Consortium Blockchain-based Federated Sensor-Cloud for IoT Services

Sudip Misra, *Fellow, IEEE*, Aishwariya Chakraborty, *Student Member, IEEE*, Ayan Mondal, *Member, IEEE*, and Dhanush Kamath

Abstract—This work addresses the problem of ensuring service availability, trust, and profitability in sensor-cloud architecture designed to *Sensors-as-a-Service* (Se-aaS) using IoT generated data. Due to the requirement of geographically distributed wireless sensor networks for Se-aaS, it is not always possible for a single Sensor-cloud Service Provider (SCSP) to meet the end-users requirements. To address this problem, we propose a federated sensor-cloud architecture involving multiple SCSPs for provisioning high-quality Se-aaS. Moreover, for ensuring trust in such a distributed architecture, we propose the use of *consortium blockchain* to keep track of the activities of each SCSP and to automate several functionalities through *Smart Contracts*. Additionally, to ensure profitability and end-user satisfaction, we propose a composite scheme, named BRAIN, comprising of two parts. Firstly, we define *miner's score* to select an optimal subset of SCSPs as *miners* periodically. Secondly, we propose a modified *multiple-leaders-multiple-followers Stackelberg game*-theoretic approach to decide the association of an optimal subset of SCSPs to each service. Thereafter, we evaluate the performance of BRAIN by comparing with three existing benchmark schemes through simulations. Simulation results depict that BRAIN outperforms existing schemes in terms of profits and resource consumption of SCSPs, and price charged from end-users.

Index Terms—Sensor-Cloud, Blockchain, Se-aaS, Game Theory, Miners, Federation.

1 Introduction

Sensor-cloud is an emerging architecture which aims to improve the ease of access and usability of the Internet-of-Things (IoT) technology to the common people. It provides a unified infrastructure to handle the data generated by various IoT devices [1], [2], specifically wireless sensor networks (WSNs). It envisions traditional WSNs in the form of services termed as *Sensors-as-a-Service (Se-aaS)* [3]. Similar to other cloud service-based architectures such as Platform-as-a-Service (PaaS) and Infrastructure-as-a-Service (IaaS), in sensor-cloud, a sensor-cloud service provider (SCSP) obtains the necessary hardware resources, i.e., WSNs, from their respective sensor-owners on rental basis and utilizing his/her cloud infrastructure, provisions these in the form of service units to the end-users. Thus, using sensor-cloud, an end-user with WSN-based applications is relieved from the complexities associated with purchasing, deploying, configuring, and maintaining their own sensor networks [4]. In exchange for these services, the SCSP and the sensor-owners earn revenue as service charge from the end-users.

Despite the manifold advantages of sensor-cloud architecture, provisioning Se-aaS as per the requirement of various types of end-users poses a serious practical challenge to the SCSP. In sensor-cloud, since the SCSP depends on heterogeneous types of geographically distributed WSN hardware. Hence, it may not always be possible for him/her to ensure service availability, as the necessary WSN(s) may not be registered with him/her. Moreover, the SCSP may also not possess the required cloud infrastructure, i.e., optimally placed cloud data centers, to meet the stringent Quality of Service (QoS) requirements of a particular end-user. Such a situation not only makes the sensor-cloud an unfavorable arrangement for an SCSP by decreasing the earned revenue but also for the end-users, as they have to approach multiple SCSPs to fulfill his/her Se-aaS requirement.

In the existing literature, the problem of unavailability of cloud and WSN resources with the SCSP has not been studied. However, traditional cloud service-based architectures also suffer from a similar problem and hence, several solutions are proposed which aim to ensure resource availability and high revenue simultaneously in these systems. Among those, the most widely accepted and feasible solution is the cloud federation model [5]–[7], defined as — *"Cloud federation comprises services from different providers aggregated in a single pool supporting three basic interoperability features - resource migration, resource redundancy, and combination of complementary resources resp. services"* [5]. Basically, in a cloud federation, multiple service providers cooperate among themselves to share their resources based on certain predefined terms for providing services, thereby ensuring higher service availability while maintaining high revenue. Since Sensor-cloud essentially follows a cloud-based Service-Oriented Architecture (SOA) [4], [8], we argue that adopting a similar model in case of sensor-cloud can aid in solving the problem of Se-aaS availability. However, the introduction of the involvement of multiple SCSPs in addition to the already-present multiple sensor-owners, and the associated revenue model demands the need to ensure trust and transparency in the system [9]. Otherwise, it is not possible to ensure that the SCSPs and the sensor-owners get paid their fair share of revenue while the end-users receive

• *The authors are with the Indian Institute of Technology Kharagpur, India (Email: {aishwariyach,smisra@cse,ayanmondal}.iitkgp.ac.in; dhanush2897@gmail.com).*

their desired QoS [10].

In the recent years, *blockchain* technology [11] is conceived as a highly-efficient means to ensure trust, transparency, and visibility in large-scale distributed systems and hence, can be used to solve the aforementioned problems in a federated sensor-cloud. Blockchain is based on the concept of a distributed ledger which is shared among each of the involved entities in the system. Whenever an entity performs any action which is termed as a "transaction", it gets recorded as an entry in the distributed ledger and cannot be tampered with in the future. However, in order to perform and record a transaction, firstly, it is required to be verified and validated by each (or subset) of the involved entities in order to prevent fraudulent transactions. Based on the permission of access and management, there are three types of blockchain – (1) *public* which is open and accessible to all, (2) *private* which is limited to one organization, and (3) *consortium* which is a hybrid of the two and controlled by a set of organizations. We argue that the consortium blockchain which has been widely used by the researchers in case of decentralized trading systems such as smart grid [12] is most suitable for the federated sensor-cloud architecture.

Therefore, in this work, we propose a paradigm shift from the traditional sensor-cloud by introducing consortium blockchain-based federated sensor-cloud architecture as a solution to the problem of improving Se-aaS availability. The major contributions of this work are listed as follows:

1) We propose a modified sensor-cloud architecture and workflow based on federation model and consortium blockchain technology, and establish its advantages.

2) Based on this architecture, we identify the various roles and activities of each involved entity and their interactions among themselves and with the blockchain.

3) Thereafter, we propose a secure Se-aaS provisioning scheme, BRAIN, to ensure that each involved entity benefits from the federation. The proposed scheme comprises of two parts – (a) miner selection based on score and (b) service provider selection based on Stackelberg game theory.

4) We analyze the proposed scheme, BRAIN, theoretically and evaluate its performance in comparison to existing benchmark schemes through simulations.

2 Related Works

The concept of sensor-cloud architecture, which integrates sensor networks with cloud computing to enable Sensor-as-a-Service (Se-aaS), was first introduced by Yuriama et al. [3], who outlined its fundamental functions and potential applications. This innovative framework has spurred extensive research focused on improving its performance, efficiency, and applicability within the Internet of Things (IoT). Chatterjee et al. [13] contributed by proposing a data center selection scheme aimed at delivering Se-aaS with high Quality of Service (QoS) by minimizing delays, while Misra et al. [14] developed a QoS-aware approach to optimally distribute service loads across multiple sensor nodes, balancing profitability for both SCSPs and sensor-owners. Chatterjee et al. [15] further expanded on this by addressing big-data challenges in sensor-cloud environments, proposing an architecture to handle large volumes of sensor-generated data.

In addition to performance optimization, the economic aspects of sensor-cloud systems have been explored to ensure financial sustainability. For instance, Chatterjee et al. [8] proposed a dynamic pricing model incorporating both hardware and infrastructure costs, aimed at maximizing user satisfaction. Chakraborty et al. [16] extended this concept by designing a cache-enabled pricing scheme that distributes service load across multiple caches to maximize SCSP profits. With the increased reliance on sensor-cloud systems, security has also become a priority. Sen et al. [17] developed a security risk assessment framework using attack graphs, evaluating vulnerabilities within sensor-cloud environments. Mahmoud and Shen [18] proposed a privacy protection method that camouflages sensor traffic, securing data transmission. Addressing trust, Chakraborty et al. [4] introduced a dynamic trust-enforcing pricing scheme, while Roy et al. [19] accounted for the presence of unreliable sensor nodes in their pricing model, improving service quality.

With cloud federation and blockchain gaining traction as methods for distributed, secure resource sharing, researchers adapted these technologies to enhance cloud services. Buyya et al. [5] pioneered the InterCloud framework, facilitating inter-provider resource sharing to achieve QoS goals. Celesti et al. [6] explored the operational challenges of forming federated clouds, such as interoperability and trust, while Kurze et al. [7] addressed economic challenges, including vendor lock-in and performance optimization. Blockchain has emerged as a powerful tool to further secure federated environments. For instance, Mashayekhy et al. [20] used game theory to reinforce federation stability, while Xu et al. [21] leveraged blockchain-based smart contracts to improve energy efficiency in cloud data centers. Additionally, Samaan [22] applied a repeated game model to manage cloud capacity sharing, emphasizing strategic revenue considerations. Kirkman et al. [23] demonstrated how blockchain-enabled smart contracts could automate data migration policies in cloud environments, enhancing data management.

