}_j}) = c_{\mathcal{A}_j} c_{\mathcal{S}_j}$, $\pi(\mathcal{A}_j \mid \mathcal{S}_j) = e^{f(\mathcal{A}_j|\mathcal{S}_j)}$, $i = 0, 1, 2, 3, 4$, $j = 0, 1, 2, 3, 4$. The cloud server first computes $c_{\pi_0} = e^{f(c_{\mathcal{A}_0}|c_{\mathcal{S}_0})} \cdot \theta'$ and $c_{\pi_1} = e^{f(c_{\mathcal{A}_1}|c_{\mathcal{S}_1})} \cdot \theta'$, where the ciphetext of c_{π_0} is $e^{0.1\times0.1\times0.5} = e^{0.005}$ and the cyphertext of c_{π_1} is $e^{0.1\times0.1\times0.5} = e^{0.005}$. Based on the implementation of the protocol $\text{argmax}(c_{\pi_0}, c_{\pi_1})$, $\mathcal{A}_1$ is executed. The cloud server computes $c_{R_t} = r_t + \gamma x(c_{\mathcal{S}_t})\theta_v$, where

$$R_t = r_t + \gamma x(\mathcal{S}_t)\theta_v = 0.5 + 0.6 \times 0.1 \times 0.5 = 0.505. \tag{19}$$

Set $t = 2$. Because $\mathcal{S}_2$ is nonterminal, the cloud server computes $c_R = x(c_{\mathcal{S}_2}) \cdot \theta'_v$, where

$$R = x(\mathcal{S}_2) \cdot \theta'_v = 0.15 \times 0.5 = 0.075. \tag{20}$$

Then, the cloud server computes

$$c_R = r_2 + \gamma \cdot x(c_{\mathcal{S}_2}) \cdot \theta'_v + \gamma \cdot x(c_{\mathcal{S}_2}) \cdot \theta'_v, \tag{21}$$

where

$$R = r_2 + \gamma \cdot x(\mathcal{S}_2) \cdot \theta'_v + \gamma \cdot x(\mathcal{S}_2) \cdot \theta'_v = 0.5 + 0.6 \times 0.15 \times 0.5 + 0.6 \times 0.15 \times 0.5 = 0.59. \tag{22}$$

The cloud server computes

$$c_{\partial f_\pi(\theta')/\partial\theta'} = \left(f(c_{\mathcal{A}_2} \mid c_{\mathcal{S}_2}) - \Sigma_{j=0}^{1} f\left(c_{\mathcal{A}_j} \mid c_{\mathcal{S}_j}\right)\right)(r_2 + \gamma \cdot x(c_{\mathcal{S}_2}) \cdot \theta'_v + \gamma \cdot x(c_{\mathcal{S}_2}) \cdot \theta'_v - x(c_{\mathcal{S}_2})\theta_v) + 2\beta \cdot \theta' \cdot f(c_{\mathcal{A}_2} \mid c_{\mathcal{S}_2}) \cdot \left(f(c_{\mathcal{A}_2} \mid c_{\mathcal{S}_2}) - \Sigma_{j=0}^{1} f\left(c_{\mathcal{A}_j} \mid c_{\mathcal{S}_j}\right)\right), \tag{23}$$

where

$$\begin{aligned}\frac{\partial f_\pi(\theta')}{\partial\theta'} &= \left(f(\mathcal{A}_2 \mid \mathcal{S}_2) - \Sigma_{j=0}^{1} f(\mathcal{A}_j \mid \mathcal{S}_j)\right)(r_2 + \gamma \cdot x(\mathcal{S}_2) \cdot \theta'_v + \gamma \cdot x(\mathcal{S}_2) \cdot \theta'_v - x(\mathcal{S}_2)\theta_v) \\ &\quad + 2\beta \cdot \theta' \cdot f(\mathcal{A}_2 \mid \mathcal{S}_2) \cdot \left(f(\mathcal{A}_2 \mid \mathcal{S}_2) - \Sigma_{j=0}^{1} f(\mathcal{A}_j \mid \mathcal{S}_j)\right) \\ &= (0.15 \times 0.15 - (0.1 \times 0.1 + 0.1 \times 0.1)) \cdot (0.5 + 0.6 \times 0.15 \times 0.5 + 0.6 \times 0.15 \times 0.5 - 0.15 \times 0.5) \\ &\quad + 2 \times 0.1 \times 0.5 \times 0.15 \times 0.15(0.15 \times 0.15 - (0.1 \times 0.1 + 0.1 \times 0.1)) = 0.0013.\end{aligned} \tag{24}$$

The cloud server computes

$$c_{\partial f_v(\theta'_v)/\partial\theta'_v} = 2\left(r_2 + \gamma \cdot x(c_{\mathcal{S}_2}) \cdot c_{\theta'_v} + \gamma \cdot x(c_{\mathcal{S}_2}) \cdot \theta'_v - x(c_{\mathcal{S}_2})\theta'_v\right) \cdot (\gamma \cdot x(c_{\mathcal{S}_2}) + \gamma \cdot x(c_{\mathcal{S}_2}) - x(c_{\mathcal{S}_2})), \tag{25}$$

where

$$\begin{aligned}\frac{\partial f_v(\theta'_v)}{\partial\theta'_v} &= 2\left(r_2 + \gamma \cdot x(\mathcal{S}_2) \cdot \theta'_v + \gamma \cdot x(\mathcal{S}_2) \cdot \theta'_v - x(\mathcal{S}_2)\theta'_v\right) \cdot (\gamma \cdot x(\mathcal{S}_2) + \gamma \cdot x(\mathcal{S}_2) - x(\mathcal{S}_2)) \\ &= 2(0.5 + 0.6 \times 0.15 \times 0.5 + 0.6 \times 0.15 \times 0.5 - 0.15 \times 0.5)(0.6 \times 0.15 + 0.6 \times 0.15 - 0.15) = 0.0309.\end{aligned} \tag{26}$$

