Ad Hoc Networks 161 (2024) 103532

Contents lists available at ScienceDirect

Ad Hoc Networks

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

A privacy-preserving location data collection framework for intelligent systems in edge computing

Aiting Yao [a], Shantanu Pal [b,*], Xuejun Li [a], Zheng Zhang [a], Chengzu Dong [b], Frank Jiang [b], Xiao Liu [b]

[a] School of Computer Science and Technology, Anhui University, Hefei, China
[b] School of Information Technology, Deakin University, Melbourne, Victoria, Australia

ARTICLE INFO

Keywords:
Location data collection
Local differential privacy
Edge computing
UAV delivery systems
Blockchain

ABSTRACT

With the rise of smart city applications, the accessibility of users' location data by smart devices has increased significantly. However, this poses a privacy concern as attackers can deduce personal information from the raw location data. In this paper, we propose a framework to collect user location data while ensuring local differential privacy (LDP) in the last-mile delivery system of Unmanned Aerial Vehicles (UAVs) within an edge computing environment. Firstly, we obtain the user location distribution Quad-tree by employing a region partitioning method based on Quad-tree retrieval in the specified data collection area. Next, the user location matrix is retrieved from the obtained Quad-tree, and we perturb the user location data using an LDP perturbation scheme on the location matrix. Finally, the collected data is aggregated using blockchain to evaluate the utility of the dataset from various regions. Furthermore, to validate the effectiveness of our framework in a real-world scenario, we conduct extensive simulations using datasets from multiple cities with varying urban densities and mobility patterns. These simulations not only demonstrate the scalability of our approach but also showcase its adaptability to different urban environments and delivery demands. Finally, our research opens new avenues for future work, including the exploration of more sophisticated LDP mechanisms that can offer higher levels of privacy without significantly compromising the quality of service. Additionally, the integration of emerging technologies such as 5G and beyond in the edge computing environment could further enhance the efficiency and reliability of UAV-based delivery systems, while also offering new challenges and opportunities for privacy-preserving data collection and analysis.

1. Introduction

The integration of edge computing and Internet of Things (IoT), has significantly enhanced people's lives, offering many convenient services [1]. For example, use of intelligent systems, e.g., unmanned aerial vehicles (UAVs) to transport and deliver goods from one location to another. As shown in Fig. 1, in a typical edge-based UAV delivery system, users can place orders for products using their smart devices, and the UAV delivers the products to the specified location. The local data, including order details and UAV status, are processed in edge servers, which receive the data through gateways from user devices. The processed data is then aggregated in a cloud server, where further analysis and storage occur. In this paper, we present a framework aimed at investigating user location privacy protection in intelligent systems, specifically within edge computing environments, using UAV delivery systems as a case study [2–4]. This domain remains under-explored in the existing literature. Our focus is on addressing privacy protection issues in edge computing, where its decentralized structure and resource limitations present distinct challenges to safeguarding privacy. However, to further advance the capabilities of UAV delivery systems, edge devices (e.g., UAVs) often necessitate the collection of user-related data, encompassing basic information (e.g., name, gender, age, etc.) and usage data (e.g., location, trajectory, behavior, etc.) [5–7]. Service providers may also gather users' location or track data to deliver personalized services and quick queries, e.g., map applications and GPS navigation. While user location data provides immense benefits, it also exposes the risk of divulging sensitive personal information. An attack on the edge devices collecting user data can lead to the compromising of personal privacy [8–10]. For instance, attackers can infer the user's home address, social relationships, and lifestyle patterns. Therefore, safeguarding the privacy of collected user data becomes imperative [11–13].

[*] Corresponding author.
E-mail addresses: shantanu.pal@deakin.edu.au (S. Pal), xjli@ahu.edu.cn (X. Li).

https://doi.org/10.1016/j.adhoc.2024.103532
Received 21 January 2024; Received in revised form 30 April 2024; Accepted 1 May 2024
Available online 7 May 2024

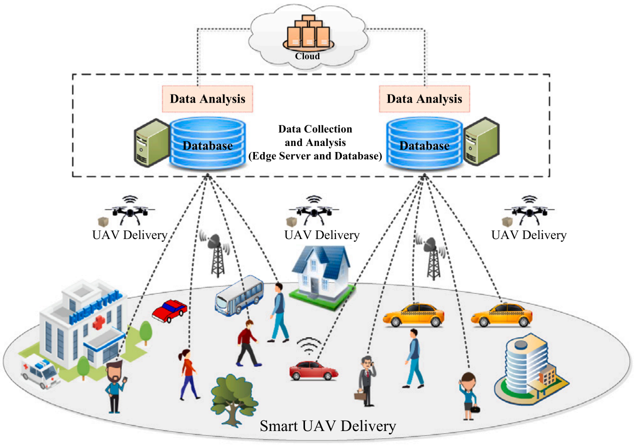


Fig. 1. Overview of an edge-based UAV delivery system.

In the context of swiftly evolving smart city technologies, the imperative for efficient and real-time data acquisition is paramount, especially for last-mile delivery systems utilizing UAVs [14–16]. Such systems are fundamentally dependent on user location data to enhance route optimization, minimize delivery duration, and boost overall operational efficacy. Nevertheless, this dependency on location data engenders substantial privacy concerns, owing to the escalating risk of malicious entities exploiting this data to deduce sensitive personal details, thereby posing threats to privacy [17,18].

Our research is driven by the critical necessity to mitigate the location data privacy issues while sustaining the efficiency and effectiveness of smart city applications. Prevailing strategies for data privacy often necessitate compromises, potentially degrading data quality or utility. Consequently, our objective is to formulate a framework that ensures robust privacy safeguards through Local Differential Privacy (LDP), simultaneously preserving the data's high utility for smart city applications.