Blockchain's potential for decentralization and security has also catalyzed research on its integration with IoT. Christidis et al. [24] were among the first to propose using smart contracts to automate IoT interactions, creating secure, decentralized processes. Reyna et al. [25] examined integration challenges, while Dorri et al. [26] suggested an optimized blockchain architecture maintained by high-resource devices to address scalability issues. Misra et al. [27] explored a secure synchronization model for IoT using Ethereum, suitable for both real-time and non-real-time devices. Blockchain has also been applied in specific IoT use cases: Leiding et al. [28] implemented a blockchain-based vehicular network for traffic and weather updates, and Aung et al. [29] developed a decentralized smart home security system. Novo [30] proposed a scalable access control system for IoT, using blockchain hubs to manage device permissions, while Zhang et al. [31] introduced a multi-contract model for access control in IoT ecosystems.

Synthesis: The aforementioned studies underscore the evolution of sensor-cloud and blockchain technologies, highlighting how blockchain enhances security, trust, and scalability across IoT applications. However, these works have the following limitations — (a) the existing works

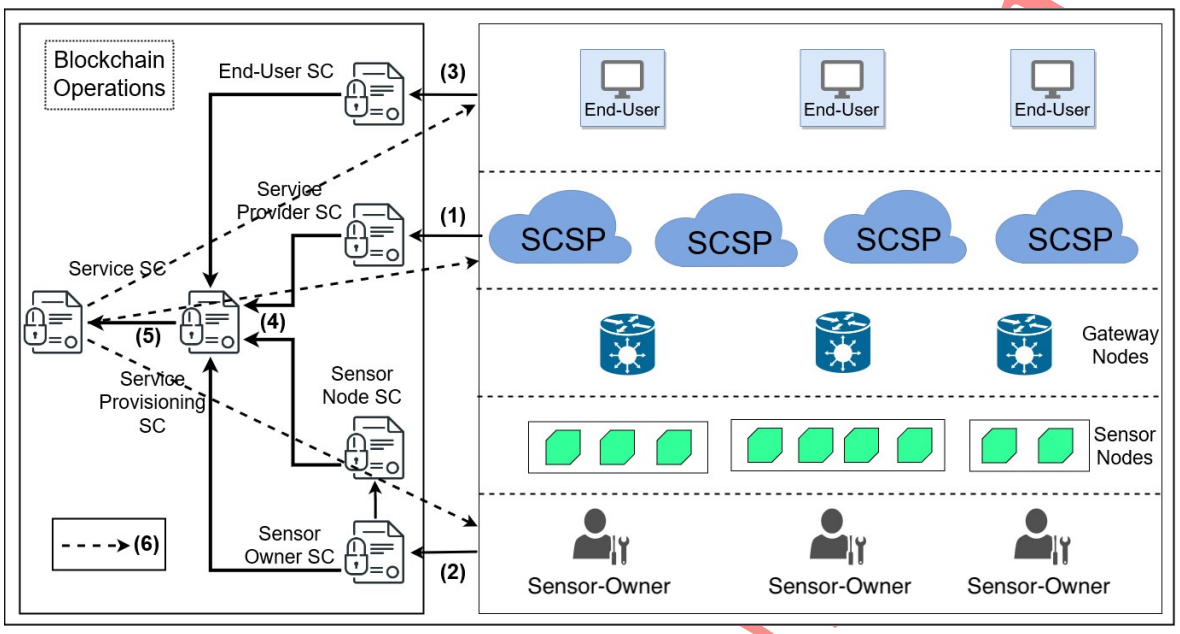

Fig. 1: Schematic Diagram of Federated Sensor-Cloud

on sensor-cloud are built on the impractical assumption that an SCSP is able to serve all types of requests with no upper-bound on the total number of services that can be handled, (b) the existing schemes on cloud services mostly consider homogeneous SOA and hence, are not suitable for sensor-cloud having heterogeneous SOA, and (c) the existing blockchain-based schemes either implement blockchain for small IoT networks or are suitable for specific use-cases. Moreover, none of the schemes consider the integration of IoT and cloud. Hence, it is necessary to design a federated architecture based on blockchain specifically for sensor-cloud in order to mitigate the problem of Se-aaS unavailability and transparency.

3 Consortium Blockchain-based Federated Sensor-Cloud Architecture

In this work, we introduce a novel architecture for sensor-cloud based on the federation model and consortium blockchain depicted in Figure 1 and explained as follows.

3.1 Involved Entities

The entities involved in the proposed architecture include multiple SCSPs each of whom have multiple registered sensor-owners. The set of sensor-owners registered with each SCSP is completely independent, partially overlapping, or completely overlapping. However, we consider that each sensor node belonging to each sensor-owner is registered with a single SCSP. Additionally, the SCSPs are heterogeneous in terms of their cloud infrastructural resources. In the proposed architecture, the same web portal is used by the SCSPs to accept the service requests.

3.2 Sensor-cloud Federation

In the proposed architecture, multiple SCSPs work together in a federation in order to provision Se-aaS. Thus, these SCSPs share their cloud infrastructure and sensor network-based resources to serve the end-users. In sensor-cloud, a single Se-aaS request may have multiple components having different requirements, each of which needs to be served using different virtual sensors. For each virtual sensor, the SCSP who is capable of serving the request with high QoS at that particular time instant is chosen to provide service. Thus, for each service, service level agreements are prepared among the SCSPs to satisfy the requirements of the end-users while maintaining QoS and cost. Here, we consider that the SCSPs are equipped with the provision to combine their provisioned service components and generate a composite service. On the other hand, the pricing policy for Se-aaS is decided based on mutual agreement of the SCSPs in the federation.

3.3 Consortium Blockchain for Sensor-cloud

In this work, we propose the use of a consortium blockchain to monitor the activities of the various involved entities without using a trusted third party. Here, we use *Smart Contracts* [23] to automate the various functionalities of the Sensor-cloud architecture. The various components [32] of the proposed consortium blockchain for sensor-cloud are:

Blockchain Network: The blockchain network for sensor-cloud comprises of SCSPs belonging to the sensor-cloud federation, sensor-owners willing to contribute their sensor nodes, and end-users requesting for Se-aaS. Initially, each of these entities needs to register to themselves in order to be a part of the blockchain network. In the proposed blockchain network, only the SCSPs have the authority to act as miners and in order to gain this authorization, each SCSP needs to make a one-time contribution of a fixed share of cryptocurrency to the sensor-cloud federation at the time of registration. The sensor-owners, on the other hand, have to provide information related to their sensor nodes while registering themselves for Se-aaS provisioning. We consider that, since the sensor nodes are highly resource-constrained

devices, the copy of the blockchain is maintained only by the gateway nodes which keep track of the usage of the connected sensor nodes. Furthermore, the end-users who require Se-aaS, register their requirements with the sensor-cloud federation and become a part of the blockchain network. However, it is ensured using smart contracts that both the end-users and the sensor-owners have limited visibility in the blockchain network.

Smart Contracts: As mentioned earlier, in this work, we propose the use of *smart contracts* to automate certain functionalities of the sensor-cloud in order to prevent the scope of malicious behavior. In the proposed architecture, six different types of smart contracts (SCs) are used:

(a) Sensor Owner SC: It is used by sensor owners to record the specifications of their deployed physical sensor nodes and their revenue transactions on the Blockchain.

(b) Sensor Node SC: It is created dynamically for each registered sensor node to store information regarding its usage statistics and physical status. It is linked with its corresponding sensor-owners and can be accessed by the gateway nodes to which the sensor node is connected.

(c) End-user SC: It is used by the end-user to request for Se-aaS by registering his/her service requirements which are received by SCSPs through the Service Provisioning SC.

(d) Service Provider SC: It is used by the SCSPs to register themselves in the sensor-cloud federation by providing information about their computational resources and contributing a fixed amount of crypto-currency which is used to make financial transactions associated with the SCSP.

(e) Service Provisioning SC: The Service Provisioning SC obtains information from all four of the aforementioned SCs, creates association among the end-user service requests, sensor-owners, and the SCSPs for provisioning Se-aaS, and generates a Service Contract SC.

(f) Service Contract SC: The Service Contract SC maintains all information regarding a particular service starting from its initiation till its termination and connects all associated end-user, sensor-owner(s), and the SCSP(s).

