Computer Communications 222 (2024) 161–176

Contents lists available at ScienceDirect

Computer Communications

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

Blockchain and differential privacy-based data processing system for data security and privacy in urban computing

Gabin Heo [a], Inshil Doh [b,*]

[a] Division of Artificial Intelligence and Software, Ewha Womans University, Seoul, 03760, Republic of Korea
[b] Department of Cyber Security, Ewha Womans University, Seoul, 03760, Republic of Korea

ARTICLE INFO

Keywords:
Blockchain
Differential privacy
Urban computing
Data processing system
Data privacy

ABSTRACT

Recently, big data related to human movement, air quality, and meteorology have been generated in urban computing through sensing technology and the computing infrastructure. However, security problems arise as data utilization increases. If the sensing data from internet of things devices are constantly exposed, the users' private information can be determined, a critical security risk that could result in privacy breaches. This paper proposes a secure data processing system using the blockchain and differential privacy for data security and privacy protection in urban computing. When a service provider requests information, the system generates it from urban computing data using machine learning. We apply differential privacy to these data to protect privacy. However, if a query repeats, differential privacy may provide insufficient privacy protection. Therefore, we reduce the total privacy cost by reusing noise for the same data and privacy parameters using the blockchain. Machine learning accuracy may decrease when noisy data are used for training. Thus, we increase accuracy by storing and appropriately using the model parameters generated by the same data in the blockchain. We design, simulate, and analyze the results of an experimental environment for reusing noise for differential privacy and parameter utilization of machine learning using the blockchain. The proposed approach reduces privacy costs compared to the existing mechanism while protecting data privacy. We demonstrate that, through parameter utilization, the accuracy improves compared to conventional mechanisms.

1. Introduction

Urban computing details the collection and analysis of data generated within a city through sensors, devices, vehicles, and buildings to address the city's concerns, such as energy requirements and transportation [1,2]. Urban computing uses various internet of things (IoT) devices, and each IoT device generates various data types, forming a cyclic structure where other IoT devices adopt data generated for machine learning [3]. Therefore, data comprise a crucial resource in urban computing, and data accuracy is critical [4]. A higher number of IoT devices generate more data. Consequently, the system can eventually create accurate information because more data inform more accurate learning results. However, more data also increase the likelihood of degrading user privacy [5]. Particularly, IoT devices have become ubiquitous observers of users' lives, exposing private data to malicious attackers. Moreover, if IoT data are continuously leaked, a malicious attacker can obtain the user's sensitive information, such as behavior, habits, hobbies, or health status [6]. With the advancement of artificial intelligence technology, data privacy violation problems have become increasingly severe, and we must address security threats that could occur in artificial intelligence modeling. Currently, urban computing service providers (SPs) collect data for the convenience of users in cities and provide the data as information [7]. For instance, in a smart mobility system, such information as "Which roads have traffic jams?" and "Where is the accident risk high?" can be obtained through location and speed information generated by multiple vehicles [8]. With this information, SPs can build a more accurate and safer autonomous vehicle system.

However, even if SPs participating in the system have been authenticated, such problems as abuse of authority may occur [9]. Specifically, a malicious user who has stolen SP privileges can query and use data that are different from the service type of the real SP. If the malicious user continues to request such data, they could eventually access sensitive data on IoT devices in the system. To solve this problem, we adopted the blockchain. Specifically, blockchains ensure transparency, data integrity, and high security by allowing all participants to own a transaction information ledger. If the requested information of the SP is recorded in the blockchain as a transaction, the system can check all details of the data requested by the SP in the past. Subsequently, when

* Corresponding author.
E-mail addresses: gjrkqls@ewhain.net (G. Heo), isdoh1@ewha.ac.kr (I. Doh).

https://doi.org/10.1016/j.comcom.2024.04.027
Received 21 December 2023; Received in revised form 18 April 2024; Accepted 22 April 2024
Available online 24 April 2024
0140-3664/© 2024 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

the SP continues to request information besides the registered service type, the system can treat the identification (ID) of the SP as abnormal to reduce risk.

In addition to the blockchain, differential privacy (DP) has been widely explored as a data privacy method, which limits the information leakage about individuals in a dataset [10,11]. Particularly, statistical values are provided by adding noise within a limited privacy protection range, conserving statistical properties but making individual data challenging to identify [12]. However, the results could include individual privacy information [13]. If the same query is repeated on the same dataset, eventually, the response can become close to the correct information, after which the system can no longer protect privacy, reducing privacy levels. Thus, setting and managing a privacy budget that is not exceeded in the dataset is critical [14]. The existing data processing systems have been criticized for inadequate data privacy protection due to the use of raw data. Moreover, SPs must use large volumes of data to provide service, leading to a substantial demand for data by SPs. This results in the problem of high data communication costs during this process. We propose a data processing system that improves these problems by enhancing privacy protection and communication environments. Unlike the existing system that transmits data to the SP, the proposed system reduces communication costs associated with the transfer of large volumes of data by providing only the processed information based on the SP requests. Furthermore, each piece of information uses noisy data with an applied DP; thus, the proposed system also ensures data privacy. To achieve this, we employ the blockchain and DP. However, instead of simply applying them, we analyze and reduce the limitations of the blockchain and DP, proposing a secure data processing system suitable for urban computing.

In this work, we use an unmanned aerial vehicle (UAV) to process SP's requests in urban computing. Specifically, the UAV generates a response by receiving data from a base station (BS) in the appropriate area according to the SP request. Afterward, the UAV records the requested information as a transaction in the blockchain, and the transaction is appropriately used when processing requests from other SPs. The BS prepares appropriate data according to the UAV request. Then, the BS applies DP for data privacy. To avoid exceeding the limit of the privacy budget of each dataset, we perform noise reuse using transactions on the blockchain. We summarize the main contributions of our work as follows.

- We analyze the problems in urban computing for data security and privacy. When generating information using data collected by IoT devices, data privacy must be protected. The data that generate the information must be accurate because the SP must apply accurate information when providing services to users.
- We propose a noise reuse mechanism using the blockchain to lower privacy costs when applying DP for data privacy in urban computing. The data type, query content, and DP parameters requested by the SP are stored in the blockchain; thus, noise can be reused based on the same type. The total privacy cost can be decreased while maintaining data privacy through noise reuse.
- We propose a method of data-adaptive parameter utilization using the blockchain to improve the accuracy of the learning results in urban computing. We configure the blockchain of UAVs and SPs as a child_chain to ensure that we can automate the transaction creation and results returned for the SP request using a smart contract. Moreover, sensitive information, such as model parameters, is not disclosed to the SP. The UAV generates a result for each request and stores the model parameters along with their data types in the blockchain. When a new request arrives, the UAV queries the blockchain and locates transactions that are the same as the data for the new request. If a UAV finds a transaction, the model parameters in that transaction are used to process new requests.
- We simulate and analyze the proposed method. The proposed approach reduces the total privacy cost through noise reuse in DP and increases the accuracy of the results using data-adaptive parameter utilization.

The remainder of this paper is organized as follows. Section 2 introduces the background and related work on urban computing, blockchains, DP, and UAVs. Next, Section 3 describes the overall structure of the proposal, noise-reuse scheme, and data-adaptive parameter utilization using the blockchain. Then, Section 4 presents the simulation results. Section 5 discusses heterogeneous IoT, the scalability of blockchain, and different user privacy protection requirements. Furthermore, we analyze security in our proposal. Finally, Section 6 concludes the work by discussing the implications and future work.

2. Background and related work

This section explains urban computing, blockchains, DP, and UAVs, which form the background of the proposed system.

2.1. Urban computing

A myriad of IoTs in urban computing generate big data through sensors. Specifically, an IoT system learns and analyzes these big data to generate new information adopted within the urban system [15]. For example, we can determine where traffic is congested using data, such as global positioning system data, speed, and vehicle destinations [16]. Users can apply this information to avoid traffic-congested areas and reduce the risk of traffic accidents. During emergencies, this information can help determine the fastest and most appropriate route. Specifically, the system should ensure data integrity to provide necessary information to users as per the time and situation. If the data are inaccurate, information created by learning the data is also inaccurate, eventually degrading the performance of the urban computing system. Therefore, data accuracy should be ensured.

Existing urban computing continuously collects data from all devices, and a database stores these data suitable for each characteristic. In particular, data can leak during data collection and transmission, and malicious attackers can hijack sensitive data [17]. The SP executes a query to the database to obtain data to generate information adapted to the user's needs. In this case, if the SP is a malicious user, it can attempt multiple queries to obtain the sensitive data of a specific person. To prevent the exposure of the sensitive data of a specific person, we employ DP. However, the privacy budget limit for each data should not be exceeded in DP.

This work applies blockchain technology to control the privacy budget through appropriate noise reuse. The proposed system satisfies data privacy requirements using the DP parameter recorded in the blockchain. Furthermore, the proposed method improves the accuracy of the information using data-adaptive model parameters. The proposed system processes big data in urban computing into information and provides the information to users, ensuring the user convenience. Moreover, we establish a secure urban computing environment by utilizing blockchain and DP to enhance data security and privacy protection.

2.2. Blockchain

The blockchain is a distributed database that ensures data integrity and transparency by guaranteeing that all nodes maintain the same ledger. All transactions in the blockchain network are recorded in the ledger, and every participant verifies all transactions. Therefore, when someone modifies a transaction, they must modify all ledgers. This change is practically impossible; thus, the blockchain guarantees integrity. Because data integrity is critical in urban computing, applying the blockchain is a suitable solution. In IoT systems, research on the integration of blockchain technology for various purposes, such as malicious node detection and data traceability, has been ongoing in various

fields, such as the smart grid and medical fields [18,19]. Recently, the blockchain has been used in deep learning environments to reduce the communication overhead of IoT devices and efficiently offload computing [20]. The blockchain has also been combined in fields for large-scale data processing and safe data sharing [21,22]. Miao et al. [23] proposed a secure data-sharing mechanism with privacy protection for the IoT as a data provider by combining the blockchain and DP in federated learning. In a collaborative data-sharing system, all shared data are recorded on the blockchain, tracking the node participating in the data request task and data usage. Xie et al. [24] proposed a trusted execution environment and blockchain-supported IoT data-sharing architecture (TEBDS) to achieve data and identity security in IoT data sharing. In the TEBDS architecture, all user operations are recorded in the form of blockchain transactions, providing transparency and traceability. The blockchain has been actively studied in combination with other mechanisms in various applications owing to these characteristics [25].

However, because all nodes must have the same ledger, the block chain has a scalability problem [26,27]. Solutions to scalability problems are classified into five categories: on-chain, off-chain, side-chain, inter-chain, and child-chain solutions [28]. The on-chain solutions increase scalability by modifying only elements within the blockchain. Such a solution can involve simply increasing the maximum block size (i.e., big blocks) or separating signatures and transactions (i.e., Segregated Witness, commonly known as SegWit) [29,30]. The off-chain solutions process transactions outside of the blockchain. A typical example is Bitcoin's lightning network. The side-chain solution connects other blockchains to the side next to the main chain, and the nodes of the side chain are added to the nodes of the main chain. In the side chain, the transaction is only processed in each chain [31]. The inter-chain method enables communication between blockchains to exchange assets between them. Finally, the child-chain comprises the main and child chains [32]. Transactions are processed in the child chain and recorded in the main chain. Unlike the side-chain, the child-chain operates by linking transactions with the main chain; thus, security is increased through double recording. When applying the existing blockchain as is, each node can quickly confirm data through the ledger. However, all nodes own the same data; therefore, there is a risk that every entity can access sensitive data. We improved the configuration of the blockchain by structuring the ledger differently based on nodes. This enhancement allows for the advantages of real-time data verification, a characteristic of the existing blockchain. We also ensure that only trusted nodes have access to sensitive data.

