Blockchain-enabled Data Collection and Sharing for Industrial IoT with Deep Reinforcement Learning

Chi Harold Liu, Qiuxia Lin, Shilin Wen

Seoul National University of Science and Technology

Computer Science and Engineering

Advanced in Blockchain Technology

Seonghyeon Gong

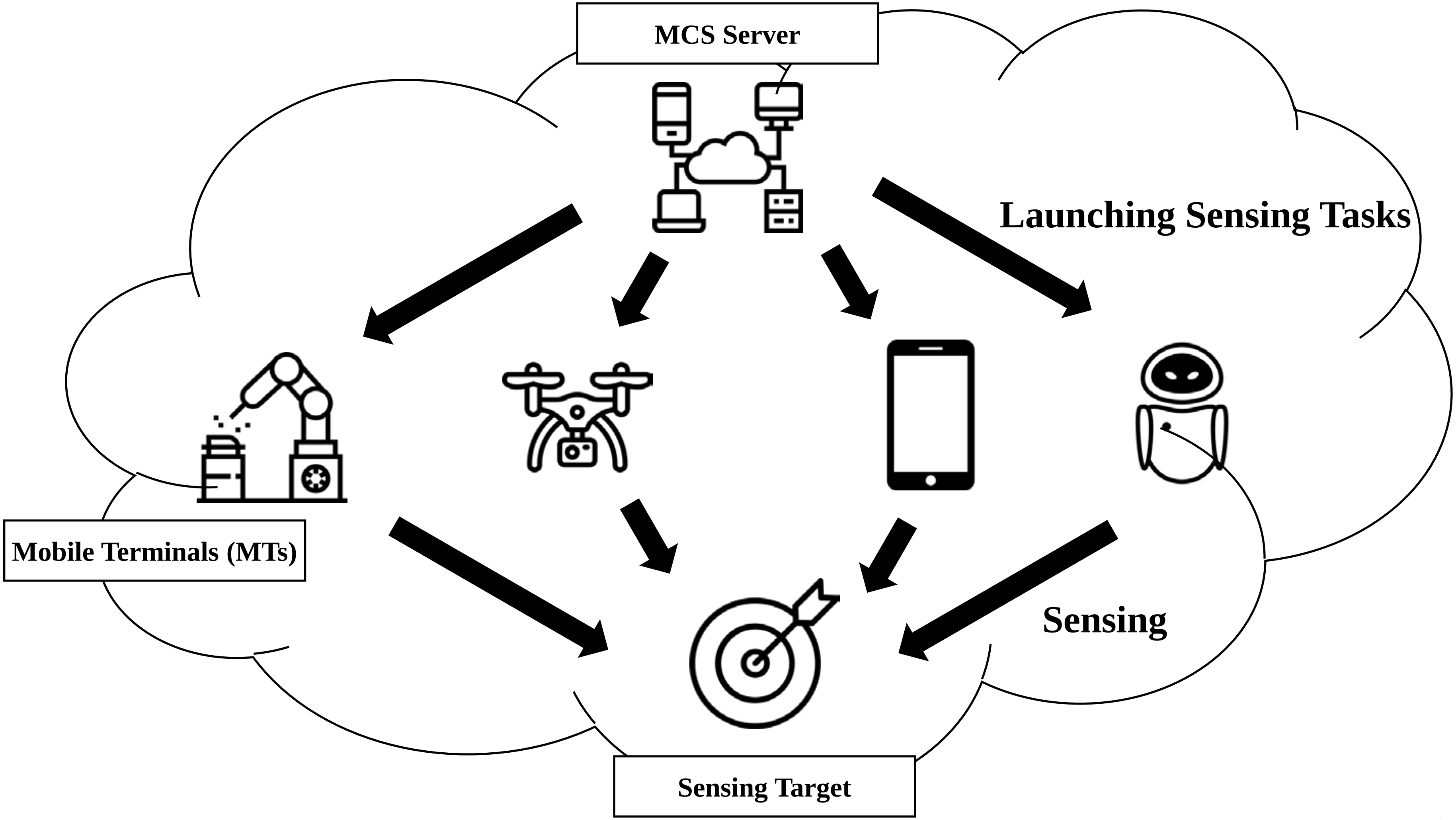
2019-04-09

Contents

1. Introduction

Concepts and Challenges of MCS system in IIoT

Rapid development of smart portable **mobile terminals (MTs)**, which are equipped with rich set of sensors, has facilitated a new type of data collection method for industrial IoT (IIoT), namely **Mobile Crowdsensing (MCS)**.



1. Introduction

Concepts and Challenges of MCS system in IIoT

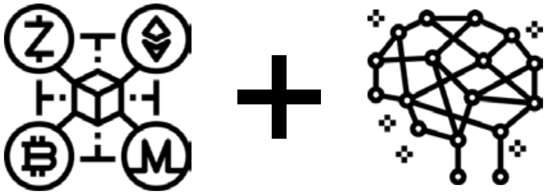
Benefits of MCS in IIoT

- ✓ mobile and scalable measures are provided
- ✓ new areas can be monitored without the need for additional dedicated devices to be installed
- ✓ subjective assessments can be easily and cost-effectively collected
- ✓ human wisdom can be straightforwardly integrated into machine intelligence
- ✓ information and decision-making processes can be shared among the whole industrial community

Main Challenges of MCS in IIoT

- ❖ How to achieve high quality data collection with **limited MT energy resource** and sensing range
- ❖ How to **ensure security** when sharing and exchanging data among MTs

Proposed framework:

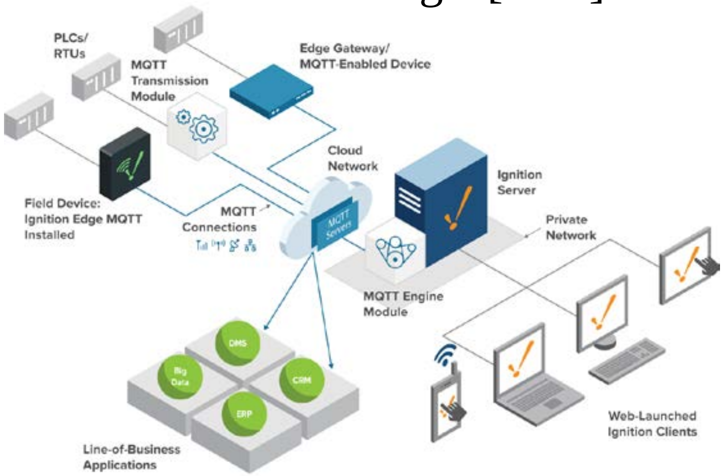


- energy efficient data collection and secure data sharing among MTs
- enabled by **blockchain** and **DRL**(Deep Reinforcement Learning)
  - DRL for achieving the maximum data collection ratio and geographic fairness
  - Blockchain for data security and reliability

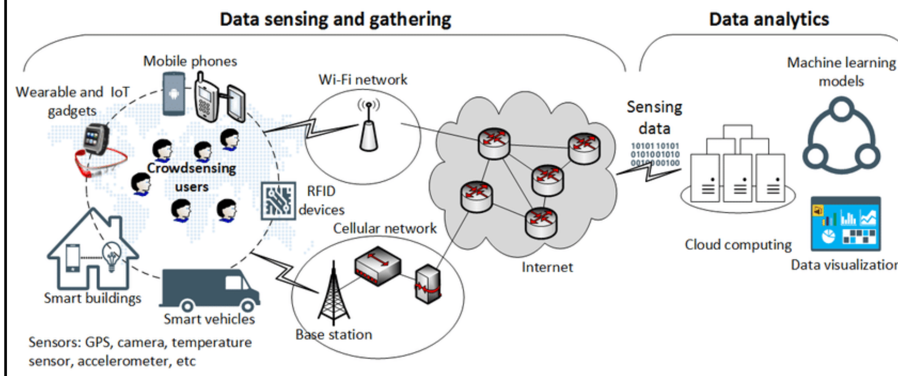
2. Related Work

overview of related works from four area: IIoT, MCS, Blockchain, DRL

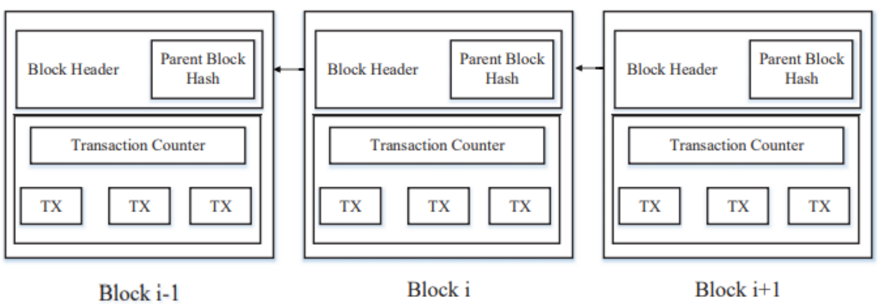
Industrial Internet of Things: [9-11]