Transactions: Transactions, generated by altering the states of the variables in Smart Contracts, form blocks in the blockchain upon validation by miners. Each block in the sensor-cloud blockchain stores information regarding Se-aaS provisioning and the associated financial transactions. For the end-users, the specification of the desired Se-aaS along with the agreed-upon QoS parameters, the start time and stop time of service, and the amount of crypto-currency exchanged with the SCSPs are stored in the blockchain. Similarly, for the sensor-owners, the detailed specifications, physical status, and usage statistics of sensor nodes, and the payment history details are stored in the blockchain. On the other hand, for an SCSP, information about the associated services, sensor nodes, the mining count, and the amount of computational resources along with the amount of crypto-currency owned are stored in the blockchain.

Consensus Protocol: In order to give equal opportunity to each authorized SCSP to participate in the consensus and prevent bias, we introduce a hybrid PoS-PoW consensus protocol[1] for the consortium blockchain of sensor-cloud in this work. Here, for each service, a subset of the authorized SCSPs are selected through the PoS mechanism as miners who subsequently compete among themselves through the PoW mechanism to mine the transactions related to the particular service. The algorithm for the selection of the subset of SCSPs is described in details in Section 4.1.

Working Principle: Provisioning Se-aaS in sensor-cloud mainly involves the following steps – (1) Registration of SCSPs, (2) Registration of sensor-owners and their nodes, (3) Registration of service requests of end-users, (4) Association of sensor nodes and their owners, and SCSPs to a particular service, (5) Service generation and provisioning, and (6) Financial transactions. Each of these steps involves the interaction among the entities and smart contracts and generate transactions which are mined as blocks in the blockchain using consensus protocol. The workflow of the proposed architecture is depicted in Figure 1.

Therefore, the SCSPs can earn revenue from two sources – through mining and through service provisioning. Now, in order to ensure that each SCSP belonging to the federation obtains a fair opportunity to earn profit, it is necessary to select the optimal sets of SCSPs for mining and for service provisioning. Thus, in this work, we design a scheme, named BRAIN, for the optimal miner subset selection and service provider selection in consortium blockchain-based federated sensor-cloud to ensure high profits for the SCSPs, while satisfying the requirements of the end-users.

4 Proposed Scheme

In this section, we propose a dynamic scheme, named BRAIN, for balanced load distribution among the SCSPs while ensuring high profit for each SCSP and maintaining high QoS of Se-aaS in the consortium blockchain-based federated sensor-cloud architecture. Whenever an end-user places a service-request to the sensor-cloud federation, a subset of the available SCSPs are selected based on their *mining score* to act as competing miners for the service. The selected miners decide the SCSP(s) to be selected for serving the request based on a *multiple-leaders-multiple-followers Stackelberg game*. Finally, the miners use the hybrid consensus protocol to mine the service level agreement details and other related transactions, for e.g., initiation, consumption, and termination, as blocks to the blockchain. The proposed scheme is discussed in detail in the following subsections.

4.1 Selection of Miners

As discussed earlier, the proposed consensus protocol for consortium blockchain-based federated sensor-cloud is hybrid in nature, comprising of a combination of PoW and PoS mechanisms. The PoS mechanism is used to select the

1. This approach is adopted to balance security with fair participation among SCSPs, thereby ensuring that each SCSP gets a fair opportunity to contribute to consensus and preventing any single SCSP from dominating the network. Thus, although the proposed scheme introduces certain delays, this combination helps in supporting the system objectives. Since this protocol is executed only once for each service, i.e., when the service request is made by an end-user prior to service initiation, it does not have any significant impact on the throughput of the IoT services.

most efficient subset of SCSPs to perform mining for a particular service. Thereafter, using the PoW mechanism, the selected miners compete among themselves to mine each transaction related to the service and receive a fraction of the service fee in exchange as an incentive. The reason behind using this PoS-PoW hybrid consensus protocol is that using this mechanism, each SCSP in the federation obtains a fair chance to earn profit as a miner and no single SCSP gains higher control over the federation compared to others.

For employing the PoS mechanism, we first define an efficiency parameter termed as 'miner's score' which is calculated by each SCSP for every other SCSP in the federation. In order to define the *miner's score*, we consider the following characteristic parameters of each SCSP:

1) **Hash Rate:** Hash Rate H_i of a miner (here, SCSP s_i) is the number of different nonce values that it is capable of checking per second for finding the appropriate block hash according to the current difficulty set by the network. It denotes the speed at which the SCSP operates and is measured in GigaHertz. The maximum of the hash rates of the SCSPs is denoted as H_{max}.

2) **Fraction of Computation Power Allocated to Mining:** In the proposed architecture, the SCSPs simultaneously acts as miners and Se-aaS providers. Thus, each SCSP s_i in the federation commits to allocating a pre-defined amount, C_i, of its total computational power, $C_{i,tot}$, to the process of mining.

3)**Mining Count:** The number of times SCSP s_i has been selected as a miner is termed as its mining count, M_i. It signifies the revenue earned by the SCSP by mining transactions in the sensor-cloud blockchain. Introducing the negative effect of M_i in miner's score calculation ensures fairness of the miner selection process to the available SCSPs. The maximum mining count of the SCSPs in the federation is denoted by M_{max}.

4) **Service Provider Count:** The number of times SCSP s_i has been selected previously for service provisioning when SCSP s_j acted as a miner is the service provider count $S_{i,j}$ of SCSP s_i assigned by SCSP s_j. This parameter signifies the amount of revenue earned by an SCSP from Se-aaS provisioning. A higher value of this parameter indicates a higher stake of an SCSP in the sensor-cloud federation.

Using the aforementioned parameters, the miner's score of an SCSP is defined as presented in Definition 1.

Definition 1. *Miner's score, denoted as ϕ_i, is a measure of the mining capability of an SCSP and its contribution to the consensus process in the sensor-cloud blockchain. It signifies the stake of each SCSP in the federation and is calculated as follows:*

$$\phi_i = \left[\alpha \frac{H_i}{H_{max}} + \beta \frac{C_i}{C_{i,tot}} + \gamma \frac{(B - M_i)}{B} + \zeta \sum_{j \in \mathcal{N}/\{i\}} \frac{S_{i,j}}{M_{max}} \right] \quad (1)$$

where α, β, γ, and ζ are constants and B denotes the total number of blocks in the blockchain at the time of calculation.

Based on the miner's score, a preference relation among the SCSPs for being selected as miners is generated and a subset of P SCSPs having the highest preference are chosen as miners for the particular service, i.e., they compete for mining in the PoW stage. The value of P is determined by the SCSP federation based on the total number of SCSPs.

4.2 Selection of Se-aaS Provider

Next, the subset of P selected SCSPs need to decide the SCSP(s) to be selected for serving the service request. To achieve this end, we propose a *modified multiple-leaders-multiple-followers Stackelberg game*-based scheme. In this scheme, the selected SCSPs acts as the *leaders* and the set of available SCSPs capable of serving the particular request act as the *followers*. Each follower *non-cooperatively* decides the optimum price to be charged for Se-aaS while maximizing its own profit. On obtaining the pricing decision of the followers, the leaders *cooperatively* decide the most suitable SCSP(s) for serving the request. Finally, through the PoW mechanism, the service provider allocation decision corresponding to the service is mined into the blockchain.

Justification for Using Stackelberg Game:

In the proposed sensor-cloud architecture, each SCSP acts rationally and tries to maximize its profit earned by mining and by provisioning Se-aaS. At the same time, each SCSP also tries to ensure the profit of other SCSPs in order to maintain their interest in the federation. In order to capture this nature of interaction among multiple SCSPs, we use a multiple-leaders-multiple-followers Stackelberg game. In this game, each follower, or SCSP capable of serving a particular request, decides its optimal pricing strategy by maximizing its profit. On the other hand, the leaders, or SCSPs selected as miners, select the most optimal SCSP for service provisioning while considering the pricing strategies of the followers as well as the overall benefit of the federation. The detailed discussion and analysis of the proposed game-theoretic scheme are presented as follows.

4.2.1 Game Formulation

We consider that each service-request q_j made by an end-user is specified in terms of three parameters — geographical region of interest G_j, type of service T_j, and data-rate R_j. Here, T_j and R_j represent the sets of one or more types of sensor data requested by the service and their corresponding data-rates, respectively. Hence, the followers in this game comprise of only those SCSPs which have the required type T_j of sensor nodes, which the available required bandwidth, deployed in region G_j.

Utility Function for Followers: The utility function $\mathcal{U}_{i,j,k}$ of SCSP s_i for serving request q_j of type k, where $k \in T_j$, quantifies the benefit earned by the SCSP by serving the request. Thus, it mainly depends on the profit earned and the resources spent by the SCSP for serving the request. The various factors influencing the utility of a follower SCSP are as follows:

1) The utility function of the SCSP varies directly with the price $P_{i,j,k}$ charged per unit service by the SCSP and inversely with the cost incurred for service provisioning. We consider that a fixed cost $C_{j,k}$ is incurred by each SCSP for provisioning each unit of service q_j of type k. However, with the increase in the price, there is a simultaneous decrease in the preference of the SCSP for being selected for Se-aaS provisioning. Hence, we argue that the utility of the SCSP also varies negatively with the profit earned.