This work applies the child-chain to automate the SP request and response while preventing the SP from checking other sensitive information. In particular, if a child-chain is employed, the size of the main chain can be reduced, and the child-chain can be divided by functions. The weakness of this approach is that the cost is high because the entire main chain must be traced to verify the child chain. However, in this proposal, only part of the transactions between the main and child chains are shared, and the remaining critical information is stored only in the main chain. Thus, the cost is not high compared to the existing child-chain mechanism because only the information that overlaps with the main chain must be verified when verifying transactions in the child chain. This work uses two blockchains. The first blockchain is shared by the BS and is used to store DP parameters. The second blockchain is the child chain shared by the SP and UAV. The child chain stores the parameters of the model created by the UAV at the request of the SP.

2.3. Differential privacy

Recently, the number and production speed of accessible data have increased with the development of big data and information technology. Furthermore, a database can be accessed easily with the development of the cloud environment. Users can obtain results by executing a query against a database. A malicious attacker can use the query results to obtain personal information from a specific person. Hence, DP has garnered research interest to overcome this concern. Attacks that reveal personal information using the query results cannot be prevented through encryption. Therefore, DP must be applied to provide statistical values so that users cannot infer data. Hence, DP limits information leakage by controlling the difference between two databases when they differ by only one object [33]. In DP, the parameter ϵ is measurable and verifiable and can be used to provide statistical values within a limited privacy protection range. The most common privacy protection method in DP is to add noise.

Definition 1 (ϵ-*DP*)**.** For adjacent datasets D and D' with only one record difference, given the privacy algorithm M, M satisfies ϵ-DP if Eq. (1) is satisfied for all outputs $R(R \in Range(M))$, where Pr denotes the probability.

$$Pr[M(D) \in R] \leq e^{\epsilon} \cdot Pr[M(D') \in R] \tag{1}$$

The noise required for each dataset depends on the sensitivity. Sensitivity parameterizes the amount of noise perturbation required for the DP mechanism [34]. The maximum possible change in the result is calculated to determine the sensitivity.

Definition 2 (*Sensitivity*)**.** For a function $f : D \rightarrow R^{dim}$, for any adjacent dataset D and D', the sensitivity of f is defined in Eq. (2).

$$\triangle f = \max_{D,D'} \|f(D) - f(D')\| \tag{2}$$

where $\| \sim \|$ represents the distance between $f(D)$ and $f(D')$, Dim denotes the query dimension, and R indicates the mapped real space [35]. After calculating the sensitivity, the result is perturbed by adding the noise generated by the data distribution. The most commonly employed distribution is the Laplace distribution [36]. Given a query function f, the output is computed as $f(D) + Laplace(0, \frac{\triangle f}{\epsilon})$, where the mean ($\mu$) of the Laplace distribution is zero, the scale is $\frac{\triangle f}{\epsilon}$, and $\triangle f$ is the sensitivity of the f function.

Furthermore, by adding noise with the actual dataset, the value that is semantically closest to the correct information is returned. Thus, the system can reduce privacy violations for specific individuals and preserve the meaning of the value. However, regardless of how randomly noise is added, the returned information may include some private information. If the users repeat the same question, they could gradually obtain personal information, and if they repeat the query several times, they could eventually obtain the correct information. Therefore, to prevent repeated queries, DP limits the query scope, called the privacy budget [37]. Zhao et al. [14] designed an algorithm to reuse noise to minimizing the accumulated privacy cost of DP. This algorithm reuses noise fully or partially for various instances of the same query type. The current query type is compared with the previous query type using the blockchain. If the current query type is the first time the blockchain encounters the query, it generally adds noise. If the current query type matches the previous query type, it compares the number of all previous and reused noise adding or calls to the server.

In an environment that employs numerous diverse data, such as urban computing, a privacy budget for each data point must be set, implementing a mechanism to lower privacy costs. Although applying traditional DP can protect the privacy of data, additional measures are necessary to reduce the overall privacy cost of the entire system. This work improves DP to protect data privacy and reduces the privacy cost of the system. Considering the urban computing characteristics where various data exist, we propose a method to set and reuse the usage epsilon(ϵ) according to the data characteristics for SP requests.

2.4. Unmanned aerial vehicle

The UAV has been applied in logistics, medicine, transportation, and military fields because of its high autonomy, portability, and relatively low operating costs [38]. Individuals and organizations also employ UAVs for prompt service recovery after disasters, data collection, and transmission in IoT networks due to their high mobility and ease of deployment [39]. When the UAV is appropriately deployed and operated, it can provide stable and cost-effective wireless communication solutions in real-world scenarios [40]. Many researchers expect the reception quality of UAVs to be much better than that of the existing ground BSs, providing stable real-time communication in wireless communication [41,42]. Moreover, UAVs can be used in various forms, such as aerial base stations, relaying, data gathering, and mobile edge computing (MEC) [43]. Many recent studies have integrated UAV and MEC. Specifically, MEC places a server close to the user to acquire user information and quickly perform big data calculations. When UAVs are used with MEC, the system can provide reliable, economical, low-latency, and efficient wireless communication services [44]. A BS based on the MEC server provides a seamless connection to the UAV. Therefore, the computing performance of the UAVs is guaranteed, such that they can offload the tasks of the ground users. As a highly mobile UAV provides services at a location close to users, the system can increase the overall information delivery speed.

This work employs UAVs with high mobility in urban computing to address the needs of SPs in real time and offload tasks. Additionally, the UAV can evaluate the accuracy of the training model by collecting real-time data from the appropriate area required by the SP. When a disaster occurs in an urban area, the communication environment can promptly be restored using a UAV, allowing users to obtain results quickly because the UAV is located close to the users.

3. Blockchain and differential privacy-based data processing system

This section discusses an urban computing system that guarantees data security and privacy using DP and the blockchain.

3.1. System design

This work proposes a blockchain-based system for real-time data processing and privacy protection in urban computing. Fig. 1 presents an overview of the proposed system, which provides services in urban computing. In the system, a BS exists in an appropriate district unit, and each BS manages the data of IoT devices within its communication range. The IoT devices periodically upload their data to the BS within the communication range. The BS collects real-time data from IoT devices and uploads the data to the cloud at specific times. The BS transmits only the data necessary for the SP request to the UAV, and the UAV performs task processing. The BS applies DP to the data to protect the privacy of the IoT devices. The DP parameters are stored in the blockchain. The UAVs with high mobility process SP requests. The SP requests information to provide convenient services to users in urban computing. When the SP creates a smart contract by specifying the required data, the UAV that meets the conditions manages the task. The SP and UAV share the UAV_chain to automate task processing through smart contracts. Even if all blockchain participants authenticate and participate in the network, we cannot completely trust SPs because internal attacks, such as privilege abuse, can occur. Therefore, we applied the child-chain and configured 5he block to ensure that the SP can only confirm the response to the request. The transactions including data type and DP parameters for each SP's request are recorded in the BS_chain. Since it is important to appropriately reuse noise to reduce total privacy costs when applying DP in the system, these records can be used for noise reuse in the next request. The BS can set appropriate DP parameters through ledger queries without additional communication with other entities via blockchain. Table 1 presents the roles and descriptions of the components of the proposed approach.

Table 1
System components and descriptions.

Components	Description
Unmanned aerial vehicle (UAV)	- UAVs form a UAV_chain and are assigned work (learning) from the BS in the mobile edge computing environment. - UAVs learn using the data from the BS and return the results.
Base station (BS)	- BSs form a BS_chain, communicate with the IoT, and upload data to the cloud. - BSs apply differential privacy for IoT data privacy. - BSs send data to the UAV.
Internet of Things (IoT)	- IoT refers to all IoT devices in urban computing. - IoT communicates through the BS, and all devices share the data generated by the IoT.
Cloud	- The cloud stores all blockchains built on the system, enabling data tracking to ensure data integrity. - Even if a denial of service attack occurs, the system can continuously provide service through the cloud.
Service provider (SP)	- The SP is a company that provides services with new information generated by IoT data from urban computing. - We assume that SPs can be semi-trusted.

Table 2
Description of notation.

Symbol	Description
I_m	The mth information request
I'_m	Result of I_m
D_m	Source data for I_m
D_{t_m}	Data type of I_m
$\tilde{D}_m$	Noisy dataset related to I_m
$Dist_m$	Districts related to I_m
SP_{m_ID}	ID of SP that requested I_m
ϵ_m	Epsilon for D_m
ϵ_q	Epsilon for the same D_{t_m} recorded in the blockchain
N_q	Noise from blockchain query
PR_q	Learning model parameter from blockchain query
α	Parameter for epsilon range scaling
β	Total privacy budget

Fig. 2 presents the general framework of urban computing [15]. In urban sensing (at the bottom of figure), the system continuously collects data using IoT devices. Afterward, in the data management stage, data are stored in a suitable database. Each database is configured with a structure suitable for the data characteristics to process the subsequent data analysis efficiently. In the data analysis stage, technology, such as data mining and machine learning, is used to generate information for service provision. Finally, in the service provision stage, a service is provided to users for convenience in urban computing based on such information. Based on this urban computing framework, we configured the IoT devices for the urban sensing phase, the cloud and BS for the urban data management phase, the UAV for the urban data analytics phase, and the SP for the service-providing phase. In summary, based on the urban computing framework, in the proposed system, lower-level IoT devices sense data and transmit it to the BS. The BS collects data from lower-level IoT devices over a specific time and transmits it to the cloud. The cloud stores and manages data according to the type. When the SP requests information to provide a service, a nearby UAV receives noisy data from the BS and performs data analytics. Then, when the UAV returns the results to the SP, the SP provides services to the user based on these results. Fig. 3 presents the overall process of the proposed system. Table 2 summarizes the notation adopted in the system. The following section discusses the specifics of the system.

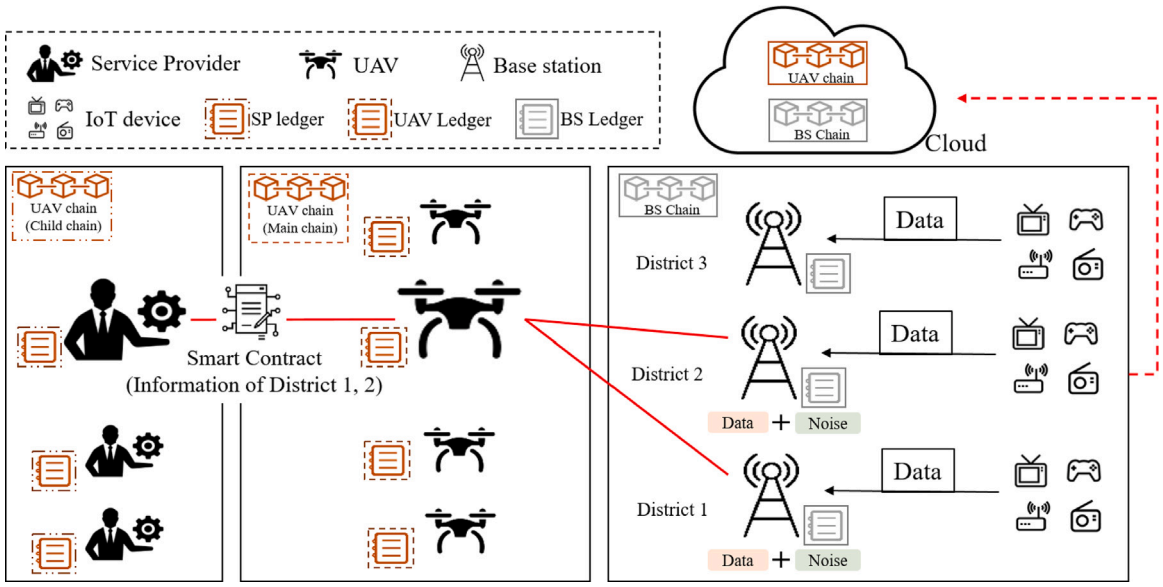


Fig. 1. Overall structure of the proposed system. Each entity constitutes a separate blockchain, and each node has a blockchain ledger.