Set $c_{d\theta} = 0$, $c_{d\theta_v} = 0$. The cloud server computes $c_{d\theta} = c_{d\theta} + c_{\partial f_\pi(\theta')/\partial\theta'}$, where

$$d\theta = d\theta + \frac{\partial f_\pi(\theta')}{\partial\theta'} = 0.0013. \tag{27}$$

We compute $c_{d\theta_v} = c_{d\theta_v} + c_{\partial f_\pi(\theta'_v)/\partial\theta'_v}$, where

$$d\theta_v = d\theta_v + \frac{\partial f_\pi(\theta'_v)}{\partial\theta'_v} = 0.0309. \tag{28}$$

With the help of the user, the cloud server gets refreshed ciphertexts $c_{d\theta_v}$ and $c_{d\theta_v}$. The cloud server computes $c_g = \alpha c_g + (1-\alpha)(c_{d\theta})^2$, where

$$\mathrm{g} = \alpha\mathrm{g} + (1-\alpha)(\mathrm{d}\theta)^2 = (1-0.5) \times 0.0013^2 = 0.000000845. \tag{29}$$

The cloud server computes $c_{g_v} = \alpha c_{g_v} + (1-\alpha)(c_{d\theta_v})^2$, where

$$g_v = \alpha g_v + (1-\alpha)(d\theta_v)^2 = (1-0.5) \times 0.0309^2 = 0.000477405. \tag{30}$$

After the cloud server obtains ciphertexts c'_g and c'_{g_v}, we set $c_g = c'_g$ and $c_{g_v} = c'_{g_v}$. The cloud server computes $c_\theta = \theta - \eta(c'_{d\theta}/\sqrt{c_g + \varepsilon})$, where

$$\theta = \theta - \eta\frac{\mathrm{d}\theta}{\sqrt{\mathrm{g} + \varepsilon}} = 0.5 - 0.1\frac{0.0013}{\sqrt{0.000000845 + 0.01}} \approx 0.4987. \tag{31}$$

The cloud server computes $c_{\theta_v} = \theta_v - \eta(c'_{d\theta_v}/\sqrt{c_{g_v} + \varepsilon})$, where

$$\theta_v = \theta_v - \eta\frac{d\theta_v}{\sqrt{g_v + \varepsilon}} = 0.5 - 0.1\frac{0.0309}{\sqrt{0.000477405 + 0.01}} \approx 0.4698. \tag{32}$$

The user can obtain refreshed $\theta \approx 0.4987$ and $\theta_v \approx 0.4698$ by using his secret key.

6.2. Secure Treatment Decision-Making Algorithm. In this subsection, based on the above privacy-preserving A3C reinforcement learning algorithm, secure treatment decision-making algorithm $\mathrm{TDM}(\theta, \theta_v, c_x, c_{\mathcal{S}}, c_{\mathcal{A}})$ is implemented in Algorithm 8, which is described as follows. In this algorithm, input parameters include θ, θ_v, and the undiagnosed patient's encrypted current state c_x, $c_{\mathcal{S}}$, and $c_{\mathcal{A}}$. Set the index col = 0. The ciphertext $c_{i,j}$ is initiated, where $i = 0, \cdots, \chi - 1$ and $j = 0, \cdots, \varphi - 1$. Firstly, $c_{\mathcal{S}}$'s element $c_{\mathcal{S}_j}$ is compared with c_x by using the above homomorphic comparison protocol

TDM($\theta, \theta_v, c_x, c_{\mathcal{S}}, c_{\mathcal{A}}$):
Input: θ, θ_v, c_x, $c_{\mathcal{S}}$, $c_{\mathcal{A}}$.
Output: c_{dr}.
For $i = 0$ to $\chi - 1$.
  For $j = 0$ to $\varphi - 1$.
  Initiate $c_{i,j}$.
  End for
End for
Initiate c_b.
For $j = 0$ to $\varphi - 1$.
  $c_b = \text{comp}(c_{\mathcal{S}_j}, c_x)$.
End for
Decrypt c_b to obtain b.
If $b = t$.
Set col = j.
$c_v = V(c_{\mathcal{S}_{\text{col}}}; \theta_v) = x(c_{\mathcal{S}_j})\theta_v$.
Set $c_{\text{hor}} = c_0$.
For $i = 0$ to $\chi - 1$.
  $c_{\pi,i} = \pi(c_{\mathcal{A}_i} \mid c_{\mathcal{S}_{\text{col}}}) = e^{f(c_{\mathcal{A}_i}|c_{\mathcal{S}_{\text{col}}})\theta} / \sum_{j=0}^{t_{\max}} e^{f(c_{\mathcal{A}_j}|c_{\mathcal{S}_j})\theta}$.
  $c_{i,\text{col}} = c_v \cdot c_{\pi,i}$.
End for
Set index = 0.
$c_{\text{hor,col}} = \text{argmax}(c_{0,\text{col}}, \cdots, c_{\chi-1,\text{col}})$.
Set index = hor.
$c_{dr} = c_{\mathcal{A}_{\text{index}}}$.