2) The utility function of the SCSP also varies proportionally with the requested data-rate $r_{j,k}$ of the service q_j. This is because of the fact that, for a given service duration, an SCSP earns higher profit by serving a request with a higher data-rate requirement.

3) The utility function of the SCSP varies negatively with the fraction of services that are being served by the SCSP s_i. This is because of two reasons: firstly, with the increase in the number of services n_i being served, the resource consumption of the SCSP increases, secondly, the increase in the total number of services allocated to s_i decreases his/her chance for being selected for further services.

Hence, we define the utility function of the SCSP as follows:

$$\mathcal{U}_{i,j,k} = \frac{r_{j,k}}{R_{max}} \left(\frac{P_{i,j,k} - C_{j,k}}{P_{max}} \right) - \frac{n_i}{b_{curr}} \left(\frac{P_{i,j,k} - C_{j,k}}{P_{max}} \right)^2 \tag{2}$$

where, R_{max}, P_{max}, and b_{curr} denote the maximum possible data-rate of Se-aaS, the maximum price that the end-user is willing to pay, and the total number of services being served at the current time, respectively. Note that, here, the first term represents the direct profit per unit service, adjusted by the data-rate. It increases as the data-rate or service price increases, which naturally aligns with the SCSP's goal of maximizing profit. On the other hand, the second term depicts a quadratic *penalty*. This squared term reflects the compounding negative effect of price increases on the SCSP's selection probability. Higher prices reduce the likelihood of the SCSP's services being selected, as the system favors cost-effective SCSPs. Thus, the squared term used here captures this effect by penalizing excessive pricing more severely as demand grows, ensuring the utility function aligns with profit while curbing excessive prices. Hence, the objective of each follower SCSP is defined as follows:

$$\arg\max_{P_{i,j,k}} \mathcal{U}_{i,j,k} \tag{3}$$

subject to the constraints – $C_{j,k} < P_{i,j,k} \leq P_{max}$ and $r_{j,k} \leq R_{max}$.

Utility Function for Leaders: The utility function $B_{l,j}$ of the SCSP s_l, acting as the leader, signifies the overall benefit to the sensor-cloud federation obtained by assigning service q_j to a set of follower SCSPs $\mathcal{S}_f$. The following factors are considered while designing the utility function for the leaders:

1) As mentioned earlier, after paying the price demanded by the SCSPs for service provisioning, the remaining part of the revenue is distributed equally among the selected set of miners. Hence, for a fixed price paid by the end-user, lower the price charged for service provisioning, higher is the revenue for the miners.

2) Considering that the cost to mine a single transaction C_m is fixed, higher data-rate demanded by a service necessitates the mining of a higher number of blocks which incurs higher cost to the miners[2].

Therefore, we define the utility function as follows:

$$B_{l,j} = P_{max} - \sum_{s_i \in \mathcal{S}_f} \sum_{k \in T_j} P_{i,j,k} x_{i,j,k} - P \sum_{k \in T_j} C_m r_{j,k} \tag{4}$$

where $\mathcal{S}_f$ denotes the set of follower SCSPs and $x_{i,j,k}$ is an association vector which denotes whether follower SCSP s_i provides a service of type k for service-request q_j. We consider that —

$$x_{i,j,k} = \begin{cases} 1, & \text{if type } k \text{ of service } q_j \text{ is assigned to } s_i \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

Hence, the objective of the leader s_l is defined as follows:

$$\arg\max_{\boldsymbol{x}_j} \mathcal{B}_{l,j} \tag{6}$$

where $\boldsymbol{x}_j$ defines the selection vector for service q_j and $\boldsymbol{x}_j = \{\cdots, x_{i,j,k}, \cdots\}$, subject to the constraint – $\sum_{k \in T_j} P_{i,j,k} r_{j,k} x_{i,j,k} < P_{max}$. The leader SCSP has to decide the vector $\boldsymbol{x}$ which maximizes his/her utility function.

4.2.2 Existence of Stackelberg Equilibrium

In this section, we determine the existence of Stackelberg equilibrium, as defined in Definition 2, in the proposed scheme, BRAIN, for selection of Se-aaS provider. Here, we consider that the SCSPs, who act as the followers, decide their strategies, i.e., the price to be charged, based on the service requirements of the end-users. Thereafter, each miner selects the SCSP to whom the service is to be allocated while ensuring the overall benefit of the sensor-cloud federation. Thereby, we argue that with the existence of equilibrium among the followers, we can ensure the existence of Stackelberg equilibrium in BRAIN. Hence, we evaluate the Karush-Kuhn-Tucker (KKT) conditions with KKT multiplier, as mentioned in Theorem 1.

Definition 2. *In BRAIN, we define the Stackelberg Equilibrium among the SCPSs for the selection of Se-aaS provider as an optimal point as the tuple* $< r^*_{j,k}, P^*_{i,j,k}, \boldsymbol{x}^*_j >$, *where* $r^*_{j,k}$ *denotes the optimal datarate requirement of service* q_j *having type* k; $P^*_{i,j,k}$ *is the optimal price charged by SCSP* s_i *for provisioning Se-aaS to service* q_j *having type* k; *and* $\boldsymbol{x}^*_j$ *denotes the optimal decision vector, i.e., the choice of the SCSP, i.e., the miner. We argue that at Stackelberg equilibrium, the following conditions are satisfied:*

$$\mathcal{U}_{i,j,k}(r^*_{j,k}, P^*_{i,j,k}) \geq \mathcal{U}_{i,j,k}(r^*_{j,k}, P_{i,j,k}) \tag{7}$$

$$B_{l,j}(r^*_{j,k}, P^*_{i,j,k}, \boldsymbol{x}^*_j) \geq B_{l,j}(r^*_{j,k}, P^*_{i,j,k}, \boldsymbol{x}_j) \tag{8}$$

Here, we observe that the decision of each follower is not affected by the decision of the leader in one stage. However, we argue that as BRAIN follows a multi-stage game, the choice of

2. It is noteworthy that, in the proposed scheme, while not every data packet is recorded, each service contract tracks key service details from initiation to termination, with updates logged periodically on the blockchain by gateway nodes. Higher data-rates increase the frequency of these updates, indirectly raising the number of transactions and blocks mined over time, which in turn impacts the overall mining workload.

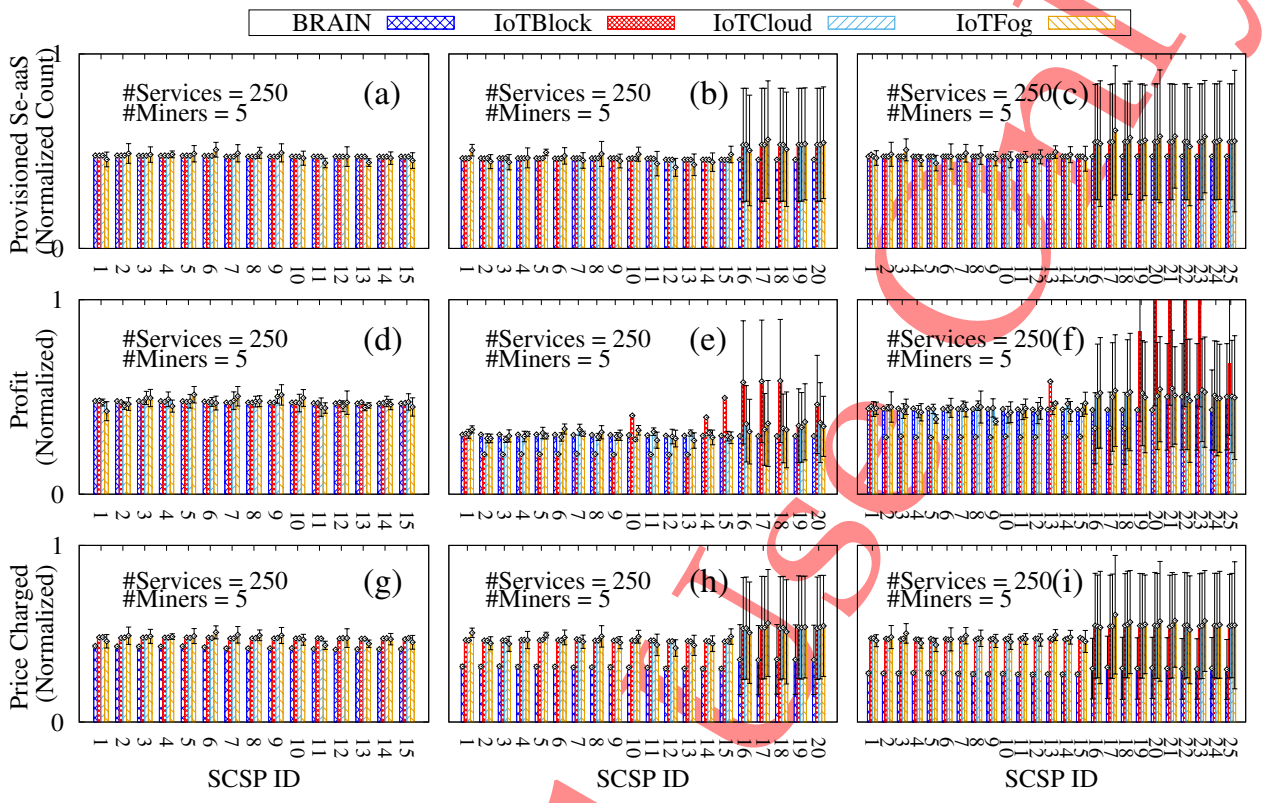

Fig. 2: Variation of the number of times each SCSP is selected for Se-aaS provisioning, his/her profit, and the corresponding price charged from the end-users with increasing number of SCSPs in the federation

the leader always has an impact on the follower in the next stage. This eventually strengthens the justification for using Stackelberg game in BRAIN, as mention earlier in Section 4.2.