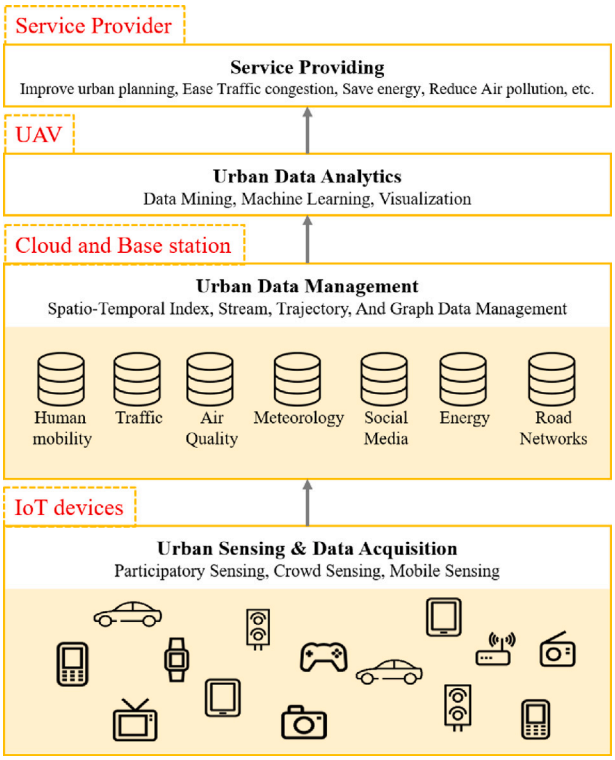


Fig. 2. Urban computing framework and components of the proposed system.

3.2. Blockchain configuration

This system employs two blockchains. First, the UAV_chain comprises the main and child chains. Specifically, the UAV can access the main chain, whereas the SPs can only access the child chain. Furthermore, we configured the UAV_chain as a child-chain to respond to the SP request automatically, preventing the SP from retrieving sensitive information. Fig. 4 presents the configuration diagram of the UAV_chain.

First, the SP sends a request for the information required through the smart contract. When the smart contract is executed, the UAV closest to the district requested by the SP performs the task. The UAV receives data from the district BS and generates the results requested by the SP. When the UAV responds to the results derived through the generated model as a smart contract, the UAV_chain creates a transaction according to the transaction configuration of the main and the child chains. Subsequently, the SP uses the result returned to the smart contract to provide the service. Table 3 presents the transaction structure of the UAV_chain. In the proposed system, the child_chain contains common data (no sensitive data) accessible to the UAV nodes and SP nodes. Moreover, it contains sensitive information that only UAVs can access. In the child_chain, consensus nodes consist of UAVs with access to all data, whereas SP nodes are configured to receive only common data for the generated transactions. As the SP connects to the network as a node in the child_chain, it can execute smart contracts to receive responses for requests. There is no need for additional communication between SP and UAVs for information transfer.

Table 3
Transaction components of an unmanned aerial vehicle (UAV)_chain.

Main chain of UAV_chain transaction information	
Field	Description
$Time$	Transaction occurrence time
UAV_{ID}	Transaction occurrence subject
SP_{ID}	SP ID of requested subject
$Data type$	Data type of SP's request
$Result_{data}$	Result of request (Learning results)
$Parameter_{model}$	Model parameter
$SP_reliability$	SP's abnormal behavior
Child chain of UAV_chain transaction information	
Field	Description
$Time$	Transaction occurrence time
SP_{ID}	SP ID of requested subject
$Data type$	Data type of SP's request
$Result_{data}$	Result of request (Learning results)

Second, the BS_chain comprises BSs and records DP parameters for reusing noise. The BS generates transactions for all SP requests and appropriately uses these transactions for later requests. Table 4 presents the transaction configuration diagram of the BS_chain for storing DP parameters. The BS_chain is structured similarly to the traditional blockchain. Consensus nodes consist of the BS, and all nodes possess the same ledger. Therefore, whenever a data request occurs, one can verify the appropriate parameters for applying DP through a blockchain

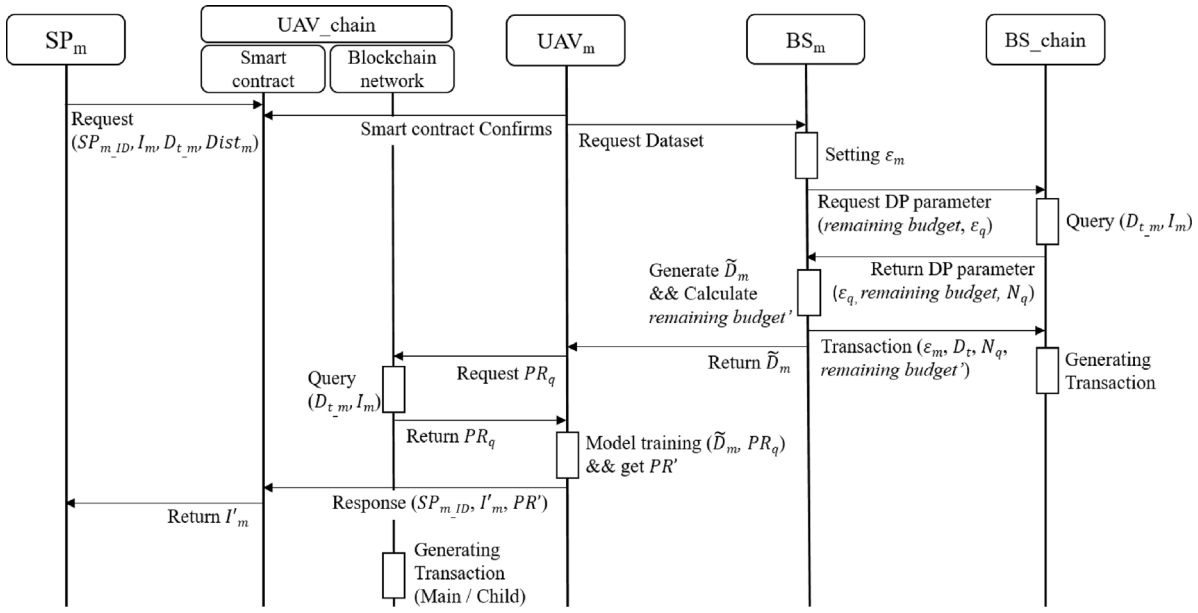


Fig. 3. Process for the overall proposed system for noise reuse and data-adaptive parameter utilization using the blockchain.

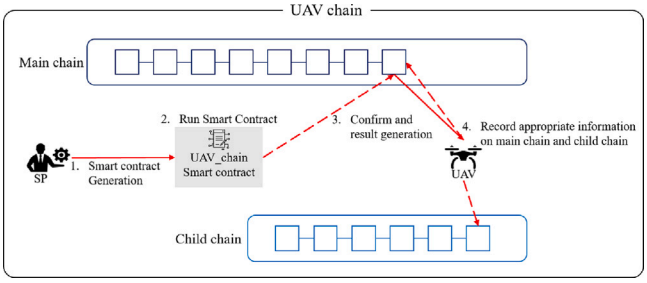


Fig. 4. Structure of the unmanned aerial vehicle (UAV)_chain consisting of the main and child chains. The service provider can generate a transaction using a smart contract.

Table 4
Transaction components of a base station (BS)_chain.

BS_chain transaction information	
Field	Description
$Time$	Transaction occurrence time
BS_{ID}	Transaction occurrence subject
UAV_{ID}	UAV ID that requested the data
$Data\,type$	Data type of SP's request
ϵ	Privacy parameter of request
$Noise$	Noise added to the data
$Remaining\,budget$	Privacy budget remaining for data with $data\,type$

query in real time. The time taken to query is negligible compared to the time required for block generation in the blockchain, allowing for swift responses when executing blockchain queries. Furthermore, by configuring the Child_chain, our system can restrict access to information for each entity, thereby solving the privacy issue that was pointed out as a limitation of existing blockchains.

3.3. IoT data security and privacy

Data are an essential resource in urban computing because they are collected from the IoT as learning data for other IoT devices in a circular structure. Data privacy protection is essential in circumstances in which the importance of data increases and the number of data increases enormously. Therefore, the integrated guarantee of data security and data privacy has become an essential requirement in all systems [45].

- *Data security:* The system should prevent hacking by external attackers to achieve security.
- *Data privacy:* The system should ensure that data are not accessed without permission or used to infer related information ensuring privacy.

This work performs SP authentication for data security and applies DP for data privacy.

Algorithm 1 presents the process for data security and privacy protection. The SP executes smart contracts by inputting data types, districts, and necessary information. When the smart contract is executed, the UAV in the corresponding district confirms the request. The UAV proceeds with SP authentication by querying the transactions of the blockchain in preparation for internal attacks, such as privilege abuse. Through the blockchain records, the UAV confirms the service type of the previous SP requests and information on abnormal behavior. If SP authentication is completed, the UAV forwards the request to the BS in the district ($Dist_m$) for data collection. Next, the BS applies DP to add noise into the data requested by the SP. The BS sets an appropriate value for epsilon (ϵ) using information that affects noise for the requested data. Then, the BS queries the blockchain to reuse noise and reduce the privacy cost of the dataset. The BS calculates the remaining budget according to the query results. Following this, the BS generates a noisy dataset and returns it to the UAV.

3.3.1. Service provider authentication

In this system, we assume that the SP and UAV participate in the system after completing authentication. The SP registers the service type and authentication when joining the blockchain network. The UAV checks the SP information recorded in the UAV_chain to verify that the requesting SP is a legitimate user. For example, if the SP registers to provide traffic information in the registration process, the information that the SP requests from the system should be closely related to the transportation service. If the SP account requires entertainment information or medical information unrelated to traffic, the system can determine that the request is inappropriate.

The SP authentication results are used to evaluate the reliability of the SP, which is applied as the epsilon setting for SP requests. If SP

Algorithm 1 Noise-reuse algorithm using the blockchain.

Input: D_{t_m}: Data type of SP_m's request, $Dist_m$: related District of the m-th request, I_m: The m-th information request, SP_m: The service provider that requested the m-th information request, SP_info: SP authentication results (abnormally behavior, number of requests for the same data), ϵ_m: epsilon of I_m

Output: $\tilde{D}_m$: Noisy datasets

SP_m executes Smart contract ← D_{t_m}, $Dist_m$, I_m
UAV_m confirms Smart contract
UAV_m → UAV_chain: Transaction query SP_{m_ID} (SP authentication)
UAV_m request D_m(Data of D_{t_m}) to BS_m ← I_m, D_{t_m}, SP_info
BS_m sets the ϵ_m about I_m ← SP_info, D_m
BS_m → BS_chain: Transaction query (I_m, D_{t_m})
if D_{t_m} is seen for the first time **then**
  BS_m sets Privacy Budget of D_{t_m}
  BS_m generates noise ← DP(ϵ_m)
  BS_m computes Remaining budget ← $\beta - \epsilon_m$
else
  if remaining budget < ϵ_m **then**
    return error of insufficient remaining budget of D_{t_m}
    Resetting β of D_{t_m}
  end if
  BS_chain returns ϵ_q, remaining budget, N_q ← blockchain query(D_{t_m}, I_m)
  if $\epsilon_m = \epsilon_q$ **then**
    BS_m using noise N_q
  else
    BS_m generates noise ← DP(ϵ_m)
    BS_m computes remaining budget ← remaining budget - ϵ_m
  end if
end if
BS_m generates $\tilde{D}_m$ ← D_m + noise
BS_m → BS_chain: Transaction generation (D_{t_m}, ϵ_m, remaining budget)
return $\tilde{D}_m$ to UAV_m

reliability is high, the epsilon value is set high, adding less noise to increase the accuracy of the results rather than protect data privacy. In contrast, if the SP reliability is low, privacy protection of the IoT data is a priority; thus, the epsilon value is set low, incorporating more noise.