Algorithm 8: Secure treatment decision-making algorithm.

Table 3: The distribution of the encrypted probability.

Encrypted probability		Encrypted actions			
		$c_{\mathcal{A}_0}$	$c_{\mathcal{A}_1}$	$c_{\mathcal{A}_2}$	$c_{\mathcal{A}_3}$
Encrypted states	$c_{\mathcal{S}_0}$	$c_{0,0}$	$c_{1,0}$	$c_{2,0}$	$c_{3,0}$
	$c_{\mathcal{S}_1}$	$c_{0,1}$	$c_{1,1}$	$c_{2,1}$	$c_{3,1}$
	$c_{\mathcal{S}_2}$	$c_{0,2}$	$c_{1,2}$	$c_{2,2}$	$c_{3,2}$
	$c_{\mathcal{S}_3}$	$c_{0,3}$	$c_{1,3}$	$c_{2,3}$	$c_{3,3}$

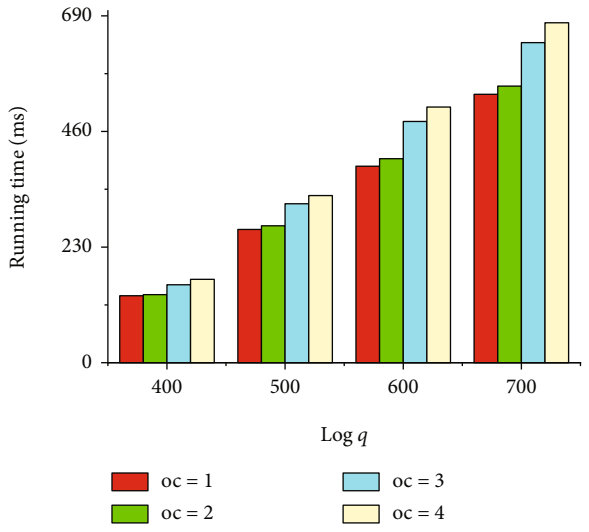


Figure 2: The efficiency of our homomorphic comparison protocol.

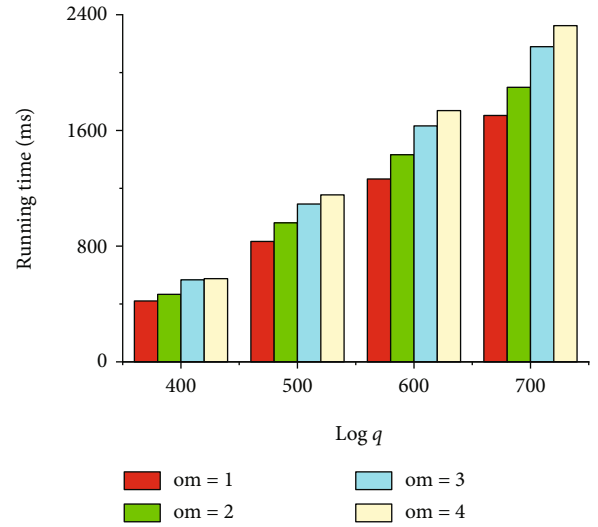


Figure 3: The efficiency of our homomorphic maximum protocol ($k = 5$).

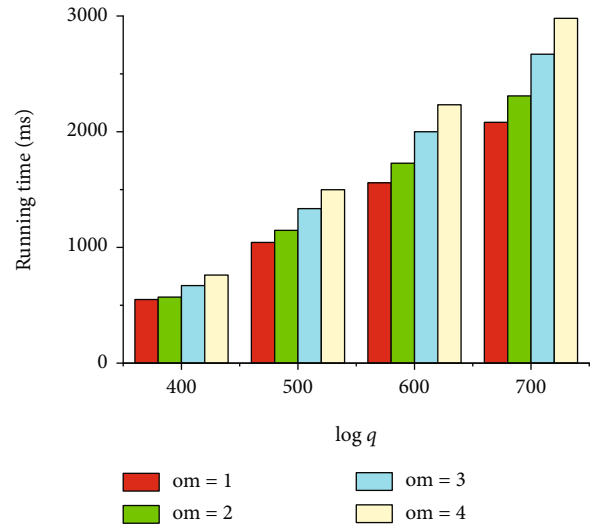


Figure 4: The efficiency of our homomorphic maximum protocol ($k = 6$).

$\text{comp}(c_{\mathcal{S}_j}, c_x)$. c_b is the encrypted comparison result. After the decryption of c_b, if the comparison result $b = t$, it means that $\mathcal{S}_j = x$, set col = j, where $j = 0, \cdots, \varphi - 1$. Next, the cloud server computes the value function's encrypted value $c_v = V(c_{\mathcal{S}_{\text{col}}}; \theta_v) = x(c_{\mathcal{S}_j}) \cdot \theta_v$, where the plaintext of c_v is $V(\mathcal{S}_{\text{col}}; \theta_v)$. Set the ciphertext $c_{\text{hor}} = c_0$. The cloud server computes the policy function's encrypted value

$$c_{\pi,i} = \pi(c_{\mathcal{A}_i} \mid c_{\mathcal{S}_{\text{col}}}) = \frac{e^{f(c_{\mathcal{A}_i}|c_{\mathcal{S}_{\text{col}}})\theta}}{\sum_{j=0}^{t_{\max}} e^{f\left(c_{\mathcal{A}_j}|c_{\mathcal{S}_j}\right)\theta}}, \tag{33}$$

where the plaintext of $c_{\pi,i}$ is $\pi(\mathcal{A}_i \mid \mathcal{S}_{\text{col}})$. The cloud server computes homomorphic multiplication between c_v and $c_{\pi,i}$, namely, $c_{i,\text{col}} = c_v \cdot c_{\pi,i}$, where $c_{i,\text{col}}$ is the