Theorem 1. *In BRAIN, given the datarate requirement $r_{j,k}$ of service q_j of type k, there exists at least one Stackelberg equilibrium point which satisfies the constraints mentioned in Equations (7) and (8).*

Proof. In to evaluate the existence of Stackelberg equilibrium in BRAIN, we take into consideration the overall utility function of the system $\mathcal{F}_i$ for service q_j of type k. Thereby, we consider that — $\mathcal{U}_j = \sum_i \mathcal{U}_{i,j,k}$. Hence, by applying the KKT conditions with KKT multiplier [4], we obtain the following *regularity* conditions:

Stationary condition:

$$\nabla \mathcal{F}_j = \nabla \sum \lambda_{i,1} \sum_i \mathcal{U}_{i,j,k} - \lambda_2 \nabla (P_{max} - P_{i,j,k}) - \lambda_3 \nabla (R_{max} - r_{j,k}) + \lambda_4 \nabla (P_{i,j,k} - C_{j,k}) \quad (9)$$

where $\lambda_{n,1}$, λ_2, λ_3, and λ_4 are KKT multipliers.

Primal feasibility condition:

$$\left.\begin{array}{r} (P_{max} - P_{i,j,k}) \geq 0 \\ (R_{max} - r_{j,k}) \geq 0 \\ (P_{i,j,k} - C_{j,k}) > 0 \end{array}\right\}, \quad \forall i \quad (10)$$

Dual feasibility condition:

$$\lambda_{i,1}, \lambda_2, \lambda_3, \text{ and } \lambda_4 \geq 0 \quad (11)$$

Complementary slackness condition:

$$\left.\begin{array}{r} \lambda_{i,1} \sum_i \mathcal{U}_{i,j,k} = 0 \\ \lambda_2 \nabla (P_{max} - P_{i,j,k}) = 0 \\ \lambda_3 \nabla (R_{max} - r_{j,k}) = 0 \\ \lambda_4 \nabla (P_{i,j,k} - C_{j,k}) \end{array}\right\} \quad (12)$$

We evaluate the Jacobian matrix $\nabla \mathcal{F}_j$, i.e., the Hessian matrix of $\mathcal{F}_j$, which is represented as follows:

$$\nabla^2 \mathcal{F}_j = \begin{bmatrix} -\frac{2\lambda_{1,1} n_1}{b_{curr} P_{max}} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & -\frac{2\lambda_{|\mathcal{S}|,1} n_{|\mathcal{S}|}}{b_{curr} P_{max}} \end{bmatrix} \quad (13)$$

Here, we observe that the Hessian matrix is a diagonal matrix having negative elements, as $n_i, b_{curr}, P_{max} \geq 0$. Therefore, we argue that in the proposed scheme BRAIN, the Stackelberg equilibrium point always exists. □

5 Performance Evaluation

To evaluate the performance of the proposed scheme, BRAIN, we conducted simulations in a Python-based simulation platform and compared the results with three existing benchmarks schemes. The details of the simulations and the results are explained in the subsequent sections.

5.1 Simulation Parameters

We simulated a blockchain-based sensor-cloud comprising of multiple SCSPs in a federation, multiple registered

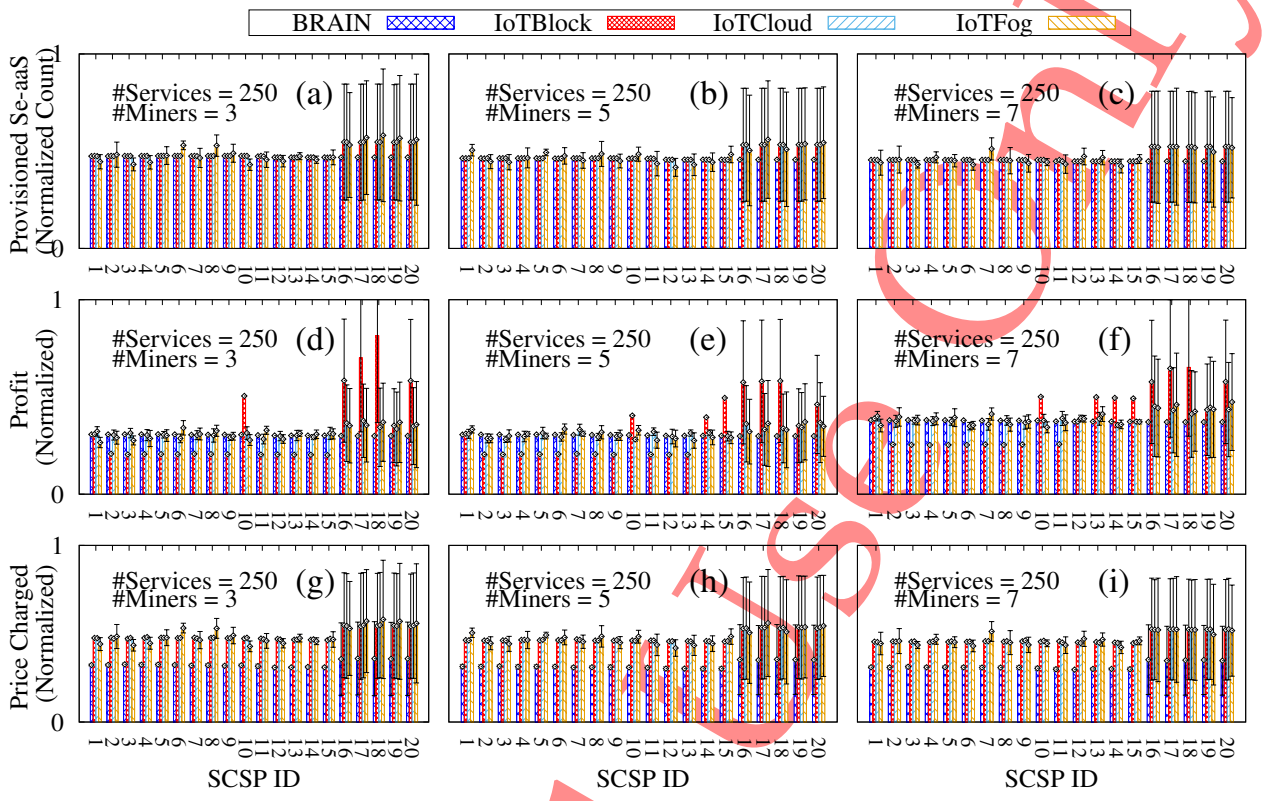

Fig. 3: Variation of the number of times each SCSP is selected for Se-aaS provisioning, their profits, and the corresponding price charged from the end-users with increasing number of miners for each round

sensor-owners, and several end-users. We considered that the service requests of the end-users, having randomly generated data-rate requirements, arrive at the SCSP federation sequentially and that each SCSP is capable of serving each of the requests. The hash rate and the computational capacity of each SCSP are decided randomly, as mentioned in Table 1. We conducted these experiments in two parts. In the first part, we varied the number SCSPs to be selected as miners in each round while keeping the total number of SCSPs constant at 20. In the second part, we varied the total number of SCSPs in the federation while keeping the number of miners fixed at 5.

TABLE 1: Simulation Parameters

Parameter	Value
Number of SCSPs	15, 20, 25
Number of miners per round	3,5,7
Hash Rate of miners	260000-280000
Computational Power of miners	30%-70%
Communication protocol	IEEE 802.15.4
Number of service requests	100, 250, 500
Maximum data-rate	250 $kbps$/service
Price paid by end-users	100 units/service

5.2 Benchmarks

In the existing literature, none of the works on sensor-cloud considered the application of federation or blockchain in its architecture. Hence, for comparative analysis, we chose existing resource management schemes for IoT networks, Fog networks, and cloud. We compared the performance of BRAIN with three existing benchmark schemes – Computing Resource Allocation in Three-Tier IoT Fog Networks (IoTFog) [33], Blockchain Meets IoT (IoTBlock) [30], and Intelligent Resource Management in Blockchain-Based Cloud Datacenters (DC) (IoTCloud) [34] – discussed as follows.