In recent years, various works have focused on enhancing location privacy protection [6,19–24]. For example, anonymous identity protection and fuzzy mechanisms have emerged as well-established protection methods [25–27]. However, with the proliferation of big data scenarios, user location data originates from diverse sources, enabling attackers to link multiple data sources and conduct identity recognition attacks. Even if attackers have access to certain background information about a user, they are still capable of deducing the user's location data by integrating different datasets. [28]. To address this issue, an additional restriction is required to prevent attackers from learning extensive details about a user's personal data when they can access individual data responses. Differential Privacy (DP) is a privacy protection technology that ensures sensitive information of specific individuals is not easily identified in the data set by appropriate disturbance or noise processing of the data. In this context, we propose utilizing the LDP method [19,29] in our study. The user location data undergoes local randomization processing, making it challenging for attackers, irrespective of the user's background information, to infer the user's actual location from the perturbed single-user location data.

In this paper, we employ the Quad-tree index and LDP to consider privacy issues related to user location data in a UAV delivery system. Quad-tree is a spatial indexing structure that partitions data into different areas, enabling hierarchical management of data [30,31]. We propose a data collection framework that ensures the privacy of users' location data while still enabling efficient data analysis and utilization. On the one hand, Quad-tree index efficiently manages and retrieves user location data in a selected data collection area. It allows for spatial partitioning of the data, making it easier to organize and access location information, essential for processing large volumes of location data in an edge computing environment. On the other hand, LDP is employed to protect individual users' location data. It adds noise to the location data at the local level (i.e., edge level), ensuring that even if an attacker gains access to the data, they cannot accurately infer the actual location of individual users [32]. LDP provides a strong privacy guarantee by introducing randomization at the data source, making it an effective mechanism to prevent re-identification attacks and preserve users' privacy.

Quad-trees allow for more flexibility in layering geographic space, allowing for dynamic, adaptive subdivision of the space to better adapt to the density and features of different locations. In addition, when combined with localized differential privacy, random noise is introduced into each leaf node or subtree to ensure that differential privacy protection is added to fine-grained location data. This approach provides more granular privacy protection and reduces excessive perturbation of the overall data.

Furthermore, we assess the effectiveness of data collection by aggregating data analysis task completions. Data analysis tasks are present to the data aggregator, and we leverage *'blockchain'* as the data aggregator [33]. Blockchain offers several advantages, including data reliability, transparency, decentralization, distributed storage, smart contract functionalities, audit traceability, and data sharing and cooperation in a trustless platform [34,35]. These features enhance efficiency for data management, exchange, and utilization [36]. In the blockchain network, participating nodes validate and record transactions or blocks by solving complex mathematical problems. In our study, we employ the 'Proof of Work' (PoW) mechanism from blockchain to evaluate the utility of the collected dataset. Data analysts can determine the usability of the collected dataset by visualizing the data with a simple PoW. As a decentralized technology, blockchain can provide a trusted environment that eliminates the control of data by a central authority. Through the PoW mechanism, the generation and verification of data can be distributed across the network, increasing the transparency and verifiability of data. By employing cryptographic techniques and decentralized consensus mechanisms, blockchain safeguards sensitive information, mitigating the risks of unauthorized access by increased control over data. Simultaneously, the distributed ledger system streamlines data access, fostering collaboration while maintaining a secure environment that fortifies data privacy and amplifies the dataset's utility [37].

As previously mentioned, blockchain offers a secure and private environment for data storage by leveraging decentralization, immutability, and cryptographic security. Decentralization ensures that control is distributed among multiple nodes, reducing the risk of unauthorized access. Immutability guarantees that data remains unchanged by making any alterations virtually impossible without consensus. Cryptographic techniques are used to authenticate users and secure transactions, ensuring that only authorized users have access to the data. Smart contracts automate data governance, enforcing privacy policies transparently. These blockchain features together create a secure, transparent, and tamper-proof infrastructure, which enhances data privacy and maximizes the usefulness of various applications [38].

Incorporating Quad-tree index and LDP into the data collection framework in a blockchain platform, we aim to achieve a balance between data utility and protecting users' privacy for UAV delivery systems. The significant contributions of this paper lie in developing a practical framework that addresses privacy concerns in UAV-based data collection, enhancing the privacy and confidentiality of user location data in an intelligent edge computing setting. The major contributions of this paper can be summarized as follows:

- Based on the inspiration of a 'box plot' in statistics, a user distribution area division method based on the Quad-tree index is proposed in the area covered by one or more edge servers. The application of the box plot idea can better capture the characteristics of user location distribution and improve the deployment effect and resource utilization of the edge server.

- The user location distribution matrix is collected according to the user distribution area division of the Quad-tree index, and LDP is used locally for random disturbance protection of user location data.
- To better evaluate the utility of data collection, we have accumulated the amount of data analysis tasks completed by a novel blockchain-based data aggregator. A detailed proof of concept prototype is implemented to demonstrate the feasibility of the framework's performance.

The rest of this paper is organized as follows. Section 2 describes the related work. Section 3 presents the preliminary knowledge, e.g., the basics of LDP, the Quad-tree, and the data analysis tasks. In Section 4, we propose our framework for the UAV delivery system. Section 5 introduces a novel blockchain-based smart contract generation rule to measure the utility of collected datasets. Section 6 provides a detailed experimental setup and discuss the results. Finally, Section 7 concludes the paper with future work.

2. Related work

Several proposals address privacy issues in UAV delivery systems. For example, Kim et al. [39] discuss the challenges of privacy and efficient UAV movements. To respect citizens' privacy, a framework with differential permissions for UAV access is proposed. The authors formulate a problem to minimize UAV movements while preserving privacy using a novel priority-based approach (using UDiPP graphs). Enayati et al. [40] discuss protecting the location privacy of UAVs. The authors propose a privacy-preserving mechanisms (PPMs) for two major UAV applications: package delivery and IoT data collection. The PPMs randomize UAV trajectories or select random spots for data collection to confuse potential adversaries. The study analyzes the trade-offs between privacy guarantees and energy consumption. Additionally, a DP-based approach is developed. The study also show that the proposed methods effectively protect UAV location privacy without significant degradation in the performance of the applications. However, in contrast to our work, above mentioned papers do not incorporate the use of Quad-tree index for location data privacy. Moreover, they do not focus on edge-based systems and blockchain technology. Our work specifically addresses the integration of intelligent edge computing and blockchain to enhance privacy-preserving UAV location data collection.