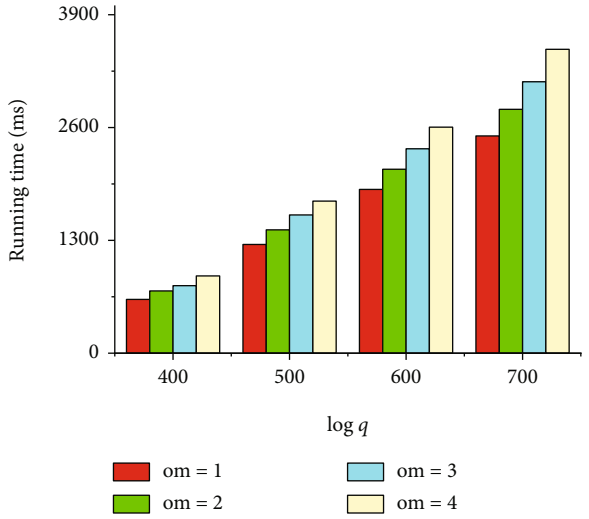


Figure 5: The efficiency of our homomorphic maximum protocol ($k = 7$).

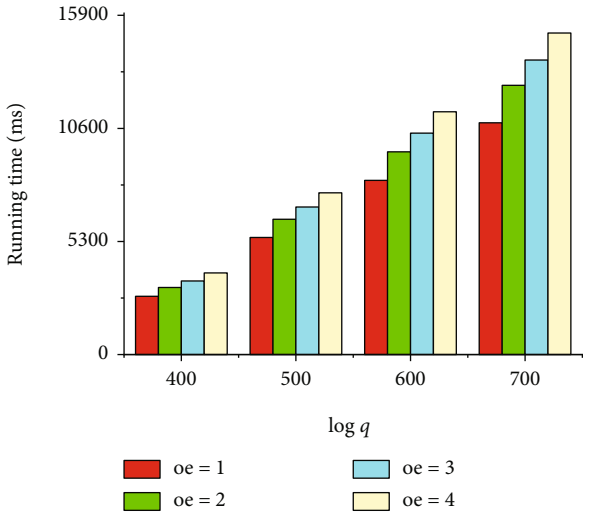


Figure 6: The efficiency of our homomorphic exponential protocol ($n = 2$).

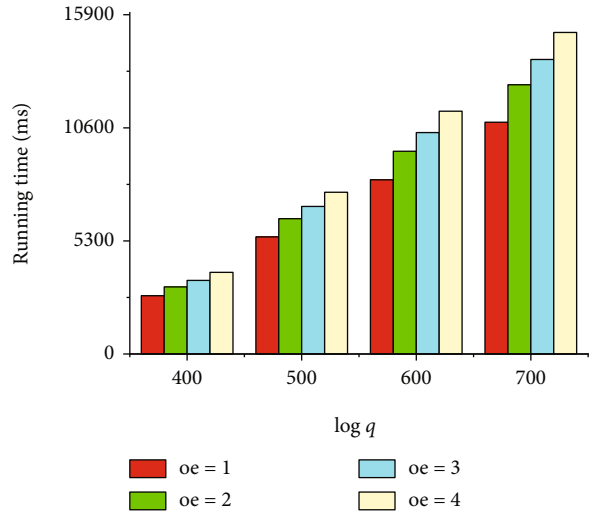


Figure 7: The efficiency of our homomorphic exponential protocol ($n = 3$).

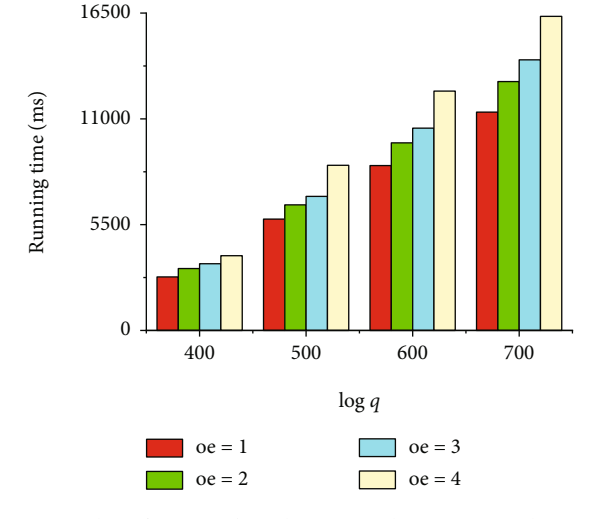


Figure 8: The efficiency of our homomorphic exponential protocol ($n = 4$).

ciphertext of $V(\mathcal{S}_{\text{col}}; \theta_v)\pi(\mathcal{A}_i \mid \mathcal{S}_{\text{col}})$, $i = 0, \cdots, \chi - 1$. Set index = 0. Finally, the cloud server computes the ciphertext $c_{\text{hor,col}}$ by using the homomorphic maximum protocol $\text{argmax}(c_0, \cdots, c_{\chi-1})$. Set index = hor. Hence, the treatment decision is $c_{dr} = c_{\mathcal{A}_{\text{index}}}$. Then, c_{dr} will be transmitted to the undiagnosed patient. In order to obtain the treatment decision dr, c_{dr} can be decrypted by using his own secret key.