3.3.2. Differential privacy for reducing total privacy cost

We can avoid wasting the privacy budget by reusing noise when a request from an SP uses the same data (Algorithm 1). The BS that receives the data request from the UAV sets the epsilon value using the SP_info and data features. For this purpose, the value of ϵ_m for each request is calculated using Eq. (3).

$$\epsilon_m = \frac{SP\,reliability}{Request\ number} \cdot \alpha \tag{3}$$

As the number of requests for the same data increases, privacy protection decreases, so the value of epsilon must be lowered to add more noise. The term $SP\,reliability$ refers to the trust score calculated from the SP authentication results, which has a value between 0 and 1 and decreases as the number of records of SP abnormal behavior increases. Additionally, the epsilon range is adjusted according to the volume of data requested by the SP because this affects epsilon. We used the error rate to determine the appropriate epsilon range. The error rate refers to the difference between the actual correct and noisy answers. If the error rate is too large, data accuracy decreases, whereas if the error rate is too low, privacy protection is reduced. Therefore, determining an appropriate error rate when adding noise is critical. Fig. 5 presents the average error rate measured by varying the epsilon

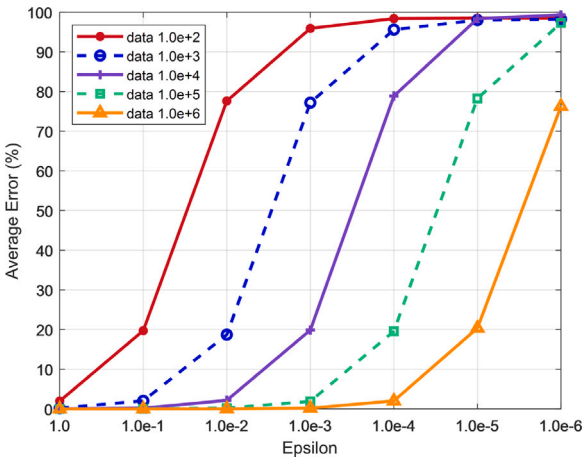


Fig. 5. Average error rate by number of data and epsilon value.

value depending on the number of data. The section can be checked in which the error rate rapidly changes depending on the number of data. For example, if the number of data is 1e+4, the error rate changes from about 20% to 80% when the epsilon value is between 1e−3 and 1e−4. Based on this, the corresponding section can be set to an appropriate epsilon value. Therefore, we set epsilon to an appropriate range considering each level of data. In other words, ϵ_m is calculated by multiplying the parameter α for epsilon range scaling. Although this simulation was conducted with few data, an appropriate epsilon range can be determined even in situations with numerous data. If data are numerous, the epsilon range is smaller.

After epsilon is set, the BS checks the BS_chain records to determine whether a request exists that is identical to the data. If the data type of the request is new, the BS sets a privacy budget for the new data and calculates the remaining budget by generating new noise. Moreover, if the data type has been previously requested, the BS compares the remaining budget to ϵ. If the remaining budget is less than ϵ, it returns an error message because no more noise can be added to the data. Then the BS resets the privacy budget for the data. Next, the BS searches for transactions with the same data. If the same data request record reoccurs and ϵ is the same, the BS reuses the noise of the transaction without additional calculations.

The requirement to add noise arises in two scenarios: when there is a new type of data request, and when there is no record of the same epsilon in the blockchain. In this case, when epsilon is set, sensitivity is calculated. The sensitivity as the maximum change can be set according to the data. Therefore, upper and lower bounds are set using the maximum and minimum values of the data. When the upper and lower bounds are close, the sensitivity decreases. This means that the maximum change amount is small. For example, in a dataset with values ranging from 0 to 100, the maximum difference between the lower and upper bounds could be 100, indicating a potential maximum change of 100. In contrast, if there is a dataset with values ranging from 0 to 1, the maximum difference between the lower and upper bounds could be 1, indicating a potential maximum change of 1. In the former case, where the maximum change is greater than in the latter case, more noise could be added, implying a decrease in the data accuracy. Therefore, sensitivity must be set appropriately depending on the dataset. After setting the sensitivity ($\triangle f$) by setting the bounds according to the dataset, we can set and add noise using the Laplace distribution with epsilon.

Using the same data means that the data distribution that determines sensitivity is the same, and the same ϵ indicates that the restrictions on how much privacy is guaranteed for the data are the same. Therefore, this approach allows privacy protection without using the budget to add noise into the same data. When the BS creates dataset

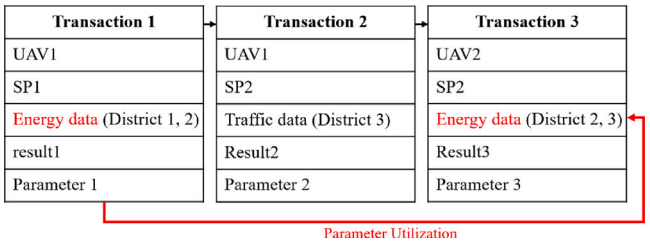


Fig. 6. An example of transaction structure for data-adaptive parameter utilization.

$\tilde{D}$, the BS creates a new transaction on the BS_chain. Afterward, the BS returns $\tilde{D}$ to the UAV. Our system reduces the total privacy cost through noise reuse while protecting data privacy by adding noise. Furthermore, we can reduce communication overhead through blockchain because additional communication with other entities is not required for setting parameters.

3.4. Data-adaptive parameter utilization for data accuracy

Urban computing requires data accuracy and security. The accuracy of information is crucial for SPs that provide critical services such as emergency services. We generated results using data from districts required by the SP to increase the accuracy of the results. In addition, we used the information recorded in the UAV_chain to provide the SP with learning results from various data.

Fig. 6 presents an example of data-adaptive parameter utilization. In machine learning, systems require considerable data to increase the learning accuracy. When learning only the data requested by the SP for a one-time query, accuracy is limited. As the UAV is responsible for machine learning in our proposal, a resource problem exists when uploading considerable data to the UAV. Therefore, if a learning result exists for the same data in the blockchain, the model parameter from that transaction can be used, and training can proceed.

Algorithm 2 Proposed algorithm for data-adaptive parameter utilization using the blockchain.

Input: D_{t_m}: Data type of SP_m's request, I_m: The m-th information request, SP_m: The service provider that requested the m-th information request, $\tilde{D}_m$: Noisy dataset
Output: I'_m: Result of I_m

$UAV_m \rightarrow$ UAV_chain: Transaction query (D_{t_m}, I_m)
if D_{type_m} is recorded on the UAV_chain and I_m is recorded on the UAV_chain **then**
    $PR = PR_q \leftarrow$ returned in UAV_chain
else
    $PR =$ None
end if
UAV_m generation deep learning model using PR, $\tilde{D}_m$
UAV_m gets PR', I'_m
$UAV_m \rightarrow$ UAV_chain: Transaction generation (PR', D_{t_m}, I'_m, SP_m)
Smart contract records the operation related transaction in UAV_chain (main chain and child chain)

Algorithm 2 presents the algorithm for the data-adaptive parameter utilization using the blockchain. After obtaining data from the BS, the UAVs start learning. The UAV queries the UAV_chain before starting to learn. If a record of the data type requested by the SP exists in the blockchain, the blockchain returns the model parameters stored in that transaction. Moreover, if there is no record of the same data type in the blockchain, the parameter is set to none and returned. Then, the UAV executes machine learning using the parameters received from the blockchain and $\tilde{D}$ received from the BS. The learning process proceeds similarly to conventional algorithms. Various learning algorithms such as the deep neural network (DNN), convolutional neural network (CNN), and others can be employed as the learning model. Before training, model configuration with appropriate parameter values is conducted to facilitate the learning process. In the case of the UAV, it is a dynamic entity; therefore, the system specifies the minimum accuracy and maximum learning time for capacity efficiency. The UAV ends learning when the minimum accuracy is reached within the set learning time. After learning, the UAV obtains new parameters. When the UAV responds to the SP request and new parameters with the smart contract, the smart contract returns a response to the SP. Finally, the UAV_chain records the response appropriately according to the transaction structure of the main and child chains. We propose a data processing system that safely processes data in urban computing systems comprising various IoT devices. Additionally, we considered the accuracy of the learning results while ensuring data security and privacy in urban computing.

4. Performance evaluation

4.1. Experimental settings

For our experiments, we used Intel Core i7-10700 64-bit CPUs @2.90 GHz and 64 GB of RAM, NVIDIA GeForce RTX 2060 SUPER GPU on Windows 10. We used the Diffprivlib v0.6 library provided by IBM to apply DP [46]. We also employed TensorFlow (version 2.10.0) and scikit-learn (version 1.1.2) in Python (version 3.8) for data-adaptive parameter utilization accuracy.

We used traffic, IoT packets, and IoT sensor datasets to simulate an urban computing environment [47]. As the traffic dataset, We used the United States accident dataset. This dataset is a car accident dataset covering 49 states in the United States. We also used the IoT device identification dataset as the IoT packet dataset [48]. This dataset contains data on packets transmitted by various IoT devices. In addition, we used the human activity dataset as the IoT sensor dataset [49]. This dataset contains sensing data on user activities using the built-in accelerometer and gyroscope in a smartphone. In addition, we simulated using modified national institute of standards and technology (MNIST) and Fashion MNIST (FMNIST). The simulation results are expressed as the results of the traffic dataset, and the results for the other datasets are presented in tables. Our system uses the proof of work (PoW) consensus algorithm in the blockchain and connects blocks by generating a hash value through the secure hash algorithm 256-bit (SHA-256) function [50]. PoW is a representative consensus algorithm for public blockchain. The person who first resolves a specific problem through computer computation is granted the right to create a block. The algorithm can adjust the block generation speed according to the difficulty level. In addition, because it is an algorithm that efficiently achieves consensus in a multi-node environment, PoW is suitable for application to urban computing where various nodes exist.

Table 5 presents the simulation environment. First, to simulate noise reuse in DP, we set the privacy budget of the dataset and subtracted epsilon as the privacy cost when performing the query. For each query, we randomly set an epsilon value between 0.01 and 0.1 and set the total privacy budget of a dataset to 10. In addition, we confirmed the error value that occurs when noise is reused using the naive Bayes and logistic regression models. We applied the root mean squared error (RMSE) and mean absolute error (MAE) to check the error values. More precisely, the RMSE is a value obtained by calculating the square root of the mean squared error (MSE), which is the sum of the squared area of the difference between the predicted and actual values of the model [51]. In other words, the RMSE is a method of penalizing a high error value. Conversely, the MAE is the average of the absolute values of the errors, which is the difference between the predicted and actual values. Moreover, the MAE has the advantage of being less sensitive to outliers than the MSE. Results using the RMSE are typically expressed

Table 5
Simulation environment.

Neural network	
Number of hidden layer	3~5
Number of neuron in hidden layers	128
Number of convolutional layers	3~5
Number of max pooling layers	3~5
Activation function	ReLU
Optimizer	Adam
Loss function	Categorical cross-entropy
Learning rate	0.001~0.01
Library version	
Diffprivlib	0.6
TensorFlow	2.10.0
Scikit-learn	1.1.2
Blockchain	
Consensus algorithm	Proof of work
Difficulty	2~10
Hash algorithm	SHA-256

in a graph, and those for the MAE are in a table. Further, the RMSE and MAE are expressed in the following equations, where y denotes the predicted value, $\hat{y}$ represents the actual value, and n is the number of data.

$$RMSE = \sqrt{\frac{\Sigma(y-\hat{y})^2}{n}} \tag{4}$$

$$MAE = \frac{\Sigma|y-\hat{y}|}{n} \tag{5}$$

Next, we used a DNN and a CNN for data-adaptive parameter utilization. We adjusted the number of layers to suit the characteristics of the dataset. The DNN model comprises five to seven layers: an input layer, three to five hidden layers, and an output layer. We used rectified linear units (ReLU) as the activation function and categorical cross-entropy as the loss function. The CNN model consists of three to five convolutional layers, three to five max-pooling layers, and a fully connected layer.

4.2. Experimental results of the noise reuse mechanism for IoT data security and privacy

We compared the total privacy cost when reusing noise in DP. We set the overall privacy budget to 10 and queried each dataset. The privacy parameter for each query has a value of between 0.01 and 0.1. The urban computing environment has several types; thus, we assumed that each dataset was stored in a different database, and we performed the simulation. In DP, the level of privacy protection applied to respond to queries is referred to as the privacy cost, which should be managed to avoid exceeding the privacy budget. Even as the number of queries increases, a system with low privacy costs ensures better privacy protection.