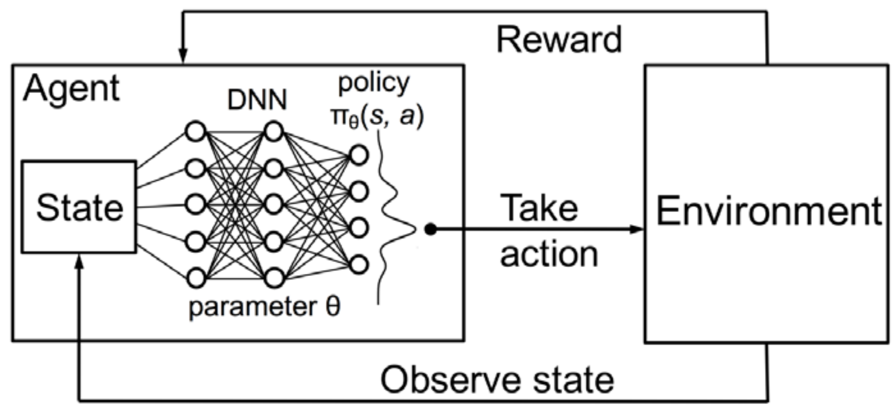
Mobile Crowd Sensing: [12-23]



Blockchain: [24-29]



Deep Reinforcement Learning: [30-31]



3. System Model

system model that combines both blockchain and DRL for efficient data collection and secure sharing

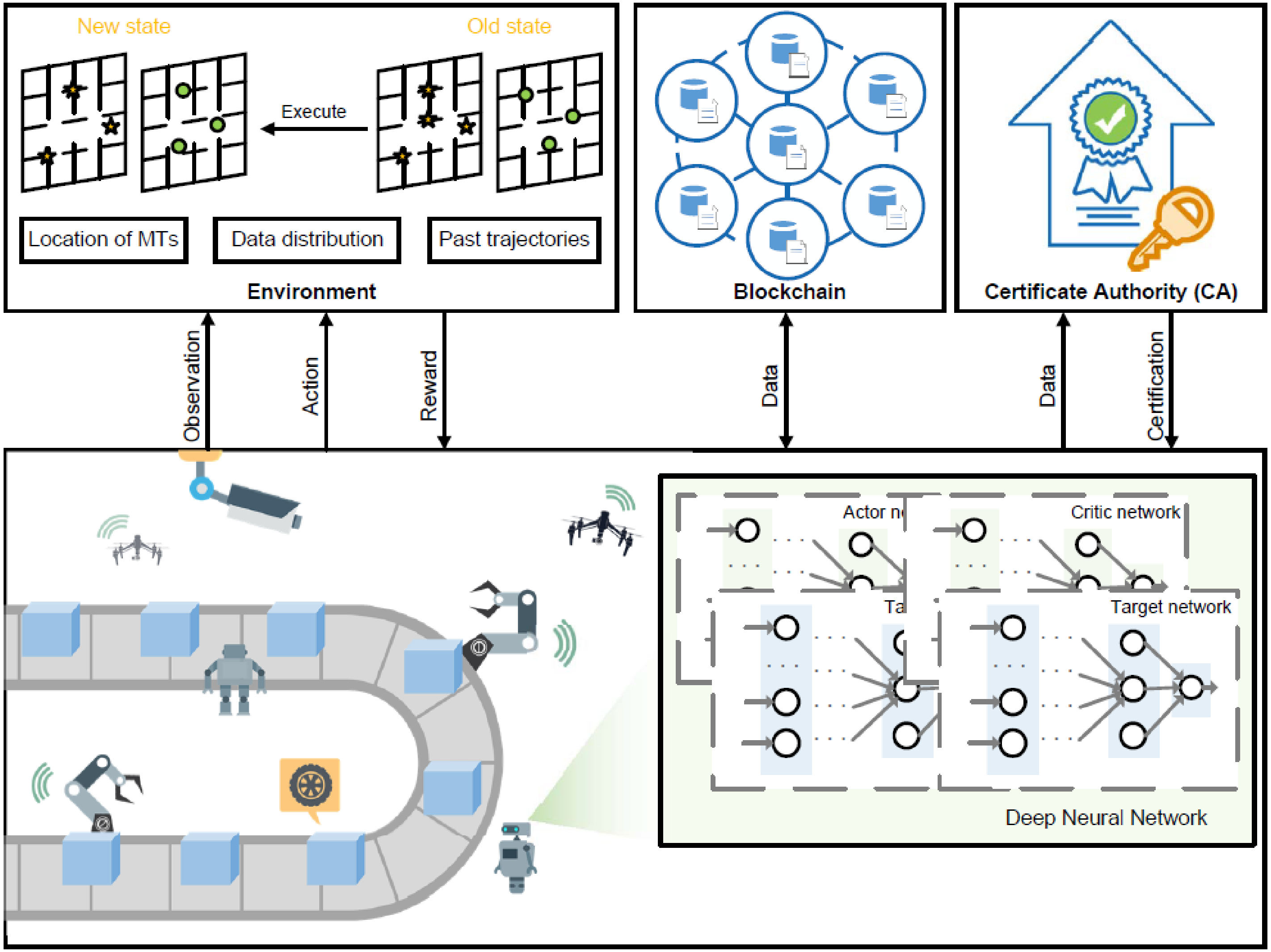
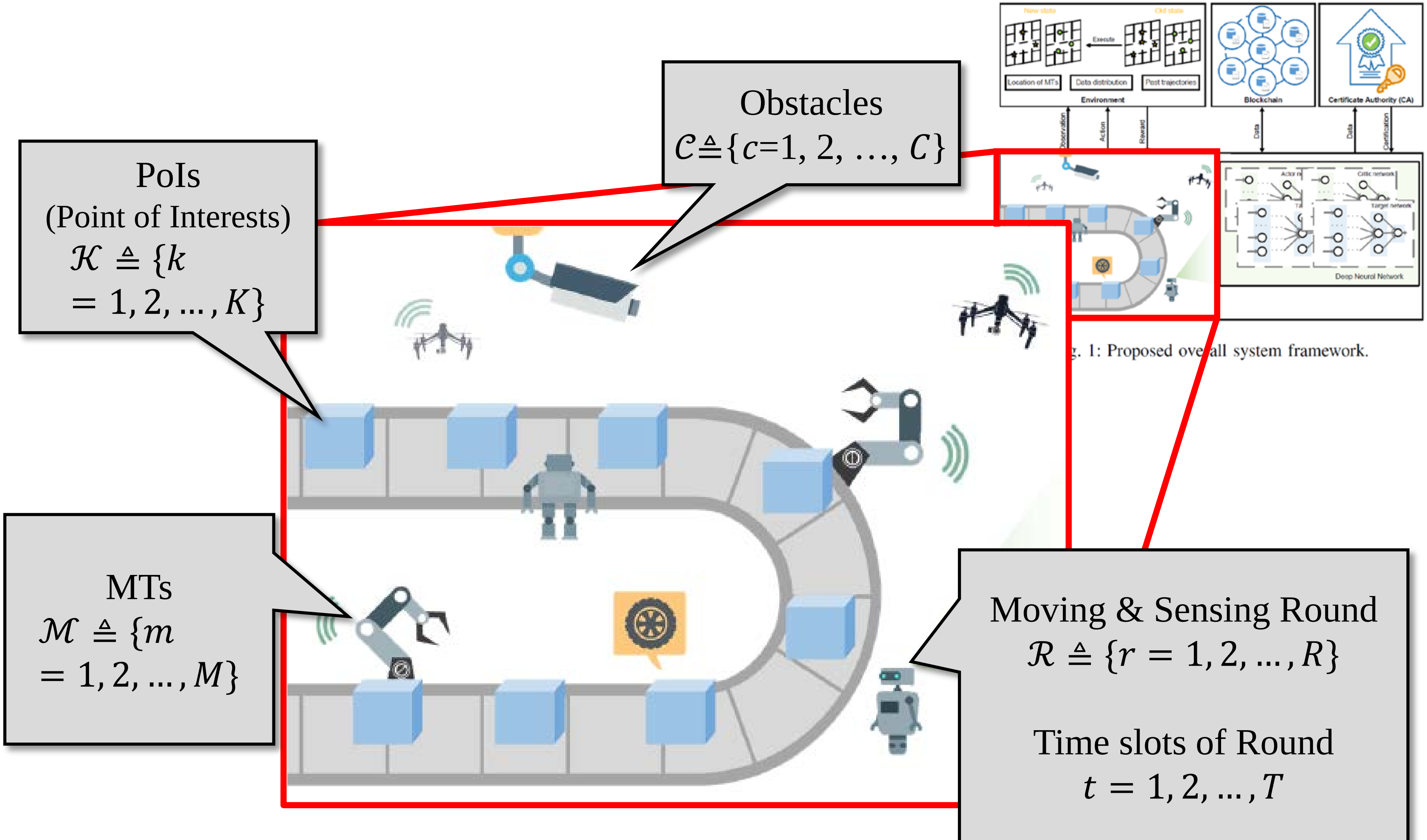