A few other proposals discuss blockchain for UAV delivery. For example, Aljumah et al. [41] propose a blockchain-based method to ensure data safety and confidentiality in UAVs, and IoT devices. The proposal integrates an IoT application with a simulated vehicle monitoring system, using pentatope-based elliptic curve encryption and secure hash algorithm for anonymity. The cloud platform stores technical information and vehicular responses, while Ethbalance MetaMask wallet facilitates BCN-based transactions. Unlike our work, the authors use elliptic curve encryption for use's data protection. Lv et al. [42] present a blockchain-based privacy protection scheme for UAV big-data. The scheme uses a number theory cryptosystem for encryption and provides privacy analysis. The scheme demonstrates low computing cost and outperforms conventional approaches, offering insights for future UAV data privacy research. Unlike ours, the paper utilizes homomorphic encryption for UAV data privacy. Moreover, no implementation is discussed for UAV delivery context.

Liu et al. [43] introduce a game-theoretic routing framework for multiple cellular-connected UAVs engaged in goods delivery and sensing tasks. The framework optimizes trajectories and sensing task selections in a decentralized manner to minimize energy consumption, maximize sensing reward, and ensure user's privacy. The approach unifies routing and sensing as a single task selection process and employs a non-cooperative potential game to protect destination and trajectory privacy.

Xu et al. [44] propose a data collection system for UAV-assisted IoT applications using blockchain and edge computing to enhance security and energy efficiency. UAVs serve as edge data collection nodes, providing network access for IoT devices through regular cruises with recharging. Charging coins are earned as rewards for data forwarding and transactions, exchanged for charging time. The use of adaptive linear prediction reduces energy consumption by uploading prediction models instead of raw data.

Others similar papers, such as Wang et al. [45] proposed a new local differential privacy (LDP) framework called "L-SRR" to address privacy concerns in location-based service (LBS) applications due to the low utility of existing LDP mechanisms. They designed a randomization mechanism called Staircase Randomized Response (SRR) and extended empirical estimates to improve the utility of SRR in different LBS applications, such as traffic density estimation and k-nearest neighbor algorithms. Zhu et al. [46] proposed solution integrates LDP with conditional random field (CRF) to enable continuous location sharing while addressing security and privacy issues. The approach involves employing CRF to model users' mobility, establishing a system that combines location set and LDP for continuous location sharing, and evaluating the system's performance on actual datasets.

In Quad-tree and privacy-preserving, there are some research works. Zhang et al. [47] introduced a novel system for protecting personal location privacy in mobile crowdsourcing systems. The system can effectively balance the location privacy protection of crowdsourcing workers and the availability of location data, thus improving the efficiency and reliability of mobile crowdsourcing systems. Yao et al. [48] introduced a proposed solution to the problem of inadequate carrier privacy protection in intelligent logistics systems. The existing intelligent logistics system only protects the privacy of the cargo owner and the consignee, while the privacy of the carrier is not enough, which may lead to the disclosure of the carrier's privacy and directly or indirectly affect the logistics efficiency. They proposed two privacy protection algorithms, namely ε-LDP algorithm for carrier multidimensional numerical sensitive data and ε-LDP algorithm for carrier location-sensitive data. Ece Alptekin et al. [49] focused on building quadtrees of spatial data under the emerging concept of LDP. They took the data from the user using a single data collection step, propagated the density estimate to all nodes, and finally made structural corrections to the quadtree.

Unlike these proposals, we propose a framework to collect user location data while preserving privacy in UAV last-mile delivery systems that uses Quad-tree index and LDP for data perturbation and blockchain for data aggregation. By using the Quad-tree index and LDP for data perturbation, our proposal can protect user privacy while still collecting valuable UAV location data for analysis.

In summary, We have observed that existing works address user and location data privacy in different UAV delivery scenarios using various combinations of blockchain, edge computing, and encryption techniques. However, none uses real-time user location data for experiments, and there is a lack of focus on combining DP and blockchain for implementing a privacy protection framework for UAV delivery systems. Our work bridges these gaps by proposing a comprehensive privacy-preserving framework for UAV delivery systems while taking the advantages of blockchain-based smart contracts. To the best of our knowledge, our proposal is the first that integrates the benefits of blockchain and edge computing for UAV delivery location data privacy, utilizing LDP, and the Quad-tree index.

3. Preliminary knowledge

In this section, we provide an introduction to three key concepts: LDP, the Quad-tree, and data analysis tasks.

3.1. Local differential privacy (LDP)

In LDP, the protection of an individual's data is implemented on their personal device rather than relying on a central server or data aggregator. Before any data leaves the device, it undergoes a process of randomization where random noise is introduced. This step effectively obscures the precise details of the original data, ensuring that once the data has been uploaded to a server, it cannot be accurately traced back to the individual. As a result, the privacy of the user is preserved even after data collection and analysis by a remote server.

Def 1 (((ε, δ)-*Local Differential Privacy* [50]). For any different elements $v, v' \in D$, a randomized algorithm $\mathcal{M}$ guarantees (ε, δ)-LDP $(\varepsilon, \delta > 0)$ for any subset of the output $\mathcal{S} \subseteq Range(\mathcal{M})$, if satisfies:

$$Pr\left[\mathcal{M}(v) \in \mathcal{S}\right] \le e^{\varepsilon} \times Pr\left[\mathcal{M}(v') \in \mathcal{S}\right] + \delta \tag{1}$$

where, $Range(\mathcal{M})$ is the range of resultant output $\mathcal{M}$. ε, δ are the privacy parameter. The privacy parameter controls the privacy degree of the mechanism. If $\delta = 0$, $\mathcal{M}$ satisfies ε-LDP.