Fig. 7 presents the total privacy cost according to the number of queries. In the figure, the naive mechanism is assumed to be the total database containing all data. Consequently, the privacy cost of the naive scheme increases with the increase in query numbers, regardless of the difference in data type and ϵ value. Conversely, the proposed scheme builds a different database according to the data type and reuses noise appropriately according to the same ϵ value, and the privacy cost is relatively low. Fig. 8 presents the privacy cost when applying DP to each data type. In both methods, the total privacy cost increases as the number of queries increases. However, in our proposed scheme, the total privacy cost is less than that for the naive method because the noise is reused when the data type and privacy parameter values are the same. Fig. 9 shows the total privacy cost of each data type. When the query number is small, not much difference occurs in the privacy cost compared to the naive method. However, as the query number increases, the difference in total privacy cost increases between the proposal and naive scheme because the probability that a query with the same privacy parameter covers the same data also increases. This result is because the corresponding information is recorded on the blockchain, and noise is reused. We confirmed that the proposed system has a lower total privacy cost than the integrated data based systems or the separated data based systems. In other words, the proposed system can reduce the total system privacy cost while protecting data privacy. In particular, urban computing collects data from various devices, resulting in a diverse range of data types and a vast amount of data. We expect that our proposed system performs even better in urban computing, where data types are diverse.

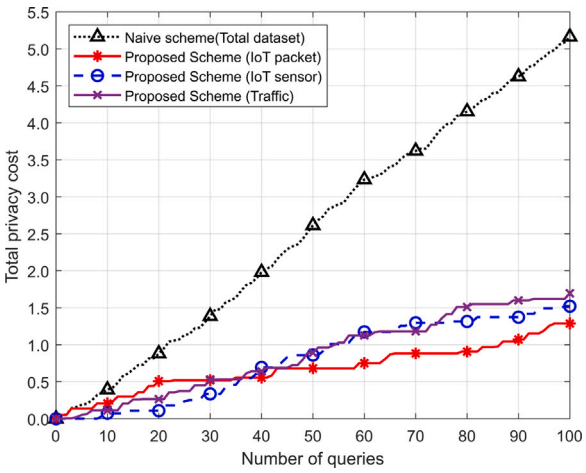


Fig. 7. Total privacy cost by number of queries. The naive scheme is the total database in which all three data types are stored.

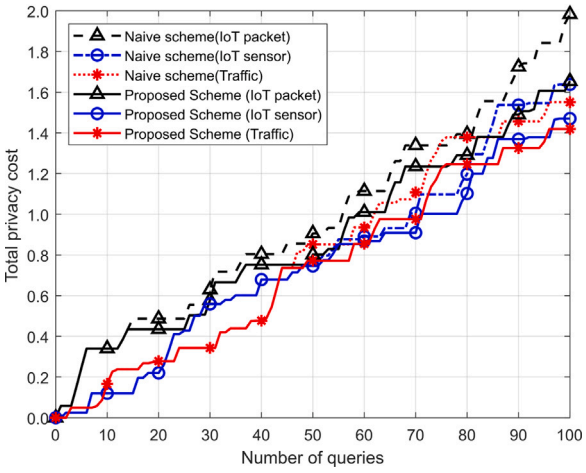
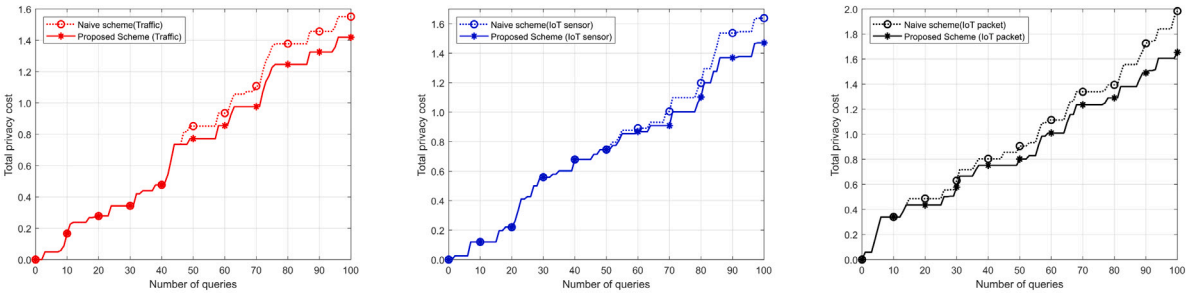


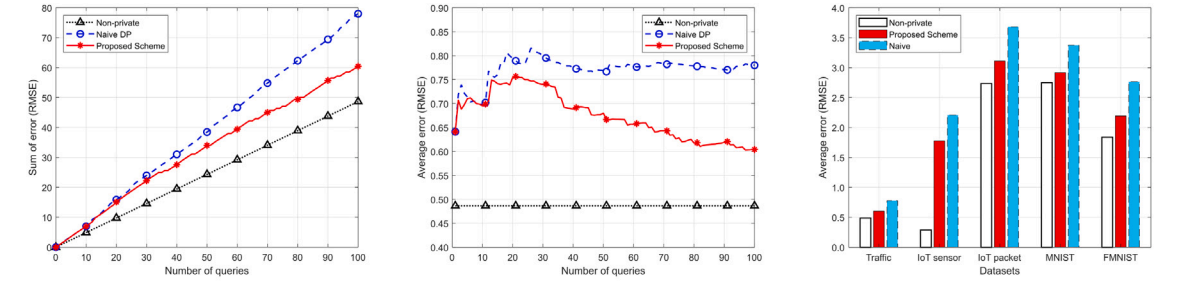
Fig. 8. Total privacy cost by the number of queries. The naive method has a different database for each data type.

Using DP adds noise to the data, lowering the data accuracy and resulting in a relatively higher error rate. Reducing the error rate while protecting privacy is crucial. Therefore, we measured and compared the system error rates of naive DP and the proposed mechanism. Next, we used the RMSE and MAE to confirm the error rate of the proposed system. We applied naive Bayes and logistic regression as learning models. The naive Bayes method can confirm fast results with low accuracy. Conversely, logistic regression has high accuracy but is relatively slow. Further, we compared the non-private method that does not apply the privacy scheme, existing DP, and our proposal for noise reuse.

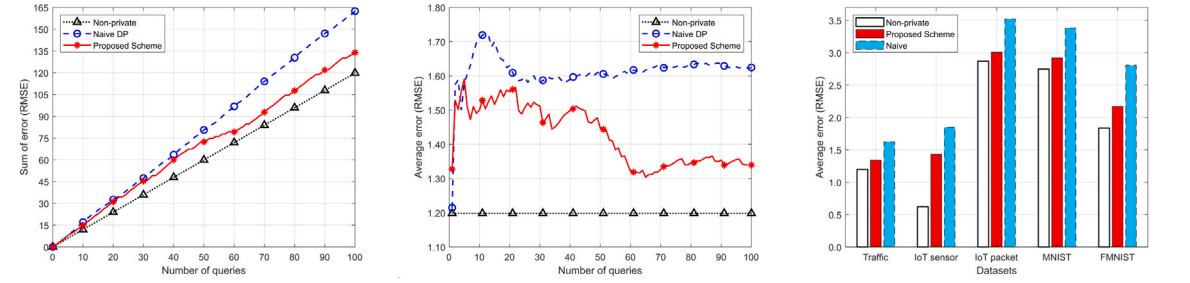


(a) Traffic dataset. (b) IoT sensor dataset. (c) IoT packet dataset.

Fig. 9. Total privacy cost of each dataset.



(a) Total error value by number of queries using logistic regression. (b) Average error value by number of queries using logistic regression. (c) Average error value by datasets using logistic regression.



(d) Total error value by number of queries using naive Bayes. (e) Average error value by number of queries using naive Bayes. (f) Average error value by datasets using naive Bayes.

Fig. 10. Error value using root mean squared error (RMSE).

Fig. 10 (a) shows the sum of errors using the RMSE in the logistic regression model. Logistic regression guarantees relatively high accuracy; therefore, the error of the non-private model is low. When learning by applying DP to the data, the accuracy is lower; thus, the error rate is relatively high. Nevertheless, the proposed method has fewer errors than the naive DP method. Fig. 10 (b) shows the average error values in the logistic regression model. The value is unstable because it is affected by random values using a random epsilon for each request. However, as the queries increase, the average error value becomes constant. The non-private model and naive DP maintain similar error values even when the number of queries increases. However, the proposed scheme can confirm that the average error value decreases as the queries increase because more noise reuse occurs as the number of queries increases; hence, additional errors for learning do not occur. Fig. 10 (c) shows the average error rate by dataset in logistic regression. As sensitivity to noise in each dataset varies, the error rate is measured differently for each dataset. However, our proposed scheme has a lower error value than the naive DP in other datasets. Fig. 10 (d) shows the sum of errors using the RMSE in the naive Bayes model. We used the naive Bayes method; hence, the non-private model also has a high error. Furthermore, if DP is applied to this model, the accuracy is lower, increasing the errors. Nevertheless, the proposed method has fewer errors than the naive DP method. Fig. 10 (e) shows the average error values in the naive Bayes model. Similar to the logistic regression model in Fig. 10 (b), non-private and naive DP maintain similar error values even when the number of queries increases. However, the proposed scheme can confirm that the average error value decreases. Thus, our proposal can lower the average error value of the system even if different models are used. Fig. 10 (f) shows the average error rate according to the dataset in the naive Bayes method. Similar to the logistic regression model, our proposed scheme has lower error values than naive DP, even on various datasets.

Table 6 presents the results of measuring the total and average error rates according to the number of requests using the MAE and RMSE. The MAE has a lower error value than the RMSE because it is less sensitive to outliers than the RMSE. Nevertheless, the proposed scheme for both metrics has a lower error rate than the existing DP. In addition, even in the naive Bayes model with a higher error rate, our proposal has a lower error rate than the naive DP. The average error is lower as the number of requests increases, regardless of the model, because the number of noise reuses increases, reducing additional errors. Additionally, Table 7 presents the average error rate according to the dataset. Regardless of the number of queries and the dataset, the proposed scheme has better results than the existing DP.

Many systems utilize data with added noise through DP for privacy protection. These systems inevitably prone to higher error rates compared to systems without privacy protection mechanisms. Our work

Table 6
Total error and average error by number of queries using the mean absolute error (MAE) and root mean squared error (RMSE) in logistic regression and naive Bayes.

Result	Model	Number of queries	Non-private		Naive DP		Proposed scheme	
			MAE	RMSE	MAE	RMSE	MAE	RMSE
Total error	Logistic regression	20	4.70	9.73	11.56	15.88	10.73	15.09
		40	9.41	19.46	22.29	31.00	19.25	27.54
		60	14.11	29.19	33.76	46.67	27.34	39.41
		80	16.46	34.05	39.69	54.82	31.22	45.01
		100	23.51	48.64	56.17	77.98	41.90	60.42
	Naive Bayes	20	16.83	23.96	26.11	32.60	24.71	31.07
		40	33.66	47.93	50.75	63.58	47.70	59.94
		60	50.50	71.89	77.40	96.76	63.43	79.24
		80	67.33	95.85	104.45	130.37	86.14	107.67
		100	84.16	119.82	130.08	162.45	107.08	133.93
Average error	Logistic regression	20	0.235	0.486	0.578	0.794	0.537	0.754
		40	0.235	0.486	0.557	0.775	0.481	0.688
		60	0.235	0.486	0.563	0.778	0.456	0.657
		80	0.235	0.486	0.567	0.783	0.446	0.643
		100	0.235	0.486	0.562	0.780	0.419	0.604
	Naive Bayes	20	1.198	1.198	1.305	1.630	1.236	1.554
		40	1.198	1.198	1.269	1.590	1.193	1.498
		60	1.198	1.198	1.290	1.613	1.094	1.366
		80	1.198	1.198	1.306	1.630	1.077	1.346
		100	1.198	1.198	1.301	1.625	1.071	1.339

Table 7
Average error value by dataset using the mean absolute error (MAE) and root mean squared error (RMSE) in logistic regression and naive Bayes.