In order to better understand Algorithm 8, we give an example, which is described as follows. In this example, we set $\chi = 4$, $\varphi = 4$, $\theta = 0.5$, $\theta_v = 0.5$, and $t_{\max} = 3$. We suppose that $(c_{\mathcal{S}_0}, \cdots, c_{\mathcal{S}_3})$ are ciphertexts of (0.1,0.2,0.3,0.4), respectively. $(c_{\mathcal{A}_0}, \cdots, c_{\mathcal{A}_3})$ are ciphertexts of (0.1,0.2,0.3,0.4), respectively. For the convenience of computation, we let $x(c_{\mathcal{S}_j}) = c_{\mathcal{S}_j}$, $f(c_{\mathcal{A}_i} \mid c_{\mathcal{S}_j}) = c_{\mathcal{A}_i} c_{\mathcal{S}_j}$, $\pi(\mathcal{A}_i \mid \mathcal{S}_j) = e^{f(\mathcal{A}_i \mid \mathcal{S}_j)}$, $i = 0, 1, 2, 3$, $j = 0, 1, 2, 3$. The calculation of $c_{i,j} = V(c_{\mathcal{S}_j}; \theta_v)\pi(c_{\mathcal{A}_i} \mid c_{\mathcal{S}_j})$ is the key component for the execution of Algorithm 8, where the corresponding plaintext of $c_{i,j}$ is $V(\mathcal{S}_j; \theta_v)\pi(\mathcal{A}_i \mid \mathcal{S}_j)$. In this example, the distribution of $c_{i,j}$ can be shown in Table 3.

If the patient requires a diagnostic service, the encrypted current state c_x is input for the implementation of Algorithm 8. Then, $c_{\mathcal{S}_{be}}$ is compared with c_x by the execution of the protocol $c_b = \text{comp}(c_{\mathcal{S}_{be}}, c_x)$, where $be = 0$. If the comparison result $b = t$, set col = be. If the comparison result $b \neq t$, the next element $c_{\mathcal{S}_{be+1}}$ will be compared with c_x until $be + 1 > 3$. We suppose that $x = \mathcal{S}_1$, namely, col = 1. Hence, $c_v = x(c_{\mathcal{S}_1})\theta_v = x(c_{\mathcal{S}_1})0.5$, where the plaintext of c_v is $0.2 \times 0.5 = 0.1$. We compute ciphertexts $c_{\pi,0}$, $c_{\pi,1}$, $c_{\pi,2}$, and $c_{\pi,3}$, which corresponding plaintexts are $e^{0.01}$/temp, $e^{0.02}$/temp, $e^{0.03}$/temp, and $e^{0.04}$/temp, respectively, where temp = $e^{0.01} + e^{0.02} + e^{0.03} + e^{0.04}$. Then, we compute ciphertexts $c_{0,1}$, $c_{1,1}$, $c_{2,1}$,

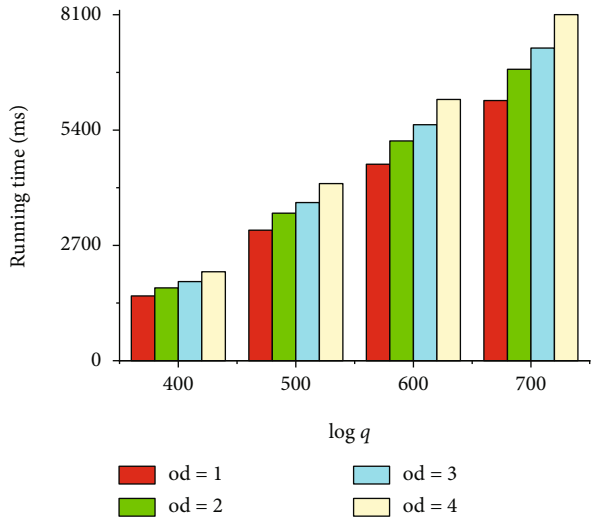


Figure 9: The efficiency of our homomorphic division protocol.

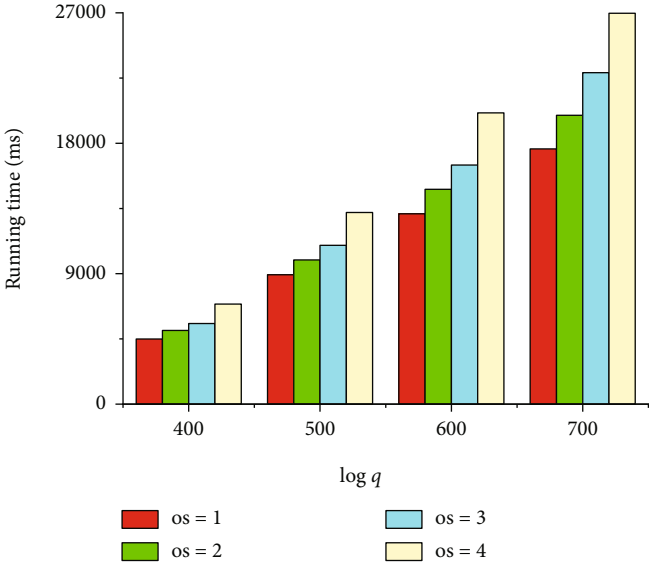


Figure 10: The efficiency of our homomorphic reciprocal of square root protocol.