In IoTFog, Zhang *et al.* [33] proposed a joint optimization-based resource allocation scheme for service-based IoT-Fog networks in which Stackelberg game is used to decide the optimal pricing and resource allocation between the data service subscribers (DSSs) and data service operators (DSOs), and a matching game is used to obtain an optimal assignment of DSOs to fog nodes. The architecture is similar to the sensor-cloud architecture except for the presence of sensor nodes and the corresponding owners. Additionally, in this work, the authors do not consider the use of blockchain. Hence, for the sake of uniformity, we consider a blockchain-based architecture with random selection of miners and random allocation of services.

In IoTBlock, Novo [30] proposed a blockchain-based model for energy-efficient resource management for request scheduling in case of cloud DCs, while considering the possibility of malfunctioning of one or more DCs. Here, a reinforcement learning-based algorithm is used within smart contracts to solve the energy cost optimization problem. In IoTCloud, Xu *et al.* [34] proposed a decentralized architecture for access control within IoT networks in which the blockchain is not integrated within the IoT devices. Here, access control is provided using a single smart contract on

real-time to the IoT devices thereby, improving scalability and reducing communication overheads among the nodes.

5.3 Performance Metrics

We evaluated the performance of the proposed scheme, BRAIN, based on the following performance metrics.

Number of times each SCSP is selected for provisioning Se-aaS: It decides the revenue earned by each SCSP by provisioning Se-aaS and is dependent on the chosen set of miners and their decisions. A higher value of this parameter implies a higher chance to earn profit and hence, is favorable to the SCSPs.

Profit of each SCSP: As sensor-cloud has an integrated business model, it is essential to ensure that the involved SCSPs earn high profits in order to maintain their participation in the federation.

Number of times each SCSP is selected as miner: The SCSPs selected as miners choose the subset of SCSPs to be allocated for serving a request. Thus, the number of times an SCSP is selected as a miner signifies its control over the functioning of the federation.

Price charged from end-user: To maintain the satisfaction of the end-users, it is essential to ensure that the price charged by each SCSP for Se-aaS from an end-user is reasonable.

Resource consumption for each SCSP: This parameter deals with the amount of resources consumed for mining the blockchain-based transactions.

5.4 Results and Discussions

From Figures 2(a)-(c) and 3(a)-(c), we observe that, the number of times each SCSP is selected for provisioning Se-aaS is equally distributed among the SCSPs using BRAIN, as compared to IoTBlock, IoTCloud, and IoTFog. This is because, in BRAIN, unlike the other schemes, the utility function of each follower decreases with the increase in the number of services that are already served by him/her. In IoTFog, service delay and computational resources possessed by each service provider are considered for service allocation. In IoTBlock, only the cost of energy consumption is considered. In IoTCloud, the authors did not consider the presence of multiple service providers. Thus, in these three cases, we observe randomness in the distribution of the service requests. This randomness becomes more evident with the increase in the number of miners in each round and the total number of SCSPs in the federation. However, these variations do not have any effect on nearly equal distribution obtained using BRAIN.

We observe the variation of the profit earned by each SCSP using the four schemes from Figures 2(d)-(f) and 3(d)-(f). We yield that this parameter also follows a similar pattern to that depicted in Figures 2(a)-(c) and 3(a)-(c). This is due to the fact that the profit of the SCSP is highly influenced by the revenue earned from provisioning Se-aaS to the end-users. Hence, we observe that using BRAIN, the profit is fairly distributed among the SCSPs in the federation unlike the other three schemes, for which the distribution is random. Moreover, we observe a slight variation in the distribution of profit in case of BRAIN as compared to the distribution of the service requests. This is because the SCSPs also earn revenue by mining transactions to the sensor-cloud blockchain and incur a cost of provisioning Se-aaS. However, both of these factors have limited effect on the profit earned by each SCSP as compared to the revenue earned from Se-aaS provisioning.

Figures 2(g)-(i) and 3(g)-(i) depict the variation in the price charged by the SCSPs from the end-users for provisioning Se-aaS using the four schemes. We observe that the price charged by the SCSPs from the end-users is lower using BRAIN, as compared to the other three schemes and this decrease becomes more prominent with the increase in the number of SCSPs in the federation. This is because, in BRAIN, the selection of SCSPs for service provisioning is done based on their utility functions, which, in turn, depend on the price being charged by them. Due to the presence of the negative quadratic term in the utility of the followers and the summation terms in the utility of the leaders, the SCSPs charging lower price are preferred over others. However, in the case of the three existing schemes, the price charged from the end-users has not been considered for the resource allocation process. Moreover, in case of BRAIN, with the increase in the number of SCSPs, the competition for being chosen for Se-aaS provisioning increases, thereby reducing the prices even further.

From Figures 4(a)-(c) and 5(a)-(c), we observe the variation in the number of times each SCSP is selected as miner for each of the four schemes. We yield that the distribution of this parameter obtained in case of BRAIN is nearly uniform, while that in case of the other three schemes is random. The slight randomness observed in case of BRAIN is due to the differences in the computational capacity of each SCSP, whose values, as mentioned earlier, have been generated randomly during simulations. In BRAIN, the computational capacity of each SCSP is taken into consideration during the miner selection process, unlike the other three schemes. Thus, using IoTCloud, IoTFog, and IoTBlock, some of the SCSPs get repeatedly chosen as miners while some others are deprived. Additionally, we observe that, with the increase in the number of SCSPs, the randomness in the distribution of the aforementioned parameter increases for each of the schemes. However, the increase is less significant in BRAIN, implying that BRAIN outperforms the existing schemes.

Figures 4(d)-(f) and 5(d)-(f) depict the variation in the resource consumption of each SCSP for mining using the four schemes. We observe that the amount of resources consumed for mining using BRAIN is significantly lower compared to IoTBlock, and higher compared to IoTFog and IoTCloud. This is because – In BRAIN, mining is carried out by a set of highly powerful SCSPs, unlike in IoTBlock, in which the computational capacity of the SCSPs is not considered for miner selection. On the other hand, in IoTCloud and IoTFog, miner selection is random and hence, the resource consumption of the SCSPs is not uniformly distributed, unlike BRAIN, in which the distribution is nearly uniform and no SCSP is depleted of resources faster than the others. Moreover, these two schemes consider that the resource allocation decisions are the only transactions mined into the blockchain, which is unlike the proposed blockchain-based sensor-cloud architecture, in which miners are required to mine all transactions related to a particular service.

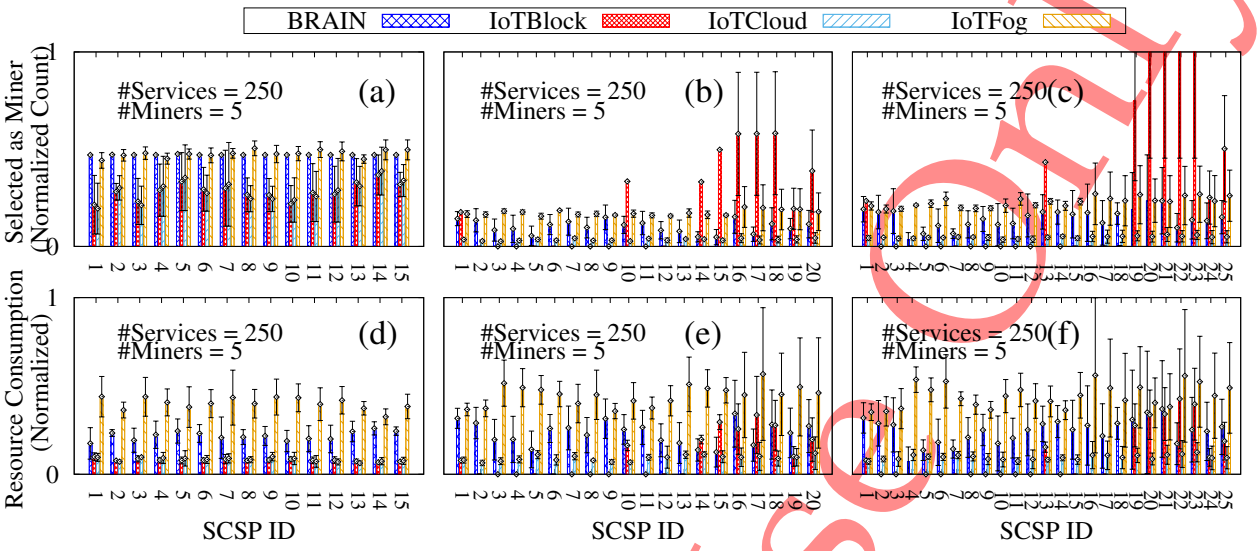

Fig. 4: Variation of the number of times each SCSP is selected as miner and the resource consumption of each SCSP with increasing number of SCSPs in the federation