Randomized response mechanism is a privacy method used to survey and collect sensitive information, designed to encourage respondents to respond to real-time situations while protecting users' privacy.

Def 2 (*Randomized Response Mechanism: RR*). Let v be a user's binary value, $\hat{t}$ be the response, then for any user's value v,

$$Pr[\hat{t} = v] = \begin{cases} \dfrac{e^{\varepsilon}}{e^{\varepsilon}+1} & if \quad t = v \\ \dfrac{1}{e^{\varepsilon}+1} & if \quad t \neq v \end{cases} \tag{2}$$

The RR outputs either true or false values with probability $\frac{e^{\varepsilon}}{e^{\varepsilon}+1}$ and $\frac{1}{e^{\varepsilon}+1}$ respectively.

3.2. Quad-tree

The basic idea of Quad-tree retrieval is to search recursively in Quad-tree according to query conditions and space segmentation rules. By comparing the intersection relationship between the query conditions and the Quad-tree node boundary frame, we can determine the subtree that needs to be searched further. In this way, the search scope can be reduced under the efficiency of the tree, and the search speed can be improved.

Def 3 (*Quad-tree* [51]). A Quad-tree is a tree-like data structure in which each internal node has exactly four child nodes. It can be used to accurately represent the northwest, northeast, southwest, and southeast directions on a map.

Let Q be a Quad-tree, where each internal node is denoted as Q_i, and the children are represented as Q_i_1, Q_i_2, Q_i_3, and Q_i_4. The Quad-tree can divide the entire space into different blocks. In the Quad-tree, nodes are recursively divided into four equal areas via horizontal and vertical lines through the midpoint of each range.

3.3. Data analysis task

Let $u_i, (i = 1, 2, \ldots, n)$ denote users, a set of users is $U = \{u_1, u_2, \ldots, u_n\}$. For a divided area, the sub-area is recorded as $A_{l_h}, h = 1, 2, \ldots, \mathcal{H}$ after the first division. The notation is $A_{l_1, l_2, \ldots, l_{\mathcal{H}}}$, where l represents the number of layers of the Quad-tree split and h represents the depth of the Quad-tree split. In this paper, we focus on two data analysis tasks: *mean estimation* m_{l_h} and *frequency estimation* f_j. The corresponding Equations are shown in (3) and (4).

$$m_{l_h} = \frac{\sum_{l_h} n(1)_{l_h} - n(0)_{l_h}}{N} \tag{3}$$

where, l_h denotes the l_hth area, $n(1)_{l_h}$ and $n(0)_{l_h}$ are the count of 1 and 0 for area A_{l_h}. N denotes the total users number in area.

$$f_j = \frac{|\{count(v_{h,j}) | v_{h,j} = 1\}|}{N} \tag{4}$$

where, $v_{h,j}$ is users location matrix element V_{u_i}, N denotes the total users number in area (detailed discussion in Section 4.3).

4. Proposed framework

In this section, we discuss the proposed privacy-preserving data collection framework for UAV delivery systems.

4.1. Framework overview

As shown in Fig. 2, the framework is composed of three layers. From bottom to top, they are: *end device* layer, *edge computing* layer, and *cloud-blockchain* layer. The end device layer consists of devices, e.g., UAVs and other IoT devices that collect the raw data and communicate directly with the users. This design allows data collection to be closer to the data source, reducing transmission latency and improving the immediacy of data. Through the user's mobile device, such as a smartphone or tablet, real-time user location information can be provided using the built-in Global Positioning System (GPS) or other positioning technology. This allows the drone to accurately understand the user's current geographic location.

The edge computing layer comprises edge servers (that process and analyze data locally) that communicate directly with end devices and provide computing resources required for latency sensitive services and data privacy protection operations. This layer provides computing resources for latency-sensitive services while supporting data privacy protection operations. This helps reduce reliance on central cloud resources and speeds up data processing while protecting user privacy.

The cloud-blockchain layer, in our framework, ensures secure and tamper-proof data storage, supports decentralized data processing through smart contracts, and provides transparency and traceability of stored data. This layer enhances data integrity, reliability, and collaboration between the edge and cloud layers. In addition to this, it also provides transparency and traceability of data storage, and for non-real-time and offline services, it provides 'centralized' data storage and computing resources.

We consider a three-layer framework because these layers work collaboratively to distribute various components and functionalities, enhancing the efficiency of a UAV delivery system supported by intelligent edge computing. Further, the layered framework optimizes data processing, storage, and communication, improving performance and seamless delivery operations. Overall, the framework optimizes data processing, storage, and communication through distributed components and functions, thereby improving the efficiency and performance of drone delivery systems while ensuring data security and privacy protection.

The framework uses LDP technology to perturb the collected user location matrix to protect the privacy of user location. This approach provides a level of guarantee the unidentifiability of a single user's location by adding noise, thereby reducing the risk of privacy breaches. Another key point is the data aggregator with blockchain technology for the accumulation of data analysis tasks completed by each region. This helps to assess the effectiveness and usefulness of the collected data. In addition, aggregating data through blockchain can improve the efficiency and availability of data sharing in smart city environments with more transparency.

Similar data sharing and convergence needs exist in different types of smart city scenarios, including smart transportation, energy management, mobility management, etc. The privacy protection approach we discuss can easily be extended to these areas to protect sensitive information during data fusion. That said, our proposed approach is

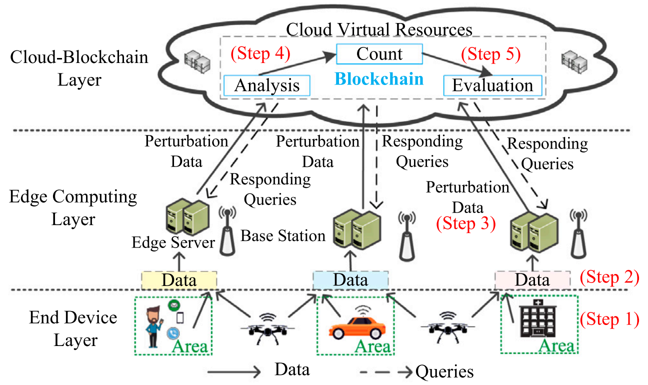


Fig. 2. A three-layer framework for data collection and analysis.

able to deal with the scale of the data with high integrity. Our privacy protection method in various smart scenarios can provide consistent and scalable privacy solutions, ensuring that the privacy rights of users are protected using the hierarchical structure of the smart city approach.