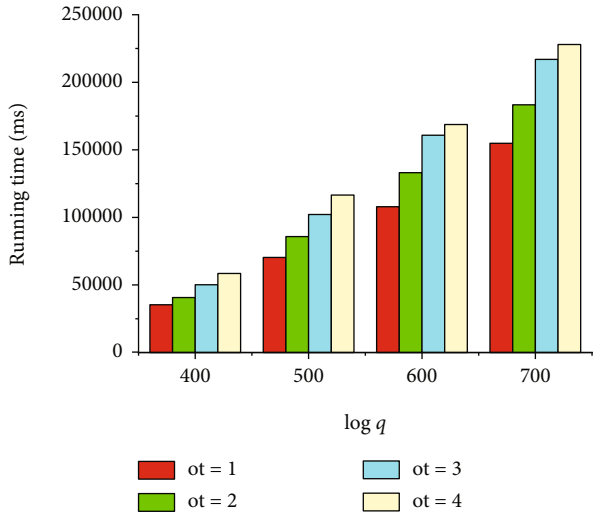


Figure 11: The efficiency of our secure A3C reinforcement learning algorithm.

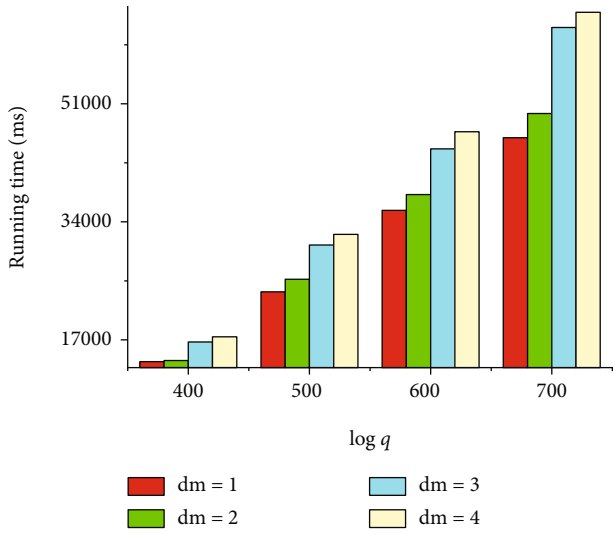


Figure 12: The efficiency of our secure treatment decision-making algorithm.

and $c_{3,1}$, which corresponding plaintexts are $0.1e^{0.01}$/temp, $0.1e^{0.02}$/temp, $0.1e^{0.03}$/temp, and $0.1e^{0.04}$/temp, respectively. Based on the execution of the protocol $\text{argmax}(c_{0,1}, c_{1,1}, c_{2,1}, c_{3,1})$, we can obtain the output ciphertext is $c_{3,1}$. The encrypted treatment decision is $c_{\mathscr{A}_3}$, which corresponding plaintext is $\mathscr{A}_3$.

7. Performance Results

In this section, based on Cheon et al.'s homomorphic encryption scheme, we analyze the efficiency of our secure computation protocols, secure A3C reinforcement learning algorithm, and secure treatment decision-making algorithm. We use the virtual machine to implement experiments without the GPU hardware platform. In our experimental environment, the operating system is macOS 10.14.6. Our personal computer has two Intel (R) Core (TM) i5 CPU processors, which runs at 2.3 GHz with 8.00 GB RAM. The operation system of a virtual machine is ubuntu 16.04. The virtual machine is allocated single Intel (R) Core (TM) i5 CPU processor with 1.0 GB RAM. In order to implement high-level numeric algorithms, we choose the NTL library. We use the GCC platform to compile our C++ codes. We adopt the UC Irvine Machine Learning Repository (http://archive.ics.uci.edu/ml/index.php) for implementing the experiments. For convenience, we set $\log q$ ranging from 400 to 700, the scaling factor $\log p = 30$.

Figure 2 shows the efficiency of our homomorphic comparison protocol, where the number of comparison *oc* ranges from 1 to 4. As shown in Figure 2, the running time of our homomorphic comparison protocol increases significantly with the increasing of *oc*. Figures 3–5 show the efficiency of our homomorphic maximum protocol, where the number

of maximum *om* ranges from 1 to 4 and the number of plaintexts *k* ranges from 5 to 7. It can be easily observed that the running time of our homomorphic maximum protocol increases significantly with the increasing of *k* and *om*. Figures 6–8 show the efficiency of our homomorphic exponential protocol, where the number of exponential operation *oe* ranges from 1 to 4 and the integer *n* ranges from 2 to 4. We can observe that the running time of our homomorphic exponential protocol increases rapidly with the increasing of *oe* and *n*.

Then, Figure 9 shows the efficiency of our homomorphic division protocol, where the number of division *od* ranges from 1 to 4. We can observe the changing trend of the running time of our homomorphic division protocol. This protocol has an obvious growth of running time with the increasing of *od* and $\log q$. Figure 10 shows the efficiency of our homomorphic reciprocal of square root protocol, where *os* ranges from 1 to 4; *os* denotes the number of operations of reciprocal of square root. With the increasing of *os* and $\log q$, more running time is needed for implementing our homomorphic reciprocal of square root protocol. It can be observed that its running time is longer than the above homomorphic comparison, maximum, exponential, and division protocols. Figure 11 shows the efficiency of our secure A3C reinforcement learning algorithm, where *ot* ranges from 1 to 4; *ot* denotes the number of operations of A3C training algorithm. With the increasing of *ot* and $\log q$, our A3C reinforcement learning algorithm requires more running time. This algorithm is responsible for training the parameters θ and θ_v. Hence, this algorithm is complicated. We can observe too much running time is needed for this algorithm, which can demonstrate the above viewpoint. Figure 12 shows the efficiency of our secure treatment decision-making algorithm, where *dm* ranges from 1 to 4; *dm* denotes the number of operations of treatment decision-making algorithm. The running time of this algorithm grows with the increasing of *dm*. This algorithm uses the optimized θ and θ_v. Hence, this algorithm is less complicated than the secure A3C algorithm. The running time of this algorithm is shorter than the secure A3C algorithm, which can verify the above viewpoint. In a conclusion, the above efficiency analysis shows the feasibility of our secure computation protocols and algorithms.