Fig. 1: Proposed overall system framework.

3. System Model

IIoT for Energy-Efficient Data Collection



3. System Model

Blockchain Network for Secure Data Sharing

Blockchain Node Categories:

- Mining Nodes – used to verify data sharing transactions and compile them into blocks.
- Non-mining Nodes – only responsible for receiving and broadcasting data sharing transactions.

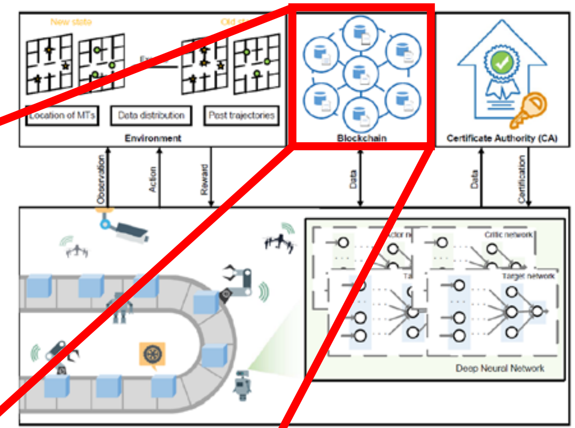
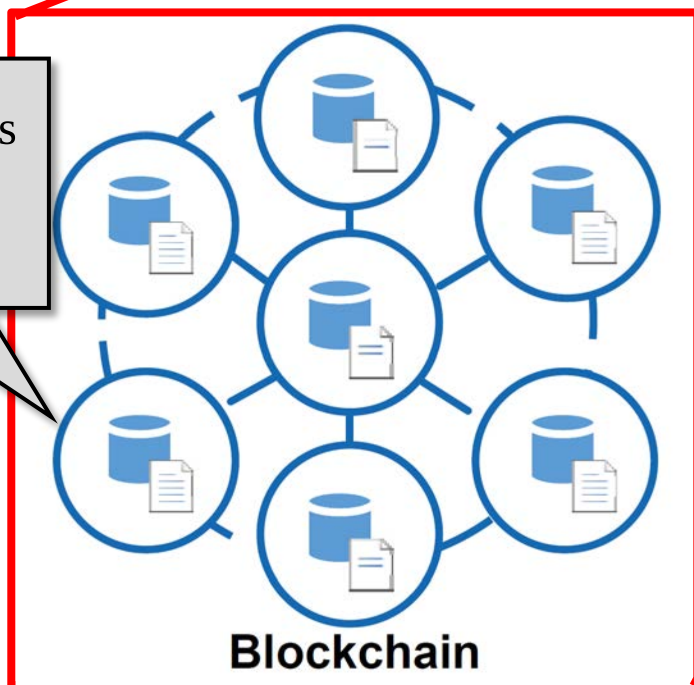


Fig. 1: Proposed overall system framework.

Ethereum Nodes
$\mathcal{N} \triangleq \{n = 1, 2, \ldots, N\}$

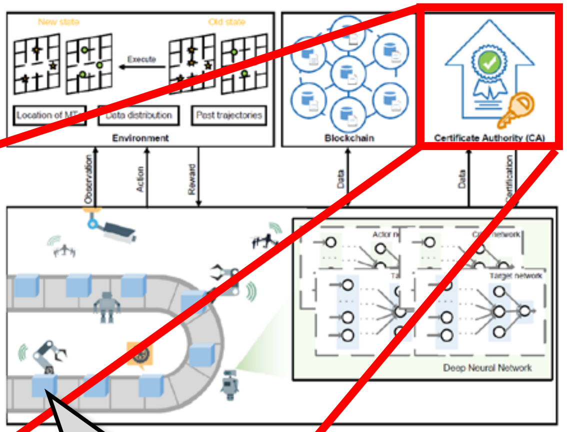


3. System Model

Blockchain Network for Secure Data Sharing

Certificate Authority (CA)

To ensure that collected data sent to the blockchain network is valid and cannot be forged, MT needs first to encrypt data with its private key and then send it to the CA.



Proposed overall system framework.

MT's Keys
(public & private)

$$(pk_m, sk_m), \forall m \in \mathcal{M}$$

4. Proposed Solution

Multi-Agent DRL based Distributed Data Collection by MTs

Traditional policy gradient based DRL (such as DQN) can only work well in a limited action space.

This work proposed new solutions.

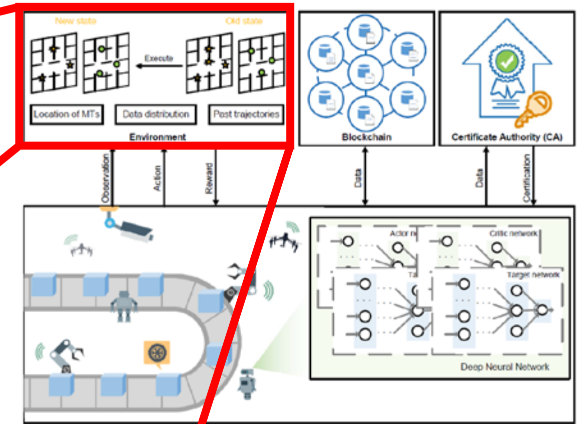


Fig. 1: Proposed overall system framework.