Model	Datasets	Non-private		Naive DP		Proposed scheme	
		MAE	RMSE	MAE	RMSE	MAE	RMSE
Logistic regression	Traffic	0.235	0.486	0.562	0.780	0.419	0.604
	IoT sensor	0.052	0.287	1.761	2.208	1.383	1.775
	IoT packet	2.016	2.735	3.040	3.681	2.582	3.110
	MNIST	0.268	1.127	2.981	3.825	2.433	3.151
	FMNIST	0.480	1.417	2.770	3.562	2.205	2.883
Naive Bayes	Traffic	0.842	1.198	1.301	1.625	1.071	1.339
	IoT sensor	0.269	0.621	1.438	1.850	1.105	1.434
	IoT packet	1.615	2.872	2.761	3.522	2.373	3.007
	MNIST	1.653	2.748	2.509	3.383	2.112	2.919
	FMNIST	1.053	1.837	1.995	2.807	1.466	2.173

proposes a blockchain-based noise-reusing DP mechanism, which can protect data privacy while reducing error rates. In naive DP, when users repeat the query several times, they can eventually obtain the correct answer. When the same request is received, the system first checks the blockchain, and if a transaction exists for the same request, it reuses the noise of that transaction. The noise recorded on the blockchain is the result of limited information leakage of the actual data. Therefore, we expect privacy to be guaranteed because this noise can prevent the actual data from being deduced. Furthermore, the privacy risk is reduced because the system only handles requests from authenticated SPs. However, as the DP provides a statistical value by adding random noise, the noise first recorded in the blockchain ledger may be a value that lowers the data utility. Then, the data utility of other requests reusing that noise also decreases. We aim to consider the data utility problem for recorded noise in the blockchain before reusing noise in future work.

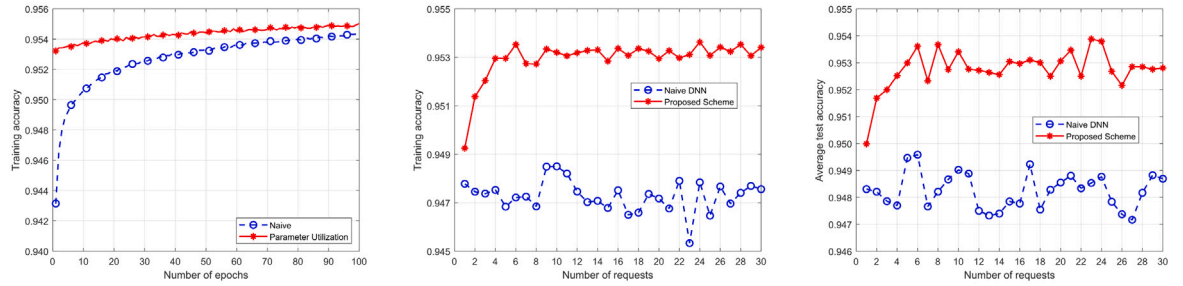
4.3. Experimental results of data-adaptive parameter utilization for data accuracy

We used the blockchain for the proposed data-adaptive parameter utilization. When a request arrives, the blockchain is queried, and if a transaction exists for the same data, the parameters stored in that transaction are used for new training. We used the DNN and CNN as learning models. In the environment considered in this work, as each SP requests data for the desired districts, we assumed each request has a different dataset. The simulation generated one model per request. If many requests occur, many models are generated. However, we tested only 30 requests owing to the limitations of the memory in the graphics processing unit.

Fig. 11 (a) shows the training accuracy by epoch when using the DNN model. Both methods show good accuracy, but with parameter utilization, the training accuracy quickly achieves over 95% performance with fewer epochs. Fig. 11 (b) shows the training accuracy in DNN by the number of requests. Both methods newly learn a DNN model for each request. As each request has different data, the naive DNN has a training accuracy of approximately 94.8% each time the request is repeated. In contrast, our proposal uses the parameters stored in the blockchain in case of the same data type, increasing the training accuracy as the number of requests increases. Fig. 11 (c) shows the test accuracy when using the DNN. Each request generates a new learning result; thus, the value appears randomly depending on the requests. However, the proposed mechanism performs better than the naive model because it uses parameters appropriately. Fig. 11 (d), (e), and (f) show the accuracy of the CNN model. Similar to the DNN, the proposed mechanism has better accuracy than the naive model.

The traffic dataset has numerous data; thus, if preprocessed well, even a basic model can exhibit good performance. Nevertheless, our scheme guarantees higher performance than previously. Thus, high performance is achieved with less learning under the same conditions. Therefore, our proposal is expected to be suitable for dynamic devices where deep learning models should not be heavy and cannot be learned for a long time.

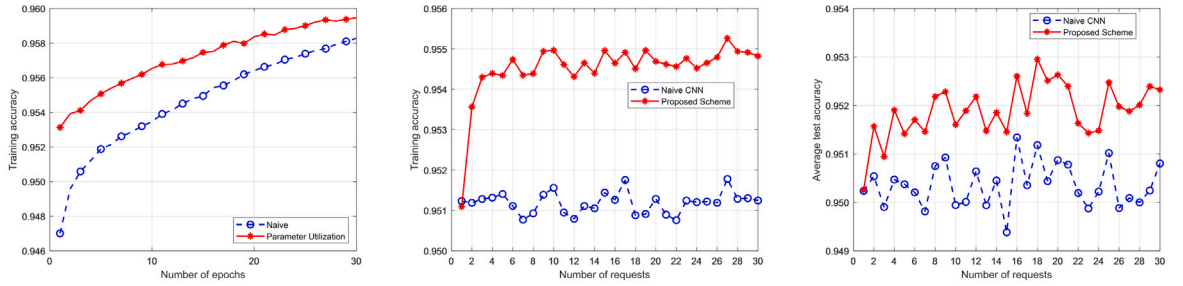
Furthermore, we simulated multiple datasets for each model, and we present the results when the request number is 30. Fig. 12 shows the accuracy of the method for each dataset using the DNN. The proposed method has higher accuracy for all tested datasets. Table 8 shows the



(a) Training accuracy of the deep neural network (DNN) by epoch.

(b) Training accuracy of DNN by request number.

(c) Testing accuracy of DNN by request number.



(d) Training accuracy of convolutional neural network (CNN) by epoch.

(e) Training accuracy of CNN by request number.

(f) Testing accuracy of CNN by request number.

Fig. 11. Training and testing accuracy for each learning model.

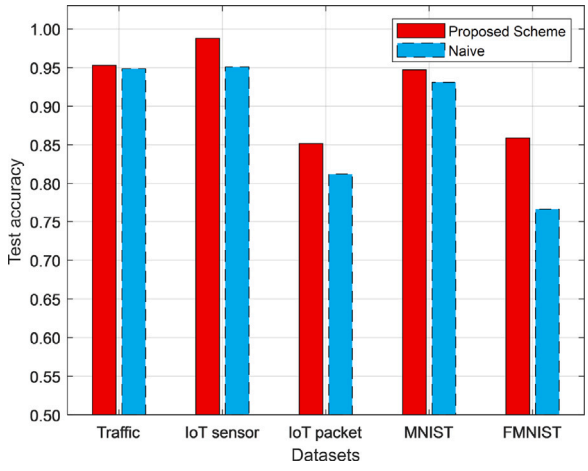


Fig. 12. Testing accuracy for deep neural network by dataset.

Table 8
Testing accuracy by dataset using the deep neural network (DNN) and convolutional neural network (CNN).

Model	Datasets	Naive model	Proposed scheme
DNN	Traffic	0.950	0.952
	IoT sensor	0.951	0.988
	IoT packet	0.812	0.852
	MNIST	0.931	0.947
	FMNIST	0.767	0.859
CNN	Traffic	0.951	0.952
	IoT sensor	0.912	0.987
	IoT packet	0.811	0.880
	MNIST	0.976	0.993
	FMNIST	0.844	0.870

test accuracy using various datasets. For both DNN and CNN models, the proposed scheme performs better in various datasets. In our proposal, we used the blockchain for data-adaptive parameter utilization.

Fig. 13 shows the average execution time of the model creation for requests in each model. The proposal incurs additional overhead because it queries the blockchain before processing each request and creates a transaction after processing. In the DNN, the time is approximately 3.9 s, but in the CNN, a time overhead of approximately 6.6 s occurs. In the blockchain, time is closely related to the consensus algorithm. Therefore, we measured the average execution time according to the PoW difficulty. The figure is shown based on DNN, and the CNN results are presented in a Table 9. Fig. 14 presents the average execution time by PoW difficulty in the DNN model. We confirmed a difference of approximately 1.7 s when the difficulty value is 8 and approximately 3.9 s when the difficulty value is 10. Fig. 15 presents the average execution time by dataset. When the PoW difficulty is 1, the DNN with the blockchain has little additional overhead time compared to that without the blockchain. In contrast, when the difficulty is 10, the overhead time is high. The traffic dataset is vast; thus, there are too many features compared to other datasets, requiring a long time to learn. We did not present the traffic dataset results in a graph because the value range is extensive compared to the other datasets. These values can be checked in the table. Table 9 lists the average DNN and CNN model execution times by PoW consensus difficulty in the dataset.

Table 10 shows the average latency for transaction processing and queries in the blockchain, which is the result when 30 nodes process 50 transactions in a blockchain that uses the practical byzantine fault tolerance (PBFT) and PoW consensus algorithms, respectively. PBFT is a consensus algorithm used in private blockchains, ensuring trust to achieve successful consensus even if there are a certain number of fault nodes in the network. PBFT is fast, and it allows for consensus in asynchronous networks, but communication overhead is high because the primary node communicates with all nodes. The transaction processing time in the blockchain has a significant influence on the overall system execution time, whereas the query latency is considered negligible. In the proposed system, conducting blockchain queries for data privacy protection and learning accuracy introduces additional query latency for every request. However, the latency of queries does not influence the overall system performance. Furthermore, we did not upload heavy data to the blockchain; hence, the increase in processing latency is not substantial. Therefore, this method is expected to be well-suited

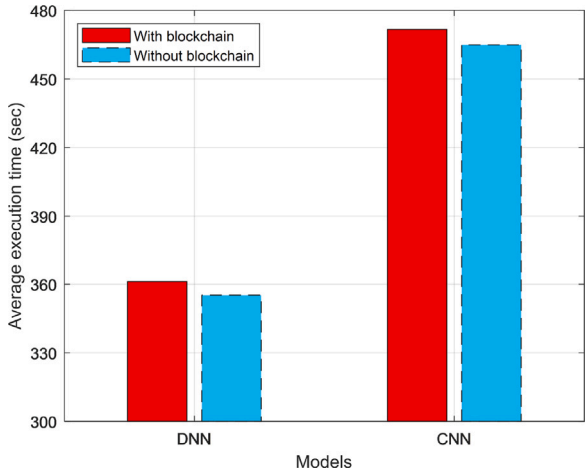


Fig. 13. Average execution time for each model. The proof of work difficulty in the blockchain is 10.

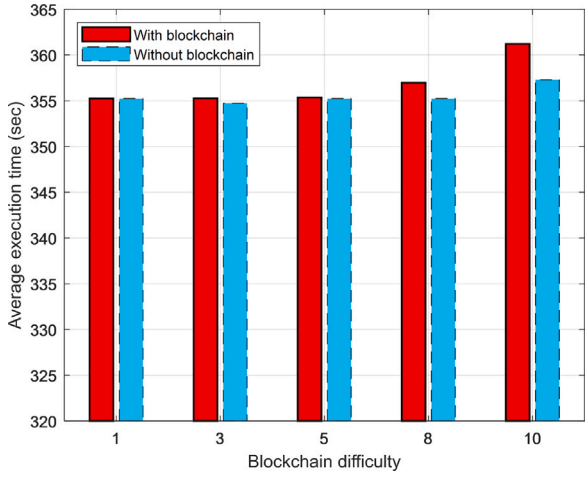


Fig. 14. Average execution time by blockchain difficulty in the deep neural network.

Table 9
Average execution time (s) by dataset using the deep neural network (DNN) and convolutional neural network (CNN).