4.2. System functionality

The framework considers the collection of user location data in the area covered by multiple edge servers. Since the user's original location is based on the longitude and latitude coordinates of the geographic location on the earth's surface, the edge servers in any area collect the exact location data for the user location data (in the same data collection area). Note that even if an edge server is attacked, other edge servers in the area will collect user location data as usual. In addition, user location data is collected in the form of a location matrix instead of directly collecting user location data, e.g., longitude and latitude, geographic coordinates, etc., which can protect the privacy of user location data to a certain extent.

There are four significant functions about this framework. Firstly, **multi-server collection**: It operates in an area covered by multiple edge servers, each collecting user location data. Secondly, **location data based on coordinates**: The user location is determined based on longitude and latitude coordinates, representing precise geographic locations on Earth. Thirdly, **resilience to server attacks**: The framework is designed so that if one edge server is attacked or compromised, other servers in the area continue to collect user location data without interruption. Fourthly, **privacy-focused data collection**: Instead of directly collecting specific details like longitude and latitude, the framework gathers location data in the form of a location matrix. This approach helps to protect the privacy of user location data. In essence, this framework provides a resilient, multi-server system for collecting user location data with an emphasis on privacy and security, using a matrix form of data collection rather than direct geographic coordinates.

The application scenarios in the edge computing environment may vary, leading to different types of collected data sets. Firstly, the data collection area covered by multiple edge servers determines the number of areas. Secondly, user data from various collection areas are gathered at the terminal device layer and consolidated at the edge server for encoding and local disturbance. Subsequently, the perturbed data is uploaded to the cloud-blockchain layer. The cumulative completion of data analysis tasks in the cloud-blockchain layer is used to assess the utility of the collected data sets. Finally, the query results of the data set are returned.

As shown in Fig. 2, the proposed three-layer data collection framework involves gathering various types of collated data (e.g., users' location data, the trajectory and position of mobile devices like cars, and fixed locations of institutions, e.g., companies) for analysis, counting, and evaluations. The steps (in *red* color) indicate the functionalities performed at each layer. The following steps are involved in collecting and evaluating the utility of the collected data in our framework: (Step 1) dividing the user location distribution area, (Step 2) expressing the user location distribution as a location matrix according to the earlier step, (Step 3) perturbing the user location matrix by LDP, (Step 4) accomplishing some data analysis tasks on the perturbed location matrix, e.g., mean estimation and frequency estimation, and (Step 5) accumulating the data analysis tasks completed and evaluating the utility of the collected data.

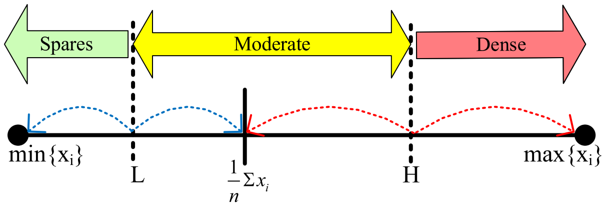


Fig. 3. SMD rule for Quad-tree index. The density of area less than L defined spare area, the density greater than H defined dense. In SMD rule, *green* express spare, *yellow* express moderate, and *red* express dense.

Next, a method based on the Quad-tree index is proposed to divide the user location distribution area as in Steps 1 and 2 (cf. Section 4.3). A detailed description of the LDP disturbed user location matrix is provided for Step 3 (cf. Section 4.4). For Steps 4 and 5, the advantages of blockchain are considered (cf. Section 5). The utility of the collected data set is evaluated by the number of tasks completed by cumulative data analysis and evaluation (cf. Section 6).

4.3. Proposed quad-tree index method

Next, we introduce our proposed Quad-tree index method as shown in Algorithm 1. In addition, with Quad-tree indexing, data with a specific spatial scope can be quickly located and queried. However, in the Quad-tree spatial index method, it is important to choose the appropriate partition conditions. Quad-tree partition conditions consider data density, space range, data distribution and target query operations. In practical application, according to data characteristics and query requirements, the most suitable partition condition is found by evaluating the impact of different partition conditions on index performance. Based on the 'box plot' idea in statistics, a Quad-tree area division rule (SMD rule: Spares-Moderate-Dense) is designed and illustrated in Fig. 3.

Before introducing the SMD rule, we introduce the conception of area density. Area density refers to the number of data points in a spatial area. In this paper, the area density denotes as: $D(A_{l_1,l_2,\ldots,l_i}),(i = 1,2,3,4)$. In the following, we will use $D(x_i)$ to represent $D(A_{l_1,l_2,\ldots,l_i})$. In Quad-tree partitioning, area density is one of the common considerations that can be used to determine whether further area partitioning is needed. In general, when the area density of a node exceeds a preset condition, a node can be further divided into four sub-nodes to achieve a more fine-grained space division. However, the selection of the condition is often difficult and has a significant impact on the partition result.