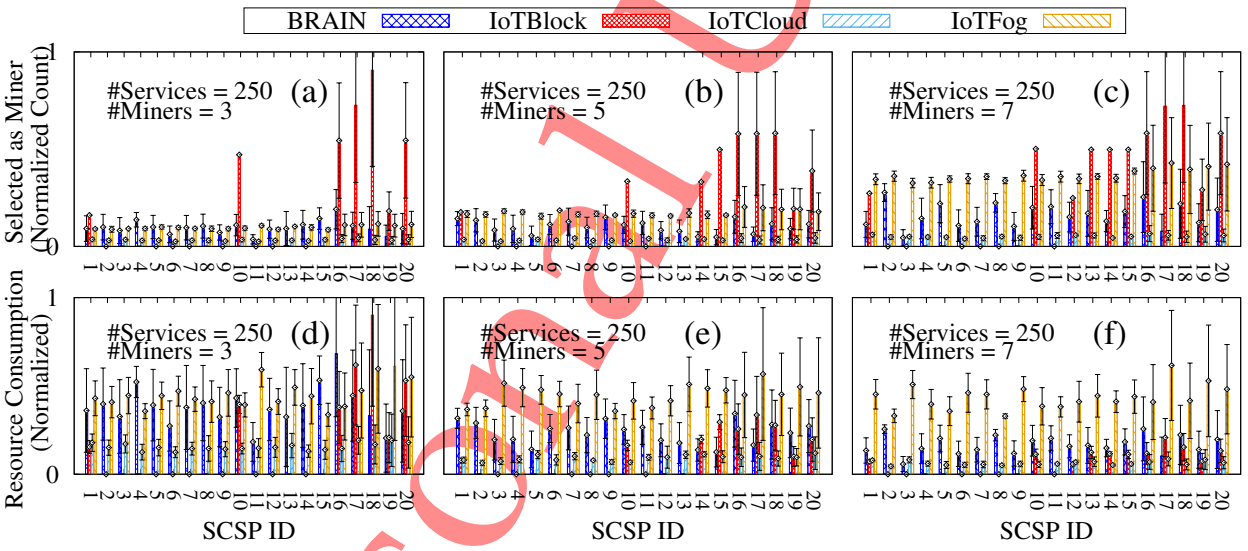

Fig. 5: Variation of the number of times each SCSP is selected as miner and the resource consumption of each SCSP with increasing number of miners for each round

6 Limitations and Discussions

In this section, we discuss how the proposed scheme ensures scalability, security and privacy, while addressing various practical challenges in deployment, geographical impact, and economic behavior. We also outline the limitations of the scheme and suggest possible remedies.

Scalability: The proposed architecture ensures scalability by leveraging a consortium blockchain with the hybrid PoS-PoW consensus mechanism, which is well-suited for handling high transaction volumes within sensor-cloud networks. This permissioned blockchain design reduces the computational load compared to public blockchains, making it adaptable for larger deployments with increasing SCSPs and IoT devices. To improce scalability further, additional measures, such as sharding and off-chain storage, could be integrated to manage data and transaction demands as the system scales in real-world scenarios.

Security and Privacy: In our system, security and privacy are core advantages of using blockchain, which offers immutability and transparency. The consortium model limits its access to trusted SCSPs, while smart contracts automate and secure data interactions, protecting against unauthorized access and tampering. Although this work does not address specific security threats, potential issues like cyber-attacks and data breaches can be mitigated by implementing additional privacy-preserving measures (e.g., data encryption and zero-knowledge proofs) and enhancing network resilience. These features, combined with regular security assessments, can help to safeguard against malicious attacks.

Practical Deployment Challenges: Despite the system's potential, there are practical deployment challenges asso-

ciated with its implementation. Establishing the necessary infrastructure for consortium blockchain and integrating it with existing cloud and IoT systems require both significant resources and customization. These challenges could be addressed through a phased deployment strategy, allowing incremental testing and scaling, as well as partnerships with cloud providers to ease integration. This approach supports a smoother, more manageable rollout in real-world environments.

Impact of Geographical Factors: Geographical factors can significantly impact service quality, as physical distance between SCSPs, sensor nodes, and end-users can affect latency and data transfer speeds. To address this, the architecture may be programmed to prioritize SCSPs that are closer to end-users for latency-sensitive applications or deploy edge nodes for localized processing. Incorporating geographical proximity into SCSP selection criteria would improve QoS and responsiveness, especially for applications that require real-time data processing.

Presence of Irrational Entities: Finally, while this work assumes rational behavior and stable economic incentives among SCSPs and end-users, real-world applications may introduce challenges from irrational behaviors or misaligned incentives. Economic fluctuations or irrational pricing strategies could destabilize resource distribution. Hence, to encourage SCSPs to adopt the proposed optimal pricing and resource allocation strategies, incentive structures can be incorporated within the smart contracts and penalty mechanisms for irrational behaviors, such as overpricing or service rejection, can be devised.

7 Conclusion

In this work, we proposed a consortium blockchain-based federated sensor-cloud architecture comprising of multiple SCSPs working together to provide higher quality Se-aaS at a reasonable price. In the proposed architecture, consortium blockchain is used to maintain trust among the SCSPs in the federation without a third party. Additionally, the functionalities of the sensor-cloud are automated using Smart Contracts. In order to ensure profits and efficient resource usage for the SCSPs in the proposed architecture, we also proposed a dynamic scheme, named BRAIN, comprising of two parts – optimal selection of miner subset for each service using *miner's score* and optimal mapping of the service provider to the services using a modified form of multiple-leaders-multiple-followers Stackelberg game. Through simulations, we observed that BRAIN outperforms the existing benchmark schemes in terms of profit and resource consumption of the SCSPs, and price charged from end-users.

In future, this work can be extended to consider the actual geographical location of the servers of the SCSPs as well as the sensor nodes while deciding the optimal service provider mapping to ensure better QoS. It can also be extended to consider the possibility of service migration among different SCSPs in case of unwanted failures. For specific scenarios, the selection probability of SCSPs can be modeled quantitatively instead of qualitatively, and its impact on the proposed scheme can also be explored.

References

[1] A. Soltani Panah, A. Yavari, R. van Schyndel, D. Georgakopoulos, and X. Yi, "Context-driven granular disclosure control for internet of things applications," *IEEE Transactions on Big Data*, vol. 5, no. 3, pp. 408–422, 2019.

[2] M.-P. Hosseini, D. Pompili, K. Elisevich, and H. Soltanian-Zadeh, "Optimized deep learning for eeg big data and seizure prediction bci via internet of things," *IEEE Transactions on Big Data*, vol. 3, no. 4, pp. 392–404, 2017.

[3] M. Yuriyama, T. Kushida, and M. Itakura, "A New Model of Accelerating Service Innovation with Sensor-Cloud Infrastructure," in *Ann. SRII Glob. Conf.*, Mar 2011, pp. 308–314.

[4] A. Chakraborty, A. Mondal, A. Roy, and S. Misra, "Dynamic Trust Enforcing Pricing Scheme for Sensors-as-a-Service in Sensor-Cloud Infrastructure," *IEEE Trans. on Serv. Comp.*, pp. 1–12, 2018, DOI: 10.1109/TSC.2018.2873763.

[5] R. Buyya, R. Ranjan, and R. N. Calheiros, "InterCloud: Utility-Oriented Federation of Cloud Computing Environments for Scaling of Application Services," in *Algo. and Arch. for Par. Proc.*, Berlin, Heidelberg, 2010, pp. 13–31.

[6] A. Celesti, F. Tusa, M. Villari, and A. Puliafito, "How to Enhance Cloud Architectures to Enable Cross-Federation," in *Proc. of IEEE 3rd Int. Conf. on Cloud Comp.*, Jul 2010, pp. 337–345.

[7] T. Kurze, M. Klems, D. Bermbach, A. Lenk, S. Tai, and M. Kunze, "Cloud federation," pp. 1–7, 2011.

[8] S. Chatterjee, R. Ladia, and S. Misra, "Dynamic Optimal Pricing for Heterogeneous Service-Oriented Architecture of Sensor-cloud Infrastructure," *IEEE Trans. on Serv. Comp.*, 2015.

[9] Z. Su, F. Biennier, Z. Lv, Y. Peng, H. Song, and J. Miao, "Toward architectural and protocol-level foundation for end-to-end trustworthiness in cloud/fog computing," *IEEE Transactions on Big Data*, vol. 8, no. 1, pp. 35–47, 2022.

[10] M. Zhang, F. Beltrán, and J. Liu, "A survey of data pricing for data marketplaces," *IEEE Transactions on Big Data*, vol. 9, no. 4, pp. 1038–1056, 2023.