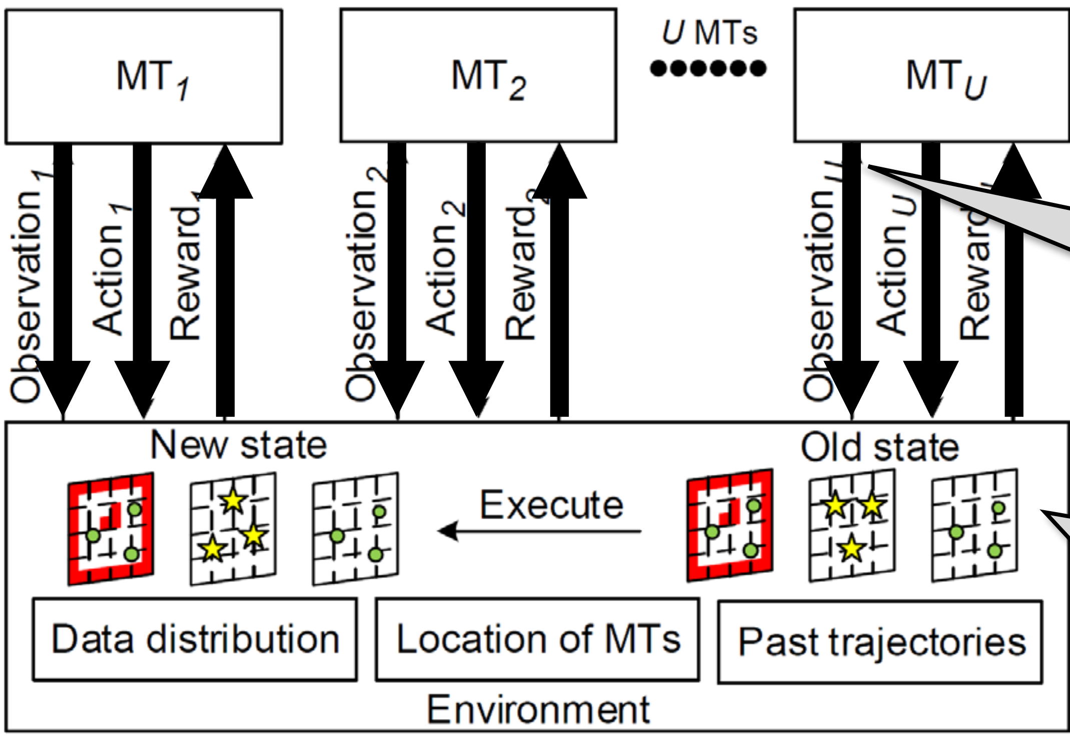


Fig. 2: Overall system flow of DRL based data collection.

Observation of MT m in state s^t of timeslot t

$$o_t^m = (x_t^m, y_t^m, e_t^m)$$

$$env\ \mathcal{S} = \{(\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3)\}$$

$$\mathcal{S}_1 = \{((x^k, y^k), (x^c, y^c))\}_{k\in\mathcal{K}, c\in\mathcal{C}}$$

$$where\ x^k, x^c \in [0, E_x]\ and\ y^k, y^c \in [0, E_y]$$

$$\mathcal{S}_2 = \{(x_t^m, y_t^m, e_t^m)\}_{m\in\mathcal{M}}$$

$$\mathcal{S}_3 = h_t(k) \in [0, T]$$

4. Proposed Solution

Multi-Agent DRL based Distributed Data Collection by MTs

Traditional policy gradient based DRL (such as DQN) can only work well in a limited action space.

This work proposed new solutions.

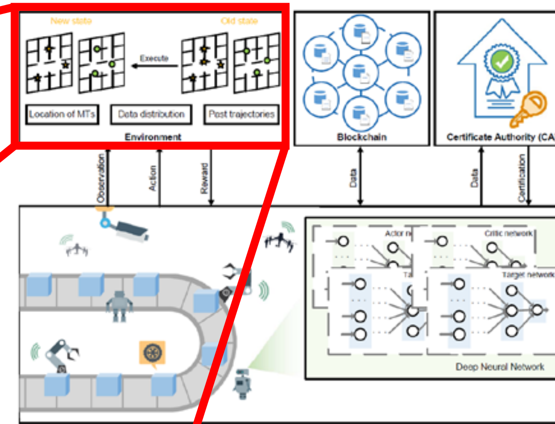


Fig. 1: Proposed overall system framework.

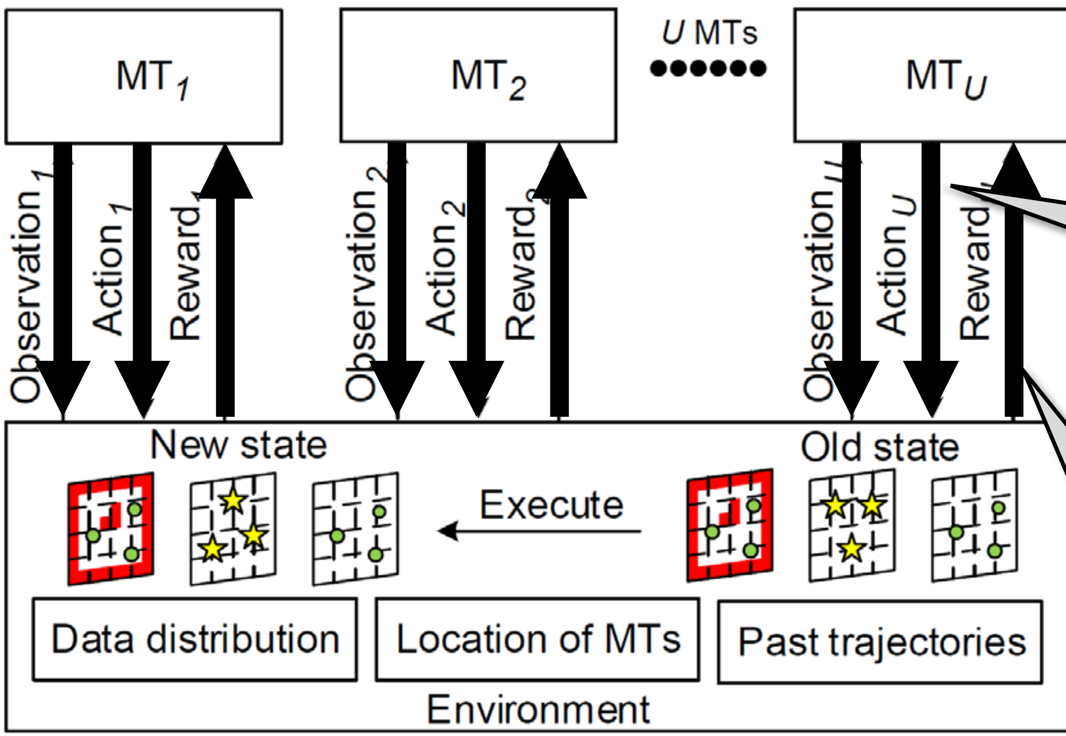


Fig. 2: Overall system flow of DRL based data collectio

Action

$$\mathcal{A} = \{(\theta_t^m, l_t^m)_{m\in\mathcal{M}} | \theta_t^m \in [0, 2\pi), l_t^m \in [0, l_{max}\}$$

geographical fairness

$$\omega_t = \frac{\left(\sum_{k=1}^{K} h_t(k)\right)^2}{K \sum_{k=1}^{K} h_t(k)^2}$$

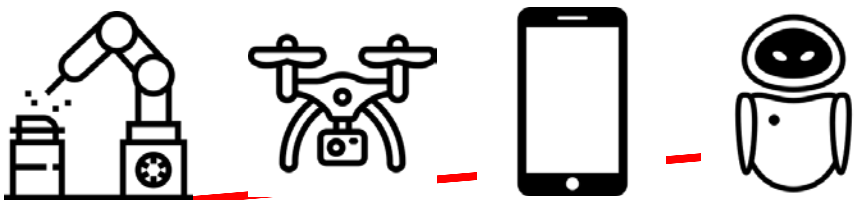
Reward

$$r_t^m = \frac{\omega_t b_t^m}{\phi(b_t^m, l_t^m)}, \forall m \in \mathcal{M}$$

4. Proposed Solution

Multi-Agent DRL based Distributed Data Collection by MTs

Each MT is implemented by 4 DNNs which serves as actor network and critic network,



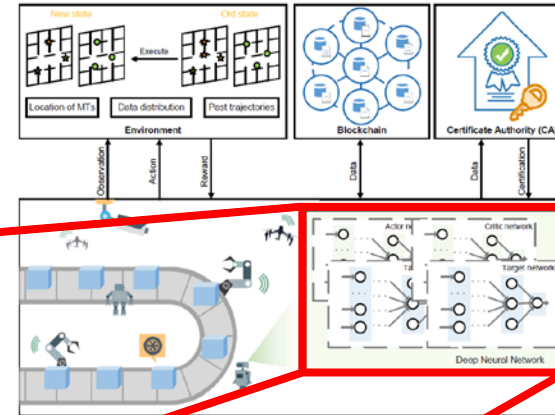
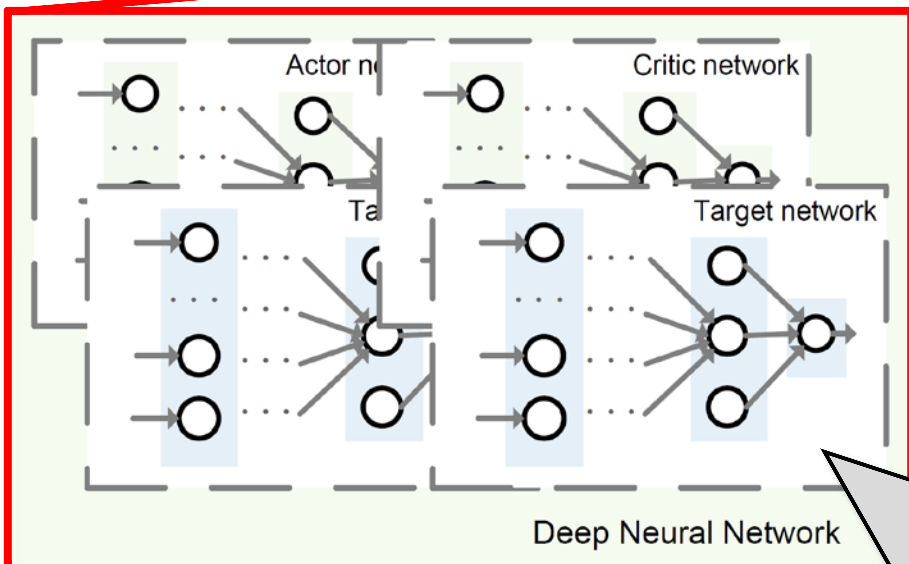


Fig. 1: Proposed overall system framework.



actor network : $\pi^m(o_t|\theta^{\pi^m})$

critic network : $Q^m(s_t, a_t|\theta^{Q^m})$

$$\theta^{Q'^m} := \tau\theta^{Q^m} + (1-\tau)\theta^{Q'^m}$$

$$\theta^{\pi'^m} := \tau\theta^{\pi^m} + (1-\tau)\theta^{\pi'^m}$$

loss function

$$L(\theta^{Q^m}) = \mathbb{E}\left[y_t^m - Q^m(s_t, a_{t+1}^1, \dots, a_{t+1}^M|\theta^{Q^m})\right],$$

$$y_t^m = \gamma Q'^m(s_t, a_{t+1}^1, \dots, a_{t+1}^M|\theta^{Q^m}) + \gamma r_{t+1}^m$$

gradient of actor network

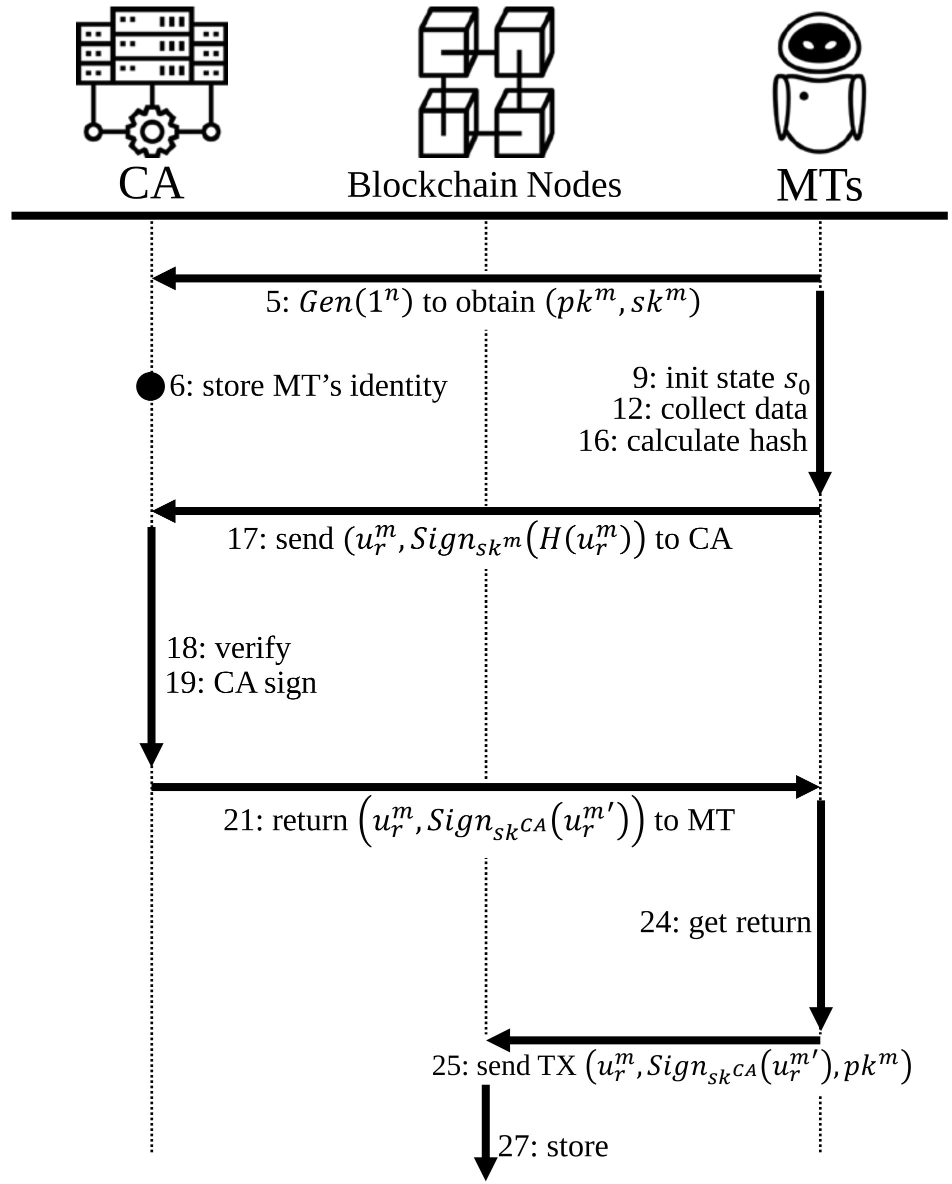
$$\nabla_{\theta^{\pi^m}} J \approx \mathbb{E}\left[\nabla_{\theta^{\pi^m}} \pi^m(o|\theta^{\pi^m})\Big|_{(o=o_t)} \nabla_a Q^m(s, a_{t+1}^1, \dots, a_{t+1}^M|\theta^{Q^m})\Big|_{a_{t+1}^m=\pi^m(o_{t+1}^m)}\right]$$

Algorithm 1 Blockchain-enabled secure data sharing among MTs

1: Initialize private blockchain, and setup N Ethereum nodes;
2: Deploy smart contract on the blockchain;
3: Blockchain starts mining;
4: **for** MT m in $\mathcal{M}$ **do**
5: Run $Gen(1^n)$ to obtain (pk^m, sk^m);
6: CA stores identity (m, pk^m);
7: **end for**
8: **for** Round $r = 1, 2, \cdots, R$ **do**
9: Initialize environment, receive initial state s_0;
10: **for** timeslot $t = 1, 2, \cdots, T$ **do**
11: **for** MT m in $\mathcal{M}$ **do**
12: Collect data (see Section III-A);
13: **end for**
14: **end for**
15: $u_r^m := \sum_{t=1}^{T} b_t^m$;
16: Hash collected data $H(u_r^m)$;
17: Send $(u_r^m, Sign_{sk^m}(H(u_r^m)))$ to CA;
18: **if** $Vrfy_{pk^m}(u_r^m || Sign_{sk^m}(H(u_r^m)) == 1$ **then**
19: **if** $u_r^m == collection[m]$ **then**
20: $u_r^{m\prime} := Sign_{sk^m}(H(u_r^m))$;
21: Return $(u_r^m, Sign_{sk^{CA}}(u_r^{m\prime}))$ to MT m;
22: **end if**
23: **end if**
24: **if** MT m receives signature from CA **then**
25: Send transaction request$(u_r^m, Sign_{sk^{CA}}(u_r^{m\prime}), pk^m$ to blockchain;
26: **if** Verify signature and identity is true **then**
27: Upload transaction to blockchain and wait for confirmation;
28: **end if**
29: **end if**
30: **end for**

4. Proposed Solution

Blockchain-enabled Secure Data Sharing among MTs



5. Security Analysis

potential attacks in the proposed system and provide solutions

- Transaction Forgery by MTs
- Eclipse Attack by Network Users
- Vulnerability Attack by Network Users
- Majority Attack by Network Users
- MT Device Failure

5. Security Analysis

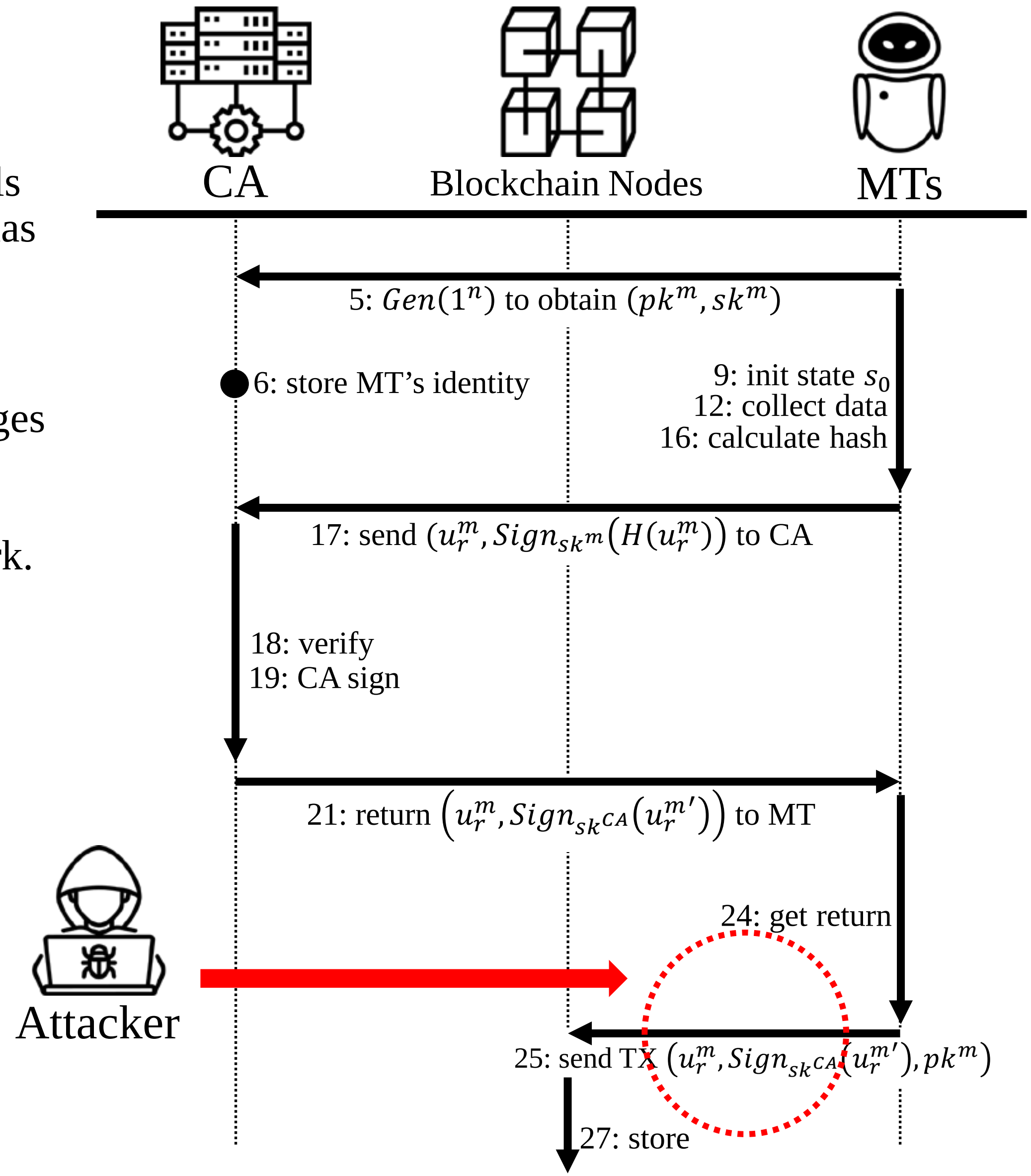
potential attacks in the proposed system and provide solutions

❖ Transaction Forgery by MTs

The attacker can intercept transaction proposals sent by legitimate user to the blockchain and has a high probability of success.

After reading the encrypted transaction messages multiple times, the attacker will try to impersonate the legitimate user to send unreal transaction proposals to the blockchain network.

We believe that it is difficult or extremely unlikely for an attacker MT to calculate the private key of CA, so it cannot forge the confirmation message of CA.



5. Security Analysis

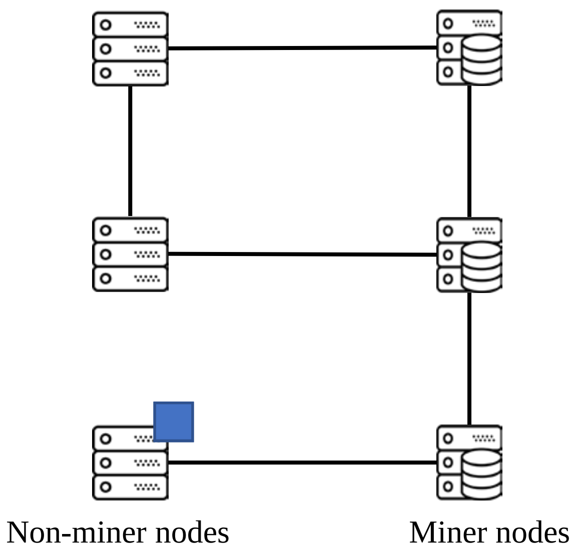
potential attacks in the proposed system and provide solutions

❖ Eclipse Attack by Network Users (Routing table poisoning attack)

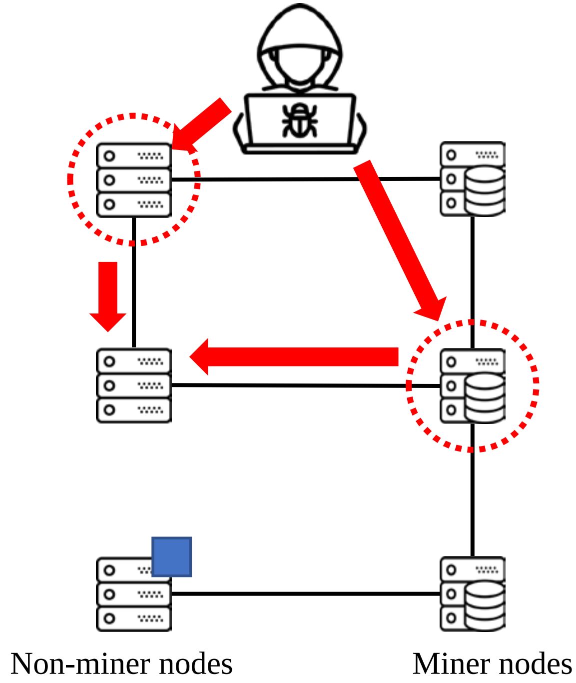
✓ an attacker can prevent victim from receiving complete information about the rest of the network.

✓ an attacker can use an Eclipse Attack to prevent Ethereum non-mining node from receiving storage and query requests to the blockchain.