8. Conclusion

Reinforcement learning is helpful for implementing dynamic treatment regimes on health data. However, private health data may be illegally leaked, falsified, or deleted in the execution of the reinforcement learning algorithm. Hence, we study secure dynamic treatment regimes on health data. In this paper, we have designed homomorphic comparison protocol, homomorphic maximum protocol, homomorphic exponential protocol, homomorphic division protocol, and homomorphic reciprocal of square root protocol. Based on these secure computation protocols, we have proposed a privacy-preserving A3C reinforcement learning algorithm for the first time. Then, it is used for implementing the secure treatment decision-making algorithm. Finally, we simulate the proposed secure computation protocols and algorithms. Simulation results show that our secure computation protocols and algorithms are feasible.

In the future research, we will use homomorphic encryption to implement other machine learning algorithms, such as distributed learning [37] and federated reinforcement learning [38], which can successfully dominate multiple real devices that have the same type and slightly different dynamics. In addition, we plan to evaluate the performance of the secure A3C algorithm in other real-world scenarios, for example, vehicular ad hoc network.

Data Availability

The data of secure computation protocols and algorithms used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the Science and Technology Innovation Projects of Shenzhen (JCYJ20190809152003992), Shenzhen Science and Technology Program (JCYJ20210324100813034), the Guangdong Basic and Applied Basic Research Foundation (2020A1515110496), and the College-Enterprise Collaboration Project of Shenzhen Institute of Information Technology (11400-2021-010201-010199).

References

[1] I. M. Tayler and R. S. Stowers, "Engineering hydrogels for personalized disease modeling and regenerative medicine," *Acta Biomaterialia*, vol. 132, pp. 4–22, 2021.

[2] X. Liu, R. Deng, K. K. Raymond Choo, and Y. Yang, "Privacy-preserving reinforcement learning design for patient-centric dynamic treatment regimes," *IEEE Transactions on Emerging Topics in Computing*, vol. 9, no. 1, pp. 456–470, 2021.

[3] Y. Liu, B. Logan, N. Liu, Z. Xu, J. Tang, and Y. Wang, "Deep reinforcement learning for dynamic treatment regimes on medical registry data," in *Proceedings of 2017 IEEE International Conference on Healthcare Informatics*, pp. 380–385, Park City, UT, USA, 2017.

[4] R. S. Sutton and A. G. Barto, "Reinforcement learning," *A Bradford Book*, vol. 15, no. 7, pp. 665–685, 1998.

[5] L. Liu, C. Chen, Q. Pei, S. Maharjan, and Y. Zhang, "Vehicular edge computing and networking: a survey," *Mobile Networks and Applications*, vol. 26, no. 3, pp. 1145–1168, 2021.

[6] X. Sun, F. R. Yu, and P. Zhang, "A survey on cyber-security of connected and autonomous vehicles (CAVs)," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–20, 2021.

[7] T. Yang, L. Kong, N. Zhao, and R. Sun, "Efficient energy and delay tradeoff for vessel communications in SDN based maritime wireless networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 6, pp. 3800–3812, 2021.

[8] T. Yang, M. Qin, N. Cheng, W. Xu, and L. Zhao, "Liquid Software-Based Edge Intelligence for Future 6G Networks," *IEEE Network*, 2021.

[9] T. Yang, J. Chen, and N. Zhang, "AI-empowered maritime internet of things: a parallel-network-driven approach," *IEEE Network*, vol. 34, no. 5, pp. 54–59, 2020.

[10] L. Liu, J. Feng, Q. Pei et al., "Blockchain-enabled secure data sharing scheme in mobile-edge computing: an asynchronous advantage actor–critic learning approach," *IEEE Internet of Things Journal*, vol. 8, no. 4, pp. 2342–2353, 2020.

[11] X. Sun, F. R. Yu, P. Zhang, Z. Sun, W. Xie, and X. Peng, "A survey on zero-knowledge proof in blockchain," *IEEE Network*, vol. 35, no. 4, pp. 198–205, 2021.

[12] S. Mao, J. Wu, L. Liu, D. Lan, and A. Taherkordi, "Energy-efficient cooperative communication and computation for wireless powered mobile-edge computing," *IEEE Systems Journal*, pp. 1–12, 2020.

[13] C. Cimpanu, "Amca data breach has now gone over the 20 million mark," 2019, https://www.zdnet.com/article/amca-data-breach-has-nowgone-over-the-20-million-mark/.

[14] W. Zhang, M. Li, R. Tandon, and H. Li, "Online location trace privacy: an information theoretic approach," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 1, pp. 235–250, 2019.

[15] R. L. Rivest, L. Adleman, and M. L. Dertouzos, "On data banks and privacy homomorphisms," *Foundations of Secure Computation*, vol. 4, no. 11, pp. 169–180, 1978.