Model	Datasets	Average execution time (s)			
		Difficulty = 1		Difficulty = 10	
		With BC	Without BC	With BC	Without BC
DNN	Traffic	355.258	355.242	361.220	357.297
	IoT sensor	2.502	2.488	7.069	2.488
	IoT packet	1.358	1.341	6.586	1.341
	MNIST	20.155	20.105	21.923	18.471
	FMNIST	19.864	19.533	21.598	18.538
CNN	Traffic	464.972	464.944	471.613	464.94375
	IoT sensor	2.502	2.488	12.119	3.606
	IoT packet	1.891	1.864	9.933	1.891
	MNIST	24.951	24.713	37.873	27.099
	FMNIST	24.938	24.394	40.254	26.696

for using the blockchain to protect data privacy and improve learning accuracy in urban computing, where multiple SPs generate numerous requests.

The number of features uploaded to the blockchain varies depending on the data characteristics; hence, the execution time varies depending on the dataset. A larger number of features in a dataset results in a greater number uploaded to the blockchain. Therefore, due to the significant data size, it takes more time to confirm transactions on the blockchain. The PoW is used when many people reach a consensus in a public environment. Many people use computing power to solve problems, and the system adjusts the mining fee and block generation time by adjusting the PoW difficulty. However, our proposal does not require a high difficulty because the UAVs and SPs are both authenticated and participate in the network. Therefore, if the system sets an appropriate difficulty or uses another consensus algorithm, we expect to minimize the overhead.

Table 10
Average latency by transaction processing and queries in the blockchain consensus algorithm.

Consensus algorithm	Average transaction processing latency (s)	Average transaction query latency (s)
PBFT	2.69	0.002
PoW(Diff = 5)	3.59	0.003
PoW(Diff = 10)	12.41	0.003

5. Discussion and analysis

The heightened apprehension among users regarding the protection of their data privacy has led to research into security in various systems. The proposed system ensures security in urban computing while protecting privacy and providing users with appropriate information. We expect our proposed system can be broadly applied in high data utilization AI fields. In particular, it can establish a secure environment when applied to medical and logistics systems that utilize sensitive data.

This section discusses the heterogeneous IoT, scalability problems in blockchain, and the different user privacy requirements in DP. Additionally, it reviews how the system can be improved to address these challenges. Then, we compared our proposal with related work and analyzed security.

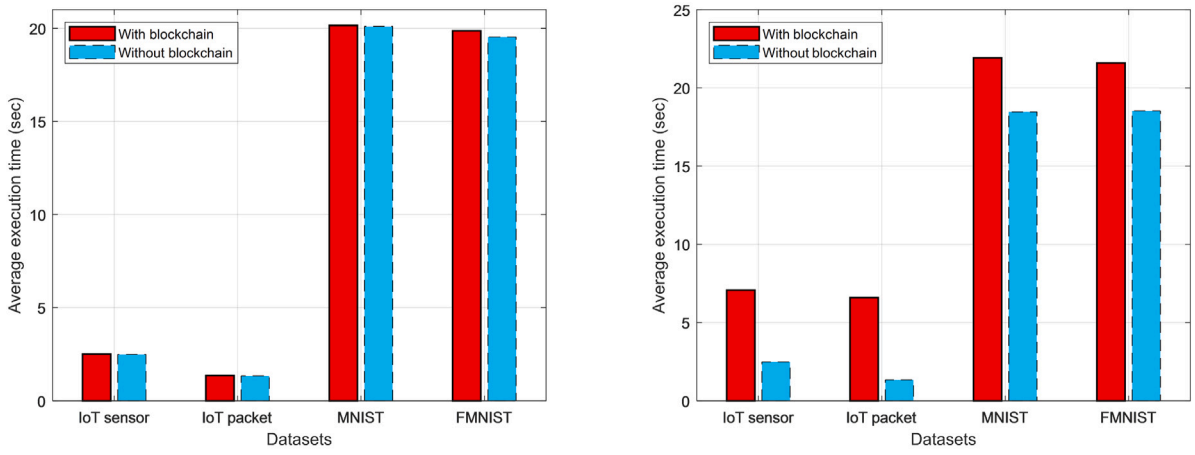
5.1. Discussion and system refinement

5.1.1. Heterogeneous IoT in urban computing

Most IoT systems assume that the capacities for computation, communication, and storage are uniform across devices. However, urban computing consists of diverse types of IoT devices, resulting in diverse storage and communication capabilities for each node. Therefore, considering the diversity of these nodes is critical to building an appropriate communication environment. Each node in the proposed system is assumed to generate only data; hence, the system is expected to be implementable even in a heterogeneous IoT environment. Additionally, if the system is improved in a heterogeneous IoT environment, IoT devices with various capabilities can be distinguished and assigned roles. Among IoT devices, those with higher computation, storage, and communication capacities can be organized as leaders within the IoT device layer. These leaders can manage communication with a higher-level BS by assuming the role of a gateway. Through this, IoT devices with lower performance, such as sensor nodes, can focus solely on sensing tasks, while offloading communication functions to higher-capacity devices, establishing a more stable system.

5.1.2. Scalability of the blockchain

The blockchain faces scalability problems as all nodes possess the same ledger, requiring a resolution for the limited node storage capacity. Therefore, managing the storage overhead of nodes in the blockchain is essential. The proposed system does not upload the learning model or raw data to the blockchain; instead, it uploads only metadata such as mechanism parameters and IDs. Therefore, the storage overhead is minimal. Additionally, as the number of nodes in a blockchain increases, the time required for consensus increases, resulting in additional latency. However, in the proposed system, a



(a) Difficulty=1. (b) Difficulty=10.

Fig. 15. Average execution time by dataset using deep neural network (DNN).

fixed number of static infrastructures, not multiple IoT devices, own the blockchain ledger. Therefore, our system is expected to incur no additional latency as the node count increases. Moreover, parallelism can be considered a technique to increase the transaction processing system to solve the blockchain scalability problem. In the proposed system, transactions can be processed in parallel based on the district. For example, while UAV 1 processes a transaction for the request of SP 1 in District 1, UAV 2 can concurrently manage another transaction for SP 2 occurring in District 2. These two transactions do not conflict; hence, transactions can be processed simultaneously. Therefore, by incorporating parallelism, the blockchain scalability problems can be reduced, establishing an optimized blockchain environment.

5.1.3. Different user privacy protection requirements in DP

In DP, there is a limitation in accurately determining how much information is leaked depending on the data. Additionally, the privacy protection requirements may vary for each system user. For example, some users may consent to providing sensitive information, whereas others may be highly restrictive about the use of sensitive data. The distribution for adding DP noise can be selected differently for each data item in the proposed system to improve these requirements. This requires some modifications to our system. We add the algorithm list (exponential function, Laplace, Gaussian, etc.) used to add noise into the transaction list in the BS_chain. Then, when a new request occurs, we can check the noise addition algorithm, epsilon, data type, and other factors, using a blockchain query. Noise reuse can only be performed if these parameters are the same. When selecting a noise addition algorithm, if the noise cannot be added directly depending on the data distribution, DP can use an exponential-based algorithm. In contrast, if noise can be added directly, the Gaussian or Laplace distribution can be selected by examining the data sensitivity [52]. This will be considered in future work.

5.2. Comparison with related work

As the utilization of big data collected through the IoT has become increasingly prevalent, there has been a significant increase in research efforts to address problems related to data security. In a related work [24], a trusted execution environment was used to build a safe execution environment. This work assumes that there is a full node with software guard extensions installed, which requires a node with additional hardware installed, and the cost of building the system increases. In contrast, our system processes data in a trusted static infrastructure and returns only the results to the SP, which is an untrusted entity. This process allows our system to satisfy data security without installing additional hardware.

Another work [23] proposed a system using the blockchain and federated learning for a secure data-sharing mechanism. In this work, each team shares the training results from local data by uploading them to the blockchain. Data privacy can be protected because training results are uploaded without uploading data. However, all shared data, including such training results, are recorded on the blockchain; thus, all nodes must have a significant number of data, leading to high storage overhead. Additionally, as the data requester, who is not a trusted entity, can verify this entire process, there may be security risks from an attack perspective. In contrast, in the proposed system, using a child_chain ensures that the SP, who is a semi-trusted entity, cannot access the training process or parameters. The SP only obtains the results, and these results are generated as privacy-protected data based on the reliability of the SP. Furthermore, only the final model parameters and metadata generated from the results are stored on the blockchain, resulting in low storage overhead for blockchain nodes. Therefore, the system avoids causing blockchain scalability problems, allowing for the establishment of a stable system.

Another work [14] proposed an algorithm that completely or partially reuses noise depending on the query type using blockchain. This approach can build a safe system by minimizing the privacy cost, but in this work, it was assumed that the client requesting the data was safe, and data with added noise were returned to the client. If the client is malicious, the client can continuously receive noise-added data from the system. The client generates the standard deviation for adding noise into the system; thus, the personal information protection rate may be degraded if a malicious client manipulates this and sends various requests. In contrast, in the proposed system, the SP sending the request is considered a semi-trusted entity, and data that can infer sensitive information are not disclosed. Furthermore, by verifying the authentication of the SP and checking previous records, the system can ensure a secure environment for accepting requests.

5.3. Security analysis

In urban computing, ensuring data security is crucial due to the utilization of numerous data. The proposal ensures data security through SP authentication and guarantees data privacy using DP. Additionally, we analyze the security aspects from an information security perspective. Recent advances in information and communications technology have led IoT systems to satisfy confidentiality, integrity, and availability but also non-repudiation, accountability, and authenticity [53].

Confidentiality: The system should provide information only to authorized users. In our system, both the UAV generating information using the data and the SP using the information are authenticated entities. Additionally, by configuring the ledger differently through the child_chain, the semi-trusted entity, SP, can only confirm responses to requests. Therefore, the proposed system ensures confidentiality.

Integrity: The system must ensure that information has not been altered. The system stores critical data, such as requests, responses, and DP parameters, on the blockchain. The SP has only child chains; hence, it cannot access sensitive information, such as model parameters, making data tampering impossible. Therefore, the proposed system ensures integrity.

Availability: The system must provide resources when legitimate users require them. In our system, we used UAVs to process SP requests in real time. Additionally, UAVs can serve as secondary gateways in areas with unstable communication. Therefore, the proposed system ensures availability.

Non-repudiation: The system must ensure the prevention of the denial of responsibility for actual actions. Our system records the information of all entities that use data and the data content on the blockchain. Therefore, the proposed system ensures non-repudiation.

Accountability: The system should be able to track user behavior. In the system, the ID of the SP using data, the requested content, the ID of the BS that processed it, and the ID of the UAV are all recorded in the blockchain. Moreover, all blockchains are stored in the cloud. Therefore, the proposed system ensures accountability.

Authenticity: The system must verify whether the user is the correct user and whether the data have been altered. The system checks whether the SP service type is the same as before using the blockchain. Therefore, if the SP service type is not identical, the system does not provide information even if an attack, such as identity theft, occurs. Therefore, the proposed system ensures authenticity.

6. Conclusion

In urban computing, where all devices communicate with one another through the internet and generate big data, newer and diverse services have been emerging for user convenience. Many SPs require considerable data to provide the best service to users, which could result in security breaches. When the system hands over the raw data of an IoT to the SPs so that they can use the data for learning, the personal information of the IoT owner could be leaked to the SPs. In addition, when the BS learns and returns only the results to the SPs, there may be a delay in the SP requesting real-time results because the primary purpose of the BS is communication. Likewise, if the IoT learns the models, proceeding with learning due to resource limitations can be challenging.

Our proposed system uses UAVs to address these concerns. In this system, the UAV offloads the task of the BSs. When a request occurs from the SP, the UAV receives data from the BS and performs the task. Further, the BS uses DP to ensure the data privacy of IoT devices. The privacy budget should be limited to avoid repeated questions and answers in the DP. We reuse noise using the blockchain to manage multiple requests while limiting the privacy budget. Additionally, we propose data-adaptive parameter utilization to increase the accuracy of the learning results. The simulations confirm that the proposal can lower the privacy costs while protecting privacy, and in machine learning, the accuracy of the results is guaranteed. Our proposal is expected to be applied not only to urban computing but also to vehicle networks such as the internet of vehicles for data security and privacy. This work performed DP using the Laplace distribution. In future work, we will consider other algorithms. By analyzing the characteristics of the Gaussian and exponential algorithms commonly used in DP, we could increase the data efficiency by selecting a more suitable algorithm for the data. Furthermore, we will improve this system by considering the data utility for the privacy parameters recorded in the blockchain.