[11] M. Swan, *Blockchain: Blueprint for a New Economy*. O'Reilly Media, Inc., 2015.

[12] M. L. Di Silvestre, P. Gallo, M. G. Ippolito, E. R. Sanseverino, and G. Zizzo, "A Technical Approach to the Energy Blockchain in Microgrids," *IEEE Trans. on Ind. Inf.*, vol. 14, no. 11, pp. 4792–4803, 2018.

[13] S. Chatterjee, S. Misra, and S. Khan, "Optimal Data Center Scheduling for Quality of Service Management in Sensor-cloud," *IEEE Trans. on Cloud Comp.*, 2015, DOI: 10.1109/TCC.2015.2487973.

[14] S. Misra and A. Chakraborty, "QoS-Aware Dispersed Dynamic Mapping of Virtual Sensors in Sensor-Cloud," *IEEE Trans. on Serv. Comp.*, pp. 1–12, 2019, DOI: 10.1109/TSC.2019.2917447.

[15] S. Chatterjee, A. Roy, S. K. Roy, S. Misra, M. Bhogal, and R. Daga, "Big-sensor-cloud infrastructure: A holistic prototype for provisioning sensors-as-a-service," *IEEE Transactions on Cloud Computing*, pp. 1–1, 2019.

[16] A. Chakraborty, A. Mondal, and S. Misra, "Cache-Enabled Sensor-Cloud: The Economic Facet," in *Proc. of IEEE WCNC*, Apr 2018, pp. 1–6.

[17] A. Sen and S. Madria, "Risk Assessment in a Sensor Cloud Framework Using Attack Graphs," *IEEE Trans. on Serv. Comp.*, 2016.

[18] M. M. E. A. Mahmoud and X. Shen, "A Cloud-Based Scheme for Protecting Source-Location Privacy against Hotspot-Locating Attack in Wireless Sensor Networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 10, pp. 1805–1818, 2012.

[19] A. Roy, S. Misra, and P. Dutta, "Dynamic pricing for sensor-cloud platform in the presence of dumb nodes," *IEEE Transactions on Cloud Computing*, pp. 1–1, 2019.

[20] L. Mashayekhy, M. M. Nejad, and D. Grosu, "Cloud federations in the sky: Formation game and mechanism," *IEEE Transactions on Cloud Computing*, vol. 3, no. 1, pp. 14–27, 2015.

[21] C. Xu, K. Wang, and M. Guo, "Intelligent Resource Management in Blockchain-Based Cloud Datacenters," *IEEE Cloud Comp.*, vol. 4, no. 6, pp. 50–59, Nov 2017.

[22] N. Samaan, "A Novel Economic Sharing Model in a Federation of Selfish Cloud Providers," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 1, pp. 12–21, 2014.

[23] S. Kirkman and R. Newman, "A Cloud Data Movement Policy Architecture Based on Smart Contracts and the Ethereum Blockchain," in *Proc. of IEEE IC2E*, Apr 2018, pp. 371–377.

[24] K. Christidis and M. Devetsikiotis, "Blockchains and Smart Contracts for the Internet of Things," *IEEE Acc.*, vol. 4, pp. 2292–2303, 2016.
[25] A. Reyna, C. Martín, J. Chen, E. Soler, and M. Díaz, "On Blockchain and Its Integration with IoT: Challenges and Opportunities," *Fut. Gen. Comp. Sys.*, vol. 88, pp. 173 – 190, 2018.
[26] A. Dorri, S. S. Kanhere, and R. Jurdak, "Towards an Optimized BlockChain for IoT," in *Proc. of the* 2^{nd} *Int. Conf. on IoT Des. and Imp.*, 2017, pp. 173–178.
[27] S. Misra, A. Mukherjee, A. Roy, N. Saurabh, Y. Rahulamathavan, and M. Rajarajan, "Blockchain at the Edge: Performance of Resource-Constrained IoT Networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 1, pp. 174–183, 2021.
[28] B. Leiding, P. Memarmoshrefi, and D. Hogrefe, "Self-managed and Blockchain-based Vehicular Ad-hoc Networks," in *Proc. of the ACM Int. Joint Conf. on Perv. and Ubi. Comp.: Adj.*, 2016, pp. 137–140.
[29] Y. N. Aung and T. Tantidham, "Review of Ethereum: Smart Home Case Study," in *Proc. of the* 2^{nd} *INCIT*, Nov 2017, pp. 1–4.
[30] O. Novo, "Blockchain Meets IoT: An Architecture for Scalable Access Management in IoT," *IEEE IoT J.*, vol. 5, no. 2, pp. 1184–1195, Apr 2018.
[31] Y. Zhang, S. Kasahara, Y. Shen, X. Jiang, and J. Wan, "Smart Contract-Based Access Control for the Internet of Things," *IEEE IoT J.*, vol. 6, no. 2, pp. 1594–1605, Apr. 2019.
[32] A. Yazdinejad, R. M. Parizi, A. Dehghantanha, and K. R. Choo, "Blockchain-enabled Authentication Handover with Efficient Privacy Protection in SDN-based 5G Networks," *IEEE Trans. on Net. Sc. and Engg.*, pp. 1–1, 2019, dOI: 10.1109/TNSE.2019.2937481.
[33] H. Zhang, Y. Xiao, S. Bu, D. Niyato, F. R. Yu, and Z. Han, "Computing Resource Allocation in Three-Tier IoT Fog Networks: A Joint Optimization Approach Combining Stackelberg Game and Matching," *IEEE IoT J.*, vol. 4, no. 5, pp. 1204–1215, Oct 2017.
[34] C. Xu, K. Wang, and M. Guo, "Intelligent Resource Management in Blockchain-Based Cloud Datacenters," *IEEE Cloud Comp.*, vol. 4, no. 6, pp. 50–59, Nov 2017.

Sudip Misra (SM'11) is a Professor at IIT Kharagpur. He received his Ph.D. degree from Carleton University, Ottawa, Canada. Prof. Misra is the author of over 350 scholarly research papers. He has won several national and international awards including the IEEE ComSoc Asia Pacific Young Researcher Award during IEEE GLOBECOM 2012. He was also the recipient of several academic awards and fellowships such as the INSA NASI Fellow Award (National Academy of Sciences, India), the Young Scientist Award (National Academy of Sciences, India), Young Systems Scientist Award (Systems Society of India), and Young Engineers Award (Institution of Engineers, India). He has also been serving as the Associate Editor of the IEEE TRANSACTIONS ON MOBILE COMPUTING, the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, the IEEE TRANSACTIONS ON SUSTAINABLE COMPUTING, the IEEE SYSTEMS JOURNAL, and the INTERNATIONAL JOURNAL OF COMMUNICATION SYSTEMS (Wiley). He is a Guest Editor of the IEEE NETWORK Magazine. He is also an Editor/Editorial Board Member/Editorial Review Board Member of the IET NETWORKS and the IET WIRELESS SENSOR SYSTEMS. Dr. Misra has 11 books published by Springer, Wiley, and World Scientific. He was invited to chair several international conference/workshop programs and sessions. Dr. Misra was also invited to deliver keynote/invited lectures in over 30 international conferences in USA, Canada, Europe, Asia, and Africa. For more details, please visit http://cse.iitkgp.ac.in/∼smisra

Aishwariya Chakraborty (S'17) is presently pursuing her Ph.D. degree from the Advanced Technology Development Centre, Indian Institute of Technology Kharagpur, India. Her current research interests include algorithm design for sensor-cloud, service-oriented architecture, and wireless sensor networks. She received her M.S. and B.Tech. degree from the Department of Computer Science and Engineering, Indian Institute of Technology Kharagpur in 2019 and West Bengal University of Technology in 2015, respectively. She is also a student member of ACM. For more details, please visit https://cse.iitkgp.ac.in/∼aishchak.

Ayan Mondal (S'13-M'21) is an Assistant Professor at IIT Indore. Prior to this, he was a Postdoctoral Researcher at Univ Rennes, Inria, CNRS, IRISA, Rennes, France. He completed his Ph.D. degree from the Department of Computer Science and Engineering, Indian Institute of Technology (IIT) Kharagpur, India in 2020. He is a former TCS Fellow. He also received his M.S. and B.Tech. degree from IIT Kharagpur in 2015 and West Bengal University of Technology in 2012, respectively. His current research interests include algorithm design for data center networks, software-defined networks, sensor-cloud, edge/fog networks, smart grid, and wireless sensor networks. He is also a member of ACM. For more details, please visit https://ayanmondal.github.io

Dhanush Narayan Kamath holds a B.Tech Degree in Electrical Engineering from the National Institute of Technology, Surat. He interned at the SWAN Lab, Indian Institute of Technology, Kharagpur in Summer, 2018. His current research interests include applications of Decentralized Ledger Technology, Machine learning, Predictive Analytics and Computer Vision.