[16] J. H. Cheon, A. Kim, M. Kim, and Y. Song, "Homomorphic encryption for arithmetic of approximate numbers," in *Proceedings of Advances in Cryptology - ASIACRYPT 2017*, pp. 409–437, Hong Kong, China, 2017.

[17] H. Kim and W. Lee, "Real-time path planning through Q-learning's exploration strategy adjustment," in *Proceedings of 2021 International Conference on Electronics, Information, and Communication*, pp. 1–3, Jeju, Republic of Korea, 2021.

[18] C. Wu, Z. Liu, F. Liu, T. Yoshinaga, Y. Ji, and J. Li, "Collaborative learning of communication routes in edge-enabled multi-access vehicular environment," *IEEE Transactions on Cognitive Communications and Networking*, vol. 6, no. 4, pp. 1155–1165, 2020.

[19] R. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Machine Learning*, vol. 8, no. 3-4, pp. 229–256, 1992.

[20] V. Mnih, A. Badia, M. Mirza et al., "Asynchronous methods for deep reinforcement learning," in *Proceedings of the 33rd International Conference on Machine Learning*, pp. 1928–1937, New York, USA, 2016.

[21] J. Feng, F. Richard Yu, Q. Pei, X. Chu, J. du, and L. Zhu, "Cooperative computation offloading and resource allocation for blockchain-enabled mobile-edge computing: a deep reinforcement learning approach," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6214–6228, 2020.

[22] C. Gentry, "Fully homomorphic encryption using ideal lattices," in *Proceedings of the 41st annual ACM symposium on Symposium on theory of computing - STOC '09*, pp. 169–178, New York, USA, 2009.

[23] C. Gentry, A. Sahai, and B. Waters, "Homomorphic encryption from learning with errors: conceptually-simpler, asymptotically-faster, attribute-based," in *Advances in Cryptology – CRYPTO 2013*, pp. 75–92, Springer, 2013.

[24] Z. Brakerski, C. Gentry, and V. Vaikuntanathan, "(Leveled) fully homomorphic encryption without bootstrapping," *ACM Transactions on Computation Theory*, vol. 6, no. 3, pp. 1–36, 2014.

[25] V. Lyubashevsky, C. Peikert, and O. Regev, "On ideal lattices and learning with errors over rings," in *Advances in Cryptology – EUROCRYPT 2010*, pp. 1–23, Springer, 2010.

[26] A. Khedr and G. Gulak, "Securemed: secure medical computation using gpu-accelerated homomorphic encryption scheme," *IEEE Journal of Biomedical and Health Informatics*, vol. 22, no. 2, pp. 597–606, 2018.

[27] X. Sun, P. Zhang, M. Sookhak, J. Yu, and W. Xie, "Utilizing fully homomorphic encryption to implement secure medical computation in smart cities," *Personal and Ubiquitous Computing*, vol. 21, no. 5, pp. 831–839, 2017.

[28] N. Dowlin, R. Gilad-Bachrach, K. Laine, K. Lauter, M. Naehrig, and J. Wernsing, "Manual for using homomorphic encryption for bioinformatics," *Proceedings of the IEEE*, vol. 105, no. 3, pp. 1–16;, 2017.

[29] D. Boneh, E.-J. Goh, and K. Nissim, "Evaluating 2-DNF formulas on ciphertexts," in *Proceedings of Theory of Cryptography Conference*, pp. 325–341, Cambridge, USA, 2005.

[30] A. Poon, S. Jankly, and T. Chen, "Privacy preserving Fishers exact test on genomic data," in *Proceedings of 2018 IEEE International Conference on Big Data*, pp. 2546–2553, Seattle, USA, 2018.

[31] J. L. Raisaro, G. Choi, S. Pradervand et al., "Protecting privacy and security of genomic data in i2b2 with homomorphic encryption and differential privacy," *IEEE/ACM Transactions on Computational Biology & Bioinformatics*, vol. 15, no. 5, pp. 1413–1426, 2018.

[32] X. Liu, R. H. Deng, K.-K. R. Choo, and J. Weng, "An efficient privacy-preserving outsourced calculation toolkit with multiple keys," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 11, pp. 2401–2414, 2016.

[33] J. Fan and F. Vercauteren, "Somewhat practical fully homomorphic encryption," 2012, https://eprint.iacr.org/2012/144.pdf.

[34] L. Jiang, L. Chen, T. Giannetsos, B. Luo, K. Liang, and J. Han, "Toward practical privacy-preserving processing over encrypted data in IoT: an assistive healthcare use case," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 10177–10190, 2019.

[35] X. Sun, P. Zhang, J. K. Liu, J. Yu, and W. Xie, "Private machine learning classification based on fully homomorphic encryption," *IEEE Transactions on Emerging Topics in Computing*, vol. 8, no. 2, pp. 352–364, 2020.

[36] C. Lomont, "Fast inverse square root," 2003, http://lomont.org/papers/2003/InvSqrt.pdf.

[37] X. Chen, C. Wu, Z. Liu, N. Zhang, and Y. Ji, "Computation offloading in beyond 5g networks: a distributed learning framework and applications," *IEEE Wireless Communications*, vol. 28, no. 2, pp. 56–62, 2021.

[38] H.-K. Lim, J.-B. Kim, C.-M. Kim, G.-Y. Hwang, H.-b. Choi, and Y.-H. Han, "Federated reinforcement learning for controlling multiple rotary inverted pendulums in edge computing environments," in *Proceedings of 2020 International Conference on Artificial Intelligence in Information and Communication*, pp. 463-464, Durban, South Africa, 2020.