In statistics, a 'box plot' visually display the data set's center location, dispersion, outliers, and skew to show a data set's distribution characteristics and outliers. By observing the box plot, analysts can obtain information about data distribution, outliers and anomalies, etc., to help with data analysis and comparison. Firstly, we calculate the area density lower bound L and higher H to get the distribution degree of area users. Secondly, that determines the degree of dispersion of

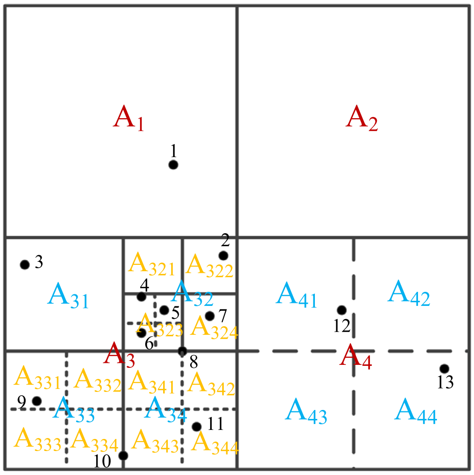


Fig. 4. The area partitioning is based on the Quad-tree index. The crossed solid lines indicate that the area is divided again according to the SMD rule. Dashed lines indicate that the area is divided once, and the division stops.

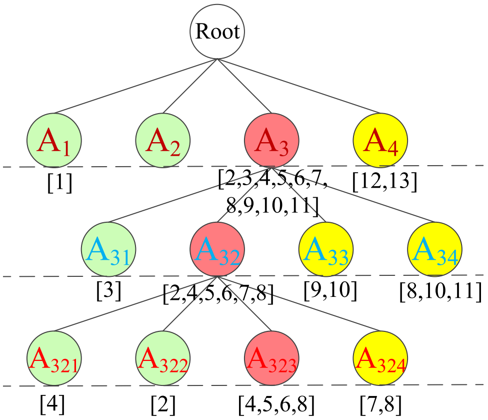


Fig. 5. The Quad-tree expression of area partitioning. The *green* express spares area, *yellow* express moderate area, and *red* express dense. The data points contained are listed below each layer of the Quad-tree.

the divided areas, e.g., spares, moderate and dense. Finally, Quad-tree determines whether the area needs to be divided according to the degree of dispersion of the area, as shown in Eq. (5).

$$P[x_i] = \begin{cases} -1, & if \quad min\{D(x_i)\} < D(x_i) < L \\ 0, & if \quad L \le D(x_i) \le H \\ 1, & if \quad H < D(x_i) < max\{D(x_i)\} \end{cases} \tag{5}$$

where, $P[x_i]$ represent the area partition degree. The area density lower bound L and higher bound H are calculated as Eqs. (6) and (7).

$$L = \frac{1}{2} \times (\overline{D(x_i)} + min\{D(x_i)\}) \tag{6}$$

$$H = \frac{1}{2} \times (\overline{D(x_i)} + max\{D(x_i)\}) \tag{7}$$

where, $\overline{D(x_i)} = \frac{1}{4}\sum_{i=1}^{4} D(x_i)$. Then, the partition number function is defined as Eq. (8).

$$f(P[x_i]) = \begin{cases} 0, & if \quad P[x_i] = -1 \\ 1, & if \quad P[x_i] = 0 \\ SMD, & if \quad P[x_i] = 1 \end{cases} \tag{8}$$

As shown in Fig. 4, the area first divided in A_1, A_2, A_3, and A_4, then the area A_1 and A_2 are stop partitioning. After the area A_3 is divided, the SMD rule is used to determine the division. Area A_4 is divided once, and then the division is stopped.

The user's location can be described by a Quad-tree matrix according to the Quad-tree index method. In the user location Quad-tree matrix, the horizontal coordinate represents the number of index levels of the Quad-tree. The ordinate indicates the number of nodes corresponding to the user in a certain layer. A Quad-tree is shown in Fig. 5. Then, a user's location matrix is obtained by the Quad-tree. For example, the Quad-tree matrix of user u_1, u_2, u_3, and u_4 are shown as follows:

$$V_{u_1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, V_{u_2} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix},$$

$$V_{u_3} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, V_{u_4} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \end{bmatrix}.$$

A Quad-tree index is a tree-like data structure used to efficiently divide two-dimensional space into smaller areas for easy management and indexing of geographic data. This partitioning method can process and query a large amount of geographic location data efficiently, especially for the scenario that needs to locate and retrieve geographic information quickly. In order to achieve evidentiary representation of user location, first, the end device in the terminal device layer in the edge computing environment reads the data and performs preliminary partitioning. Then, each area density boundary is calculated. Then, a partition function is constructed based on the regional density boundary, and the result of the partition function is used to decide whether to further divide the sub-region. Repeat the above operations to gradually build a Quad-tree structure. Finally, according to the area division of the Quad-tree, a matrix representation of the user's location is obtained.

4.4. Data perturbing with LDP

Next, we discuss the process of data perturbing with LDP. The process concluded *encoding* and *perturbing*, with Step 3 in Section 4.2.

4.4.1. Encoding method

Algorithm 1: Build Quad-tree

```
Require: Spatial datasets for geographical referencing $geoData$, the
    number of partitions $repeat$.
Ensure: A Quad-tree based geographic position $qtGeo$
if $repeat = 0$ then
    return new node(geoData,area)
if $repeat = 1$ then
    while $i$ in $[nw, ne, sw, se]$ do
        $LocationArea[i] \leftarrow geoData$
        return $BuildQuadTree(LocationArea[i], 0)$
else if $repeat = 2$ then
    while $i$ in $[nw, ne, sw, se]$ do
        $LocationArea[i] \leftarrow geoData$
        $L = \frac{1}{2} \times (avg \sum_i LocationArea[i] + min\{LocationArea[i]\})$
            $(i \in [nw, ne, sw, se])$
        $H = \frac{1}{2} \times (avg \sum_i LocationArea[i] + max\{LocationArea[i]\})$
            $(i \in [nw, ne, sw, se])$
    while $i$ in $[nw, ne, sw, se]$ do
        $repeat[i] \leftarrow check(LocationArea[i], L, H)$
        return $BuildQuadTree(LocationArea[i], repeat[i])$
```

The users' location need to encoding by the Quad-tree index. The Algorithm 1 to build a Quad-tree based on geolocation data is described as follows: Firstly, the algorithm read the coordinate dataset of the geographic location and then divide it into four sub-areas based on the locations in the dataset, which are (northwest) nw, (northeast) *ne*, (southwest) sw, and (southeast) se. Secondly, the area density lower bound L and higher bound H in each area are calculated by Eqs. (6) and (7). Thirdly, the partition number function is calculated by Eq. (8). If $f(P[x_i]) = SMD$, then the sub-area will further continue to be divided. If $f(P[x_i]) = 1$, then the subarea will be divided again subsequently. If $f(P[x_i]) = 0$, then no further division operation will be performed. Then, we obtain the Quad-tree structure. Finally, the users' location matrix are obtained by the Quad-tree of the area.

4.4.2. Perturbing method

The users' location matrix need to perturbed via LDP, which uses the Randomized Response (RR) mechanism. After obtaining the users' location matrix, iteratively read the position information stored in each leaf node and record the path information of each leaf node. The leaf node of each position using the unique thermal encoding and LDP mechanism. Then, according to the perturbation result, aggregate and thus correct the error caused by noise. Finally, new random coordinates are generated based on the new perturbation results. Since only the leaf node position is perturbed, the coordinates are generated only in the area covered by the parent node. Which significantly reduces the additional bias caused by the position perturbation. Moreover, we can efficiently count the number of positions in each area.

LDP technique ensures that the privacy of individual user data is maintained by adding noise to the data. The RR mechanism, a part of LDP, provides a way to collect statistical data about a population while maintaining individual privacy. In this context, it is used to obfuscate the exact positions in the location matrix, making it difficult to trace back to the individual user's exact location. The unique thermal encoding mentioned likely refers to a specific method of encoding the data before applying LDP, perhaps intended to optimize the perturbation process or to maintain data integrity after noise is added. Furthermore, the method's focus on only perturbing leaf node positions and generating coordinates within parent nodes' areas shows a tailored approach to minimizing data distortion while ensuring privacy. This selective perturbation helps maintain the utility of the data for statistical analysis while reducing the risk of introducing significant errors or biases. In summary, this technique effectively balances the need for user privacy with the utility of the location data, ensuring that while individual locations are obscured to protect privacy, the overall statistical characteristics of the data remain useful for analysis.

5. Collection data utility

In this section, we discuss the method of evaluating the collection data utility with blockchain. The method concluded data analysis task (cf. Section 5.1) and data utility within blockchain (cf. Section 5.2), with Steps 4 and 5 discussed in Section 4.2.

5.1. Mean and frequency estimation

In our proposed method, the data analysis tasks are mean and frequency estimation. Algorithm 2 describes the proposed method for mean and frequency estimation.

Lemma 1. *For Algorithm 2, the variance of the perturbed value V^* in the worst case is:*

$$Var[V^*] = \frac{e^{\frac{\varepsilon}{2}}}{(e^{\frac{\varepsilon}{2}}+1)^2}$$

Proof. Since $V^* \in \{0, 1\}$, the expectation of V^* is computed as:

$$E[V^*] = 1 \times \frac{e^{\frac{\varepsilon}{2}}}{e^{\frac{\varepsilon}{2}}+1} + 0 \times \frac{1}{e^{\frac{\varepsilon}{2}}+1} = \frac{e^{\frac{\varepsilon}{2}}}{e^{\frac{\varepsilon}{2}}+1}$$

$$E[(V^*)^2] = 1^2 \times \frac{e^{\frac{\varepsilon}{2}}}{e^{\frac{\varepsilon}{2}}+1} + 0^2 \times \frac{1}{e^{\frac{\varepsilon}{2}}+1} = \frac{e^{\frac{\varepsilon}{2}}}{e^{\frac{\varepsilon}{2}}+1}$$

then,

$$Var[V^*] = E[(V^*)^2] - (E[V^*])^2 = \frac{e^{\frac{\varepsilon}{2}}}{e^{\frac{\varepsilon}{2}}+1} - (\frac{e^{\frac{\varepsilon}{2}}}{e^{\frac{\varepsilon}{2}}+1})^2 = \frac{e^{\frac{\varepsilon}{2}}}{(e^{\frac{\varepsilon}{2}}+1)^2} \quad \square$$

Theorem 1. *The Algorithm 2 satisfies $\frac{\varepsilon}{2}$-Local differential privacy.*

Proof. For user u_i, let $V^*_{u_i}$ and $V^*_{u_i}$ be the output of Algorithm 2.

$$Pr[V^*_{hj} = 1 | V_{hj} = 1] = \frac{e^{\frac{\varepsilon}{2}}}{e^{\frac{\varepsilon}{2}}+1}$$

$$Pr[V^*_{hj} = 1 | V_{hj} = 0] = \frac{1}{e^{\frac{\varepsilon}{2}}+1}$$

Then, for any neighbor matrix V_{u_i} and V'_{u_i},

$$\frac{Pr[V^*_{hj} = 1 | V_{u_i}]}{Pr[V^*_{hj} = 1 | V'_{u_i}]} = \frac{Pr[V^*_{hj} = 1 | V_{hj} = 1]}{Pr[V^*_{hj} = 1 | V_{hj} = 0]}$$

$$\leq \frac{e^{\frac{\varepsilon}{2}}}{e^{\frac{\varepsilon}{2}}+1} / \frac{1}{e^{\frac{\varepsilon}{2}}+1} = e^{\frac{\varepsilon}{2}}$$

Thus the Algorithm 2 satisfies $\frac{\varepsilon}{2}$-Local differential privacy. □

5.2. Employing blockchain for data utility

In our framework, we utilize blockchain as a transparent, trustless, and decentralized platform for managing and processing data [52]. We introduce five *smart contracts* operating on the blockchain to efficiently execute the activities of the framework. These contracts enable task issuance, execution, completion statistics, block generation, and a novel PoW consensus mechanism. A brief description of them is as follows:

- *Task Issuance Smart Contract*: It is responsible for issuing new data analysis tasks to the network. This smart contract gets activated each time a new data analysis task needs to be issued. The information stored may include the task type (e.g., mean estimation, frequency estimation, etc.), the expected completion time, the data source, and the rewards associated with completing the task. The contract ensures all necessary details are transparent and agreed upon by all parties involved.
- *Task Execution Smart Contract*: It oversees the progress of newly created tasks and handles any possible disputes or failures. This contract is invoked when a node (or multiple nodes) in the network decides to take on a data analysis task. The contract keeps track of the assigned tasks, their progress, and their outcomes. The contract might also handle re-assignments of tasks.
- *Task Completion Statistics Smart Contract*: Upon task completion, it keeps track of the details of the completed tasks and uses this data (i.e., task completion data) for performance evaluation. This might involve recording the number of tasks completed by each node, the time taken for each task, and the accuracy of the task outcomes. This data can be used to evaluate and rank the performance of the nodes in the network and reward them accordingly.

Algorithm 2: Mechanism for Data Analysis of Collection Data under ε-LDP

```
Require: Users' location matrix $V$, Quad-tree deep $h$, privacy budget $\varepsilon$.
Ensure: Perturbed matrix $V^*$, Mean vector $m^*$, Frequency vector $f^*$
Perturbed matrix $V = V^* \in \{0,1\}$ by sampling each $v_{h,j} (j = 1,2,3,4)$ in-dependently from the following distribution:
    $Pr[V^*_{h,j} = 1] = \begin{cases} \frac{e^{\frac{\varepsilon}{2}}}{e^{\frac{\varepsilon}{2}}+1} & if \quad V_{h,j} = 1 \\ \frac{1}{e^{\frac{\varepsilon}{2}}+1} & if \quad V_{h,j} = 0 \end{cases}$
for each $V^*_{h,j}$ do
    Aggregator counts 1 and 0 in matrix $V^*$:
        $n_1 = count(1)$
        $n_0 = count(0)$
    Total count: $N = n_1 + n_0$
    Aggregator calibrates the counts as:
        $n^*_1 = \frac{(p-1)N + n_1}{2p-1}$
        $n^*_0 = \frac{(p-1)N + n_0}{2p-1}$
    where
        $p = \frac{e^{\frac{\varepsilon}{2}}}{e^{\frac{\varepsilon}{2}}+1}$
    clip $n^*_1, n^*_0$ to $[0, N]$
    Aggregator calculates mean $m^* = \frac{n^*_1 - n^*_0}{N}$
    Aggregator calculates frequency $f^*$
    Aggregator calibrates the $f^*$ as:
        $f^* = \frac{p-1+f^*}{2p-1}$
    where
        $p = \frac{e^{\frac{\varepsilon}{2}}}{e^{\frac{\varepsilon}{2}}+1}$
return $V^*$, $m^*$, $f^*$.
```

- *Block Generation Smart Contract*: Once a data analysis task is completed, this contract packages the completed task results into a block and adds it to the blockchain. It ensures the task outcome and verification are properly documented. The block will contain the initial task details, the node(s) that performed the task, the task outcome, and the verification of the results. When a new block is generated, the contract also records the time of its creation; this timestamp is crucial for maintaining the chronological order of the blockchain and can also be used for performance evaluation and debugging purposes.
- *PoW Smart Contract*: It implements a PoW consensus mechanism, ensuring that any node adding a new block to the blockchain has performed a certain amount of work (cf. Algorithm 3). The system described implements a Proof of Work (PoW) consensus mechanism, which is pivotal in a blockchain network to ensure data integrity and security. In this context, any node that wishes to add a new block to the blockchain must prove that it has expended a certain amount of computational work, aligning with the specified PoW algorithm. This mechanism serves not just to secure the network, but also plays a role in evaluating the quality of data collected across various geographic areas. The contract, or set of rules encoded within the blockchain, assesses data quality based on several key criteria: the size of the area from which the data is collected, the quantity of the data, and the distribution of users within that area. This evaluation recognizes that data quality can vary significantly depending on these factors. For example, smaller areas with high user density might yield higher-quality data due to more intensive user interactions and data points. The system further refines its assessment of data quality by considering the speed of task completion relative to the area size. Faster completion of tasks in a given area suggests higher quality and relevance of the data, as it indicates a higher level of user engagement and activity. This nuanced approach to data quality assessment ensures that the blockchain network not only securely stores data but also maintains a high standard of data utility and relevance.

Our proposed approach to PoW not only ensures the integrity of the blockchain network, but also significantly enhances the quality of the data being processed and analyzed within the framework. It aligns the incentives of the nodes with the overarching goal of high-quality data collection and analysis, fostering a more reliable and efficient blockchain-based data processing framework.

Algorithm 3: Proof of Work (PoW) Smart Contract

```
Require: $block, Quad-treeArea, data, usersID$.
Ensure: Boolean value indicating if the block passes the PoW contract
$work \leftarrow$ ProveWork($block$);
if $work$ = False then
    return False                                  ▷ Block failed proof of work
$areaSize \leftarrow$ CalculateAreaSize($quadtreeArea$);
$dataQuantity \leftarrow$ CountData($data$);
$userDistribution \leftarrow$ CalculateUserDistribution($users$);
$dataQuality \leftarrow$ EvaluateDataQuality ($dataQuantity, areaSize, userDistribution$)
if $dataQuality$ < threshold
    return False        ▷ Collected data did not meet quality standards
else
    for each $transaction$ in $block$ do
        if $transaction$ is invalid
            return False                  ▷ Invalid transaction found in block
        else
            $transactionData \leftarrow$ ExtractData($transaction$);
            if $transactionData$ is malicious
                return False              ▷ Malicious data found in transaction
            else
                if not CheckConsistency(transactionData, block)
                    return False          ▷ Inconsistent data found in block
Update $block$ status to verified
return True       ▷ Block passes proof of work, data quality and transaction checks
```