✓ Ethereum has released an updated version of software, making the number of malicious nodes needed to carry out such an attack from two to thousands



Normal blockchain network



Forged network

5. Security Analysis

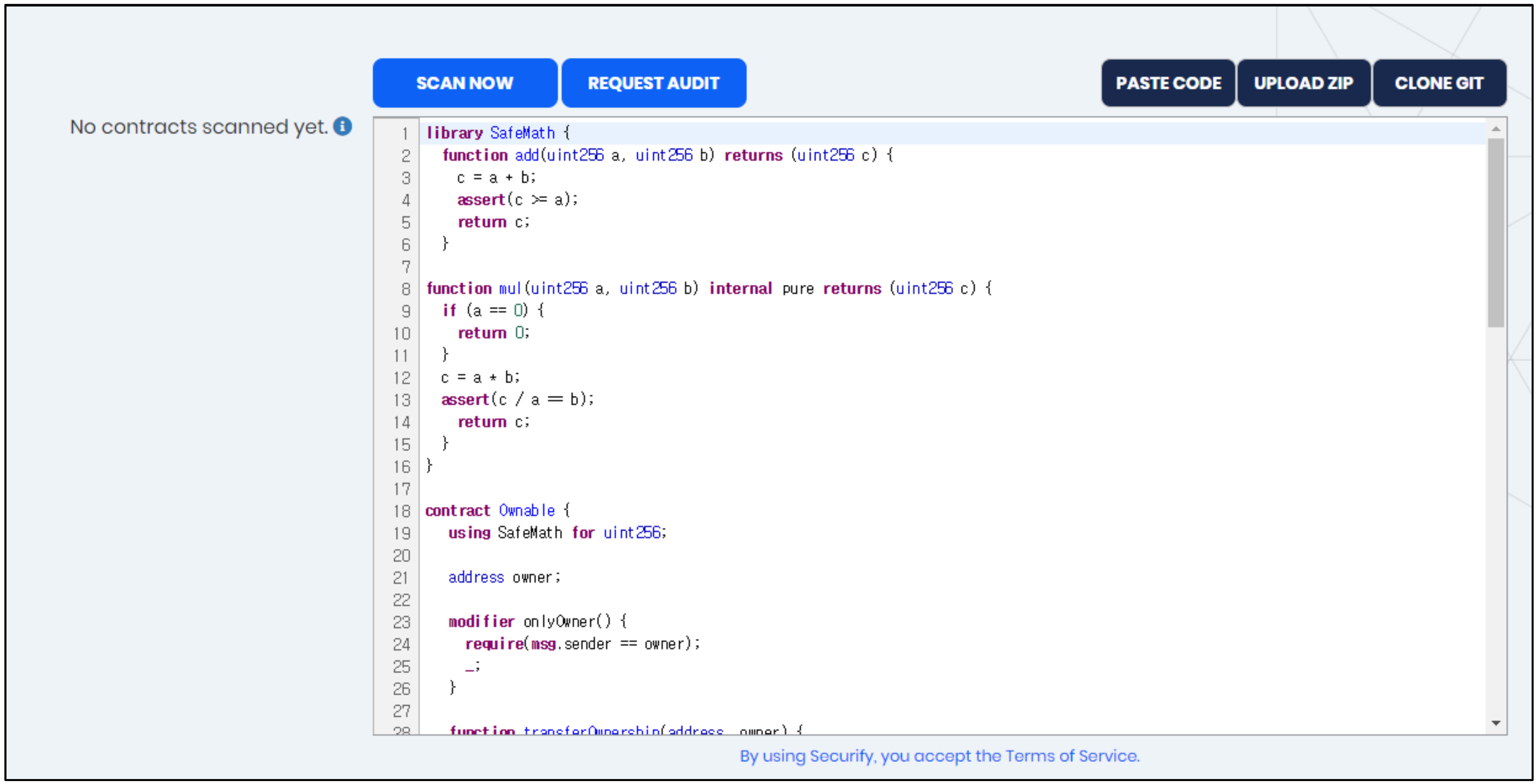
potential attacks in the proposed system and provide solutions

❖ Vulnerability Attack by Network Users

✓ All users on the blockchain can see the smart contract deployed on the blockchain.

✓ When a smart contract has a critical vulnerability, it is very easy for attackers to exploit.

✓ When designing the smart contract, we have avoided recursive calling vulnerability, timestamp dependence, arithmetic problem, return value problem, and completed code audits and security tests.



Smart contract online vulnerability check service of securify.chainsecurity.com

5. Security Analysis

potential attacks in the proposed system and provide solutions

- ❖ Majority Attack by Network Users
- ✓ If the attacker controls more than half of the computing power in the network, we can confirm that the blockchain network is not secure.
- ✓ The attacker can take advantage of the computing power to tamper with the records on the blockchain.
- ✓ The newly generated chain can belong to him/her completely, and it may not even contain any block that was mined by other miners.
- ✓ Because the longest chain is always considered to be the best credibility, the attacker can reverse the issued transaction, thus achieving double spending problem.

- ✓ Considering that our blockchain network is set up as a private chain, and the mining nodes are owned and controlled by system.

5. Security Analysis

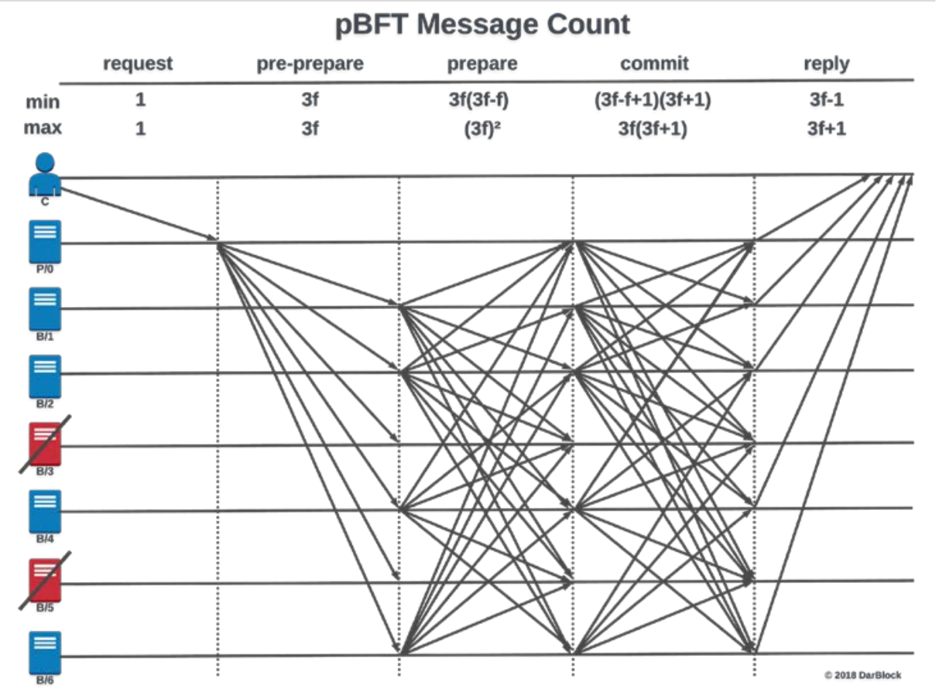
potential attacks in the proposed system and provide solutions

❖ MT Device Failure

✓ with the long-term use of these devices, some may have software or hardware problems which may prevent these devices from continuing to function normally

✓ the network there may exist malicious users that send false data to other users and this is specifically described as “The Byzantine Generals Problem”.

✓ In order to address these possible problems, certain consensus algorithms in a blockchain such as **Paxos** and **PBFT(Practical Byzantine Fault Tolerance)** have been adopted in our proposal to mitigate or overcome them.



6. Performance Evaluation

Simulation Setting

	Control Group	Experimental Group (proposed solution)	Comparison Criteria
Data Storage (DoS & DDoS)	MySQL • Windows 10 • MySQL 5.7.22 • Intel Core i7-4790 3.60GHz • RAM 12GB	Blockchain • Utuntu 16.04 LTS • Geth 1.7.2 • Inter Core i7-6700 3.4GHz • RAM 16GB • 2 mining threads	• Immediate query failure ratio • Query successful ratio
Moving Trajectories	Random	Deep reinforcement learning based	• Data collection ratio • Energy usage ratio • Geographic fairness
DDoS Attacker	Windows 10, Intel Core i7-8750 2.2GHz, RAM 8GB Windows 10, Intel Core i7-6700 3.4GHz, RAM 16GB		

6. Performance Evaluation

Simulation Setting

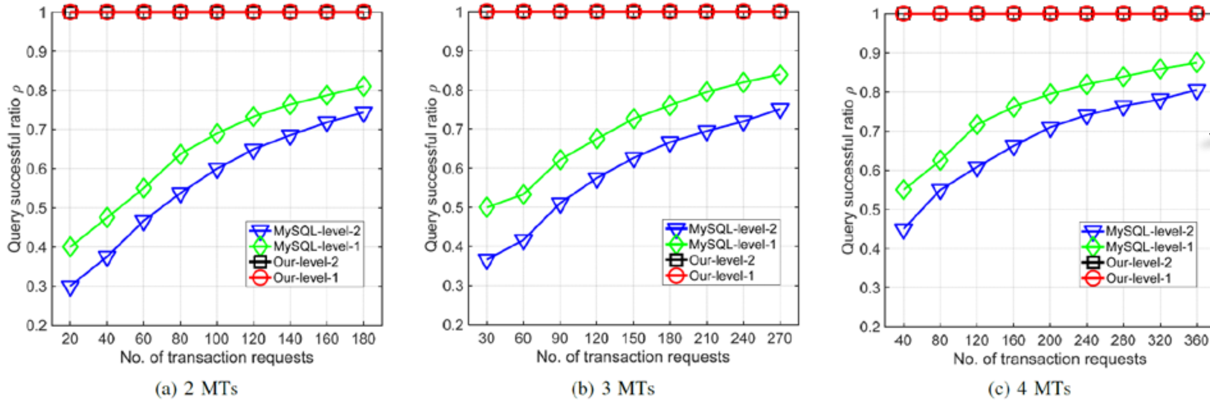


Fig. 4: Impact of DoS attack and number of MTs on query successful ratio.

Query successful ratio

Immediate query failure ratio

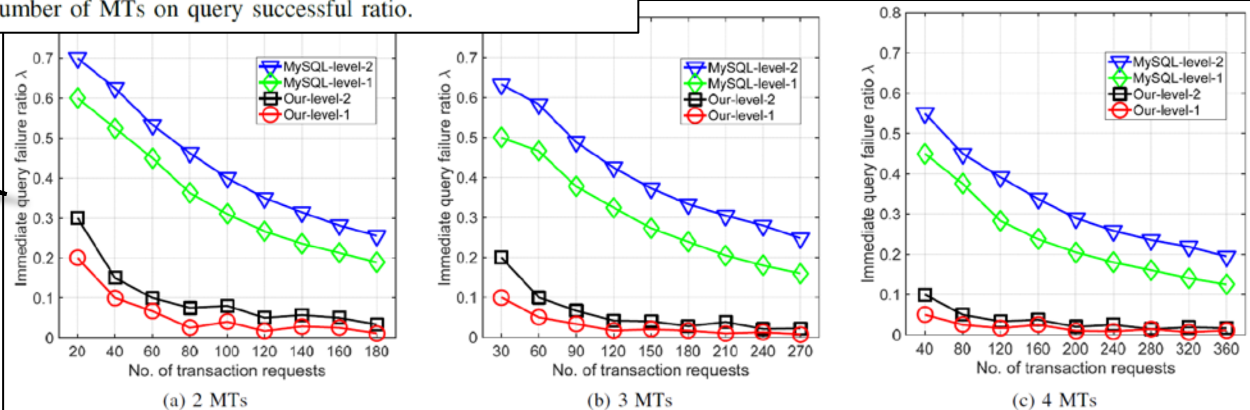


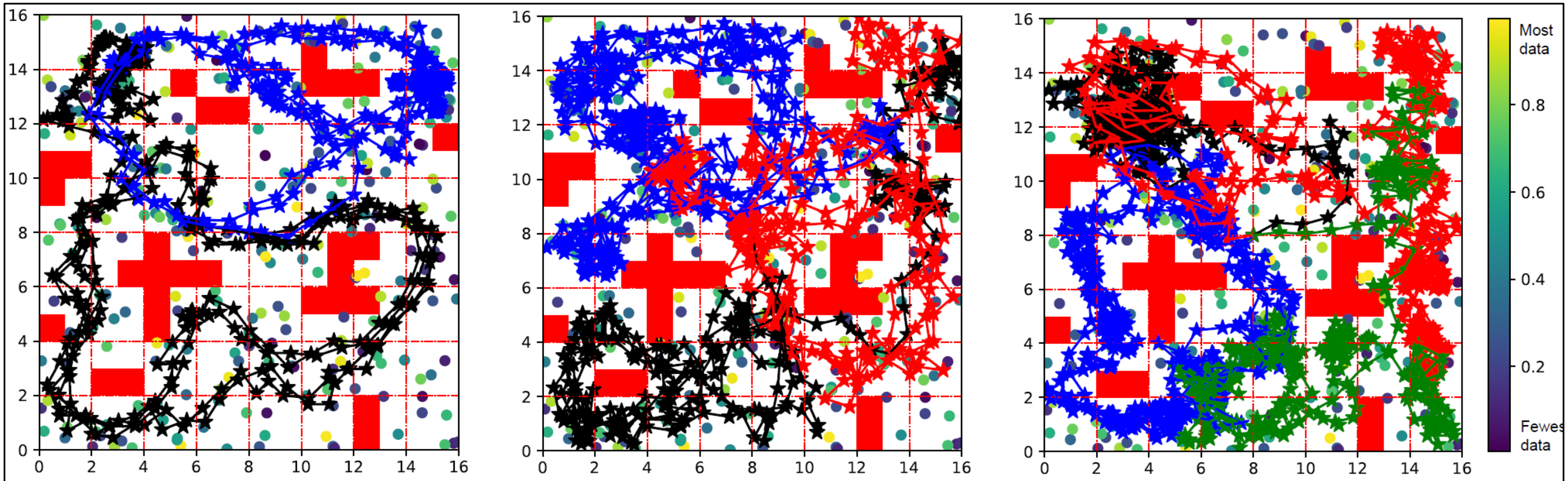
Fig. 5: Impact of DoS attack severity and number of MTs on immediate query failure ratio.

TABLE I: Results of DDoS attack.

No. of MTs		2					3					4				
No. of trans. requests		20	60	100	140	180	30	90	150	210	270	40	120	200	280	360
λ	MySQL-level-2	0.600	0.483	0.370	0.286	0.222	0.500	0.422	0.333	0.252	0.196	0.400	0.325	0.255	0.193	0.150
	MySQL-level-1	0.500	0.333	0.22	0.157	0.122	0.400	0.311	0.207	0.148	0.115	0.300	0.225	0.175	0.129	0.100
	Ours-level-2	0.200	0.100	0.030	0.014	0.022	0.167	0.067	0.033	0.024	0.011	0.100	0.017	0.010	0.011	0.003
	Ours-level-1	0.100	0.067	0.020	0	0	0.067	0.022	0.02	0.014	0	0.05	0	0.005	0.007	0
ρ	MySQL-level-2	0.400	0.517	0.630	0.714	0.778	0.500	0.578	0.667	0.748	0.804	0.600	0.675	0.745	0.807	0.850
	MySQL-level-1	0.500	0.667	0.780	0.843	0.878	0.600	0.689	0.793	0.852	0.885	0.700	0.775	0.825	0.871	0.900
	Ours-level-2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	Ours-level-1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

6. Performance Evaluation

Simulation Setting



(a) 2 MTs. (b) 3 MTs. (c) 4 MTs.

Fig. 3: MT trajectories in a plant/factory (stars for MTs, red blocks for obstacles and dots for PoIs).

TABLE II: Impact of no. of MTs on data collection ratio, energy usage ratio, and geographic fairness.

No. of MTs		2	3	4
Data collection ratio σ_D	Random	0.403	0.434	0.489
	Ours	0.836	0.895	0.844
Energy usage ratio σ_E	Random	0.163	0.171	0.186
	Ours	0.299	0.419	0.433
Geographic fairness ω_t	Random	0.287	0.305	0.344
	Ours	0.632	0.661	0.628

7. Conclusion

New framework for efficient data collection and secure data sharing based on Blockchain and DRL

- an **joint framework for both efficient data collection and secure data sharing** scheme combining **Ethereum blockchain** and **DRL** for MCS enabled IIoT scenarios.

- fully distributed DRL scheme that help each MT to sense nearby PoIs to achieve **maximum data collection** amount, **geographic fairness** and **minimum energy consumption**.

- blockchain is used to share data among MTs to pertain their **security levels**.

8. Opinion

- Good mathematical modeling and simulation of the IIoT environment.
- Critical considerations in IIoT
  - How to achieve high quality data collection with limited MT energy resource and sensing range
  - How to ensure security when sharing and exchanging data among MTs

But,

- Blockchain for availability(for DDoS tolerance)?
  - ✓ For availability, isn’t the existing IDS/IPS system better in performance?
- Can MT performs DRL operations using real-world environmental information?
- Is the cost-effective in configuring mining nodes?
- What are the countermeasures against APT attacks?

Proposal

- Consortium blockchain-based system model using cybersecurity framework could be more realistic and more scalable.

Thank you