CRediT authorship contribution statement

Gabin Heo: Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Conceptualization. **Inshil Doh:** Writing – review & editing, Visualization, Validation, Supervision, Investigation, Formal analysis.

Declaration of competing interest

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

Data availability

No data was used for the research described in the article.

Acknowledgments

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government, Ministry of Science and ICT (MSIT) (NRF-2023R1A2C1005712)

References

[1] Y. Zheng, Urban computing: enabling urban intelligence with big data, Front. Comput. Sci. 11 (1) (2017) 1–3.
[2] T.H. Silva, A.C. Viana, F. Benevenuto, L. Villas, J. Salles, A. Loureiro, D. Quercia, Urban computing leveraging location-based social network data: a survey, ACM Comput. Surv. 52 (1) (2019) 1–39.
[3] Y. Zheng, L. Capra, O. Wolfson, H. Yang, Urban computing: concepts, methodologies, and applications, ACM Trans. Intell. Syst. Technol. 5 (3) (2014) 1–55.
[4] I.A.T. Hashem, R.S.A. Usmani, M.S. Almutairi, A.O. Ibrahim, A. Zakari, F. Alotaibi, S.M. Alhashmi, H. Chiroma, Urban computing for sustainable smart cities: Recent advances, taxonomy, and open research challenges, Sustainability 15 (5) (2023) 3916.
[5] A.D. Kounoudes, G.M. Kapitsaki, A mapping of IoT user-centric privacy preserving approaches to the GDPR, Internet Things 11 (2020) 100179.
[6] C. Li, B. Palanisamy, Privacy in internet of things: From principles to technologies, IEEE Internet Things J. 6 (1) (2018) 488–505.
[7] C. Pathmabandu, J. Grundy, M.B. Chhetri, Z. Baig, Privacy for IoT: Informed consent management in Smart Buildings, Future Gener. Comput. Syst. 145 (2023) 367–383.
[8] W. Zhang, Z. Zhang, H.-C. Chao, Cooperative fog computing for dealing with big data in the internet of vehicles: Architecture and hierarchical resource management, IEEE Commun. Mag. 55 (12) (2017) 60–67.
[9] S. Zahra, W. Gong, H.A. Khattak, M.A. Shah, H. Song, Cross-domain security and interoperability in internet of things, IEEE Internet Things J. 9 (14) (2021) 11993–12000.
[10] W. Xue, Y. Shen, C. Luo, W. Xu, W. Hu, A. Seneviratne, A differential privacy-based classification system for edge computing in IoT, Comput. Commun. 182 (2022) 117–128.
[11] B.B. Sezer, H. Turkmen, U. Nuriyev, PPFchain: A novel framework privacy-preserving blockchain-based federated learning method for sensor networks, Internet Things 22 (2023) 100781.
[12] Y. Zhao, J. Chen, A survey on differential privacy for unstructured data content, ACM Comput. Surv. 54 (10s) (2022) 1–28.
[13] T. Zhu, P. Xiong, G. Li, W. Zhou, Correlated differential privacy: Hiding information in non-IID data set, IEEE Trans. Inf. Forensics Secur. 10 (2) (2014) 229–242.
[14] Y. Zhao, J. Zhao, J. Kang, Z. Zhang, D. Niyato, S. Shi, K.-Y. Lam, A blockchain-based approach for saving and tracking differential-privacy cost, IEEE Internet Things J. 8 (11) (2021) 8865–8882.
[15] M. Bouroche, I. Dusparic, Urban computing: The technological framework for smart cities, in: Handbook of Smart Cities, Springer, 2020, pp. 1–25.
[16] K. Su, J. Li, H. Fu, Smart city and the applications, in: 2011 International Conference on Electronics, Communications and Control, ICECC, IEEE, 2011, pp. 1028–1031.
[17] C. Yin, J. Xi, R. Sun, J. Wang, Location privacy protection based on differential privacy strategy for big data in industrial internet of things, IEEE Trans. Ind. Inform. 14 (8) (2017) 3628–3636.
[18] C. Wang, J. Zhao, Network approaches in blockchain-based systems: Applications, challenges, and future directions, Comput. Commun. 212 (2023) 141–150.

[19] A. Kumari, M.M. Patel, A. Shukla, S. Tanwar, N. Kumar, J.J. Rodrigues, ArMor: A data analytics scheme to identify malicious behaviors on blockchain-based smart grid system, in: GLOBECOM 2020-2020 IEEE Global Communications Conference, IEEE, 2020, pp. 1–6.
[20] S. Tanwar, Q. Bhatia, P. Patel, A. Kumari, P.K. Singh, W.-C. Hong, Machine learning adoption in blockchain-based smart applications: The challenges, and a way forward, IEEE Access 8 (2019) 474–488.
[21] M. Zhang, Z. Sun, H. Li, B. Niu, F. Li, Y. Xie, C. Zheng, A blockchain-based privacy-preserving framework for cross-social network photo sharing, in: IEEE INFOCOM 2022-IEEE Conference on Computer Communications Workshops, INFOCOM WKSHPS, IEEE, 2022, pp. 1–6.
[22] Y.I. Alzoubi, A. Al-Ahmad, H. Kahtan, Blockchain technology as a Fog computing security and privacy solution: An overview, Comput. Commun. 182 (2022) 129–152.
[23] Q. Miao, H. Lin, J. Hu, X. Wang, An intelligent and privacy-enhanced data sharing strategy for blockchain-empowered Internet of Things, Digit. Commun. Netw. 8 (5) (2022) 636–643.
[24] H. Xie, J. Zheng, T. He, S. Wei, C. Hu, TEBDS: A Trusted Execution Environment-and-Blockchain-supported IoT data sharing system, Future Gener. Comput. Syst. 140 (2023) 321–330.
[25] A. Kumari, R. Gupta, S. Tanwar, Amalgamation of blockchain and IoT for smart cities underlying 6G communication: A comprehensive review, Comput. Commun. 172 (2021) 102–118.
[26] S. Biswas, K. Sharif, F. Li, B. Nour, Y. Wang, A scalable blockchain framework for secure transactions in IoT, IEEE Internet Things J. 6 (3) (2018) 4650–4659.
[27] B.K. Mohanta, D. Jena, S.S. Panda, S. Sobhanayak, Blockchain technology: A survey on applications and security privacy challenges, Internet Things 8 (2019) 100107.
[28] S. Kim, Y. Kwon, S. Cho, A survey of scalability solutions on blockchain, in: 2018 International Conference on Information and Communication Technology Convergence, ICTC, IEEE, 2018, pp. 1204–1207.
[29] J. Garzik, Block size increase to 2MB, Bitcoin Improv. Propos. 102 (2015).
[30] M. Basile, G. Nardini, P. Perazzo, G. Dini, SegWit extension and improvement of the BlockSim bitcoin simulator, in: 2022 IEEE International Conference on Blockchain, Blockchain, IEEE, 2022, pp. 115–123.
[31] A. Singh, K. Click, R.M. Parizi, Q. Zhang, A. Dehghantanha, K.-K.R. Choo, Sidechain technologies in blockchain networks: An examination and state-of-the-art review, J. Netw. Comput. Appl. 149 (2020) 102471.
[32] R. Blum, T. Bocek, Superlight–A permissionless, light-client only blockchain with self-contained proofs and BLS signatures, in: 2019 IFIP/IEEE Symposium on Integrated Network and Service Management, IM, IEEE, 2019, pp. 36–41.
[33] C. Dwork, Differential privacy: A survey of results, in: Theory and Applications of Models of Computation: 5th International Conference, TAMC 2008, Xi'an, China, April 25-29, 2008. Proceedings 5, Springer, 2008, pp. 1–19.
[34] C. Dwork, F. McSherry, K. Nissim, A. Smith, Calibrating noise to sensitivity in private data analysis, in: Theory of Cryptography: Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006. Proceedings 3, Springer, 2006, pp. 265–284.
[35] Y. Qiao, Z. Liu, H. Lv, M. Li, Z. Huang, Z. Li, W. Liu, An effective data privacy protection algorithm based on differential privacy in edge computing, IEEE Access 7 (2019) 136203–136213.
[36] Y. Liu, B.D. Cruz, E. Tilevich, Trusted and privacy-preserving sensor data onloading, Comput. Commun. 206 (2023) 133–151.
[37] F.Z. Errounda, Y. Liu, Adaptive differential privacy in vertical federated learning for mobility forecasting, Future Gener. Comput. Syst. 149 (2023) 531–546.
[38] O.S. Oubbati, M. Atiquzzaman, P. Lorenz, M.H. Tareque, M.S. Hossain, Routing in flying ad hoc networks: Survey, constraints, and future challenge perspectives, IEEE Access 7 (2019) 81057–81105.
[39] D.S. Lakew, U. Sa'ad, N.-N. Dao, W. Na, S. Cho, Routing in flying ad hoc networks: A comprehensive survey, IEEE Commun. Surv. Tutor. 22 (2) (2020) 1071–1120.
[40] M. Song, C. Li, J. Ye, X. Gong, Y. Luo, Efficient consensus algorithm based on improved DPoS in UAV-assisted mobile edge computing, Comput. Commun. 207 (2023) 86–99.
[41] M. Mozaffari, W. Saad, M. Bennis, Y.-H. Nam, M. Debbah, A tutorial on UAVs for wireless networks: Applications, challenges, and open problems, IEEE Commun. Surv. Tutor. 21 (3) (2019) 2334–2360.
[42] M.M. Azari, G. Geraci, A. Garcia-Rodriguez, S. Pollin, UAV-to-UAV communications in cellular networks, IEEE Trans. Wireless Commun. 19 (9) (2020) 6130–6144.
[43] X. Jiang, M. Sheng, Z. Nan, X. Chengwen, L. Weidang, W. Xianbin, Green UAV communications for 6G: A survey, Chin. J. Aeronaut. 35 (9) (2022) 19–34.
[44] E.T. Michailidis, K. Maliatsos, D.N. Skoutas, D. Vouyioukas, C. Skianis, Secure UAV-aided mobile edge computing for IoT: A review, IEEE Access 10 (2022) 86353–86383.
[45] S.S. Albouq, A.A. Abi Sen, A. Namoun, N.M. Bahbouh, A.B. Alkhodre, A. Alshanqiti, A double obfuscation approach for protecting the privacy of IoT location based applications, IEEE Access 8 (2020) 129415–129431.
[46] N. Holohan, S. Braghin, P. Mac Aonghusa, K. Levacher, Diffprivlib: the IBM differential privacy library, 2019, arXiv preprint arXiv:1907.02444.
[47] S. Moosavi, M.H. Samavatian, S. Parthasarathy, R. Teodorescu, R. Ramnath, Accident risk prediction based on heterogeneous sparse data: New dataset and insights, in: Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, 2019, pp. 33–42.
[48] E. Tsukerman, Machine Learning for Cybersecurity Cookbook: Over 80 Recipes on How to Implement Machine Learning Algorithms for Building Security Systems Using Python, Packt Publishing Ltd, 2019.
[49] D. Anguita, A. Ghio, L. Oneto, X. Parra, J.L. Reyes-Ortiz, et al., A public domain dataset for human activity recognition using smartphones, in: Esann, Vol. 3, 2013, p. 3.
[50] S. Nakamoto, Bitcoin: A peer-to-peer electronic cash system, Decentralized Bus. Rev. (2008) 21260.
[51] T.O. Hodson, Root-mean-square error (RMSE) or mean absolute error (MAE): when to use them or not, Geosci. Model Dev. 15 (14) (2022) 5481–5487.
[52] B. Niu, Y. Chen, B. Wang, Z. Wang, F. Li, J. Cao, AdaPDP: Adaptive personalized differential privacy, in: IEEE INFOCOM 2021-IEEE Conference on Computer Communications, IEEE, 2021, pp. 1–10.
[53] R. Von Solms, J. Van Niekerk, From information security to cyber security, Comput. Secur. 38 (2013) 97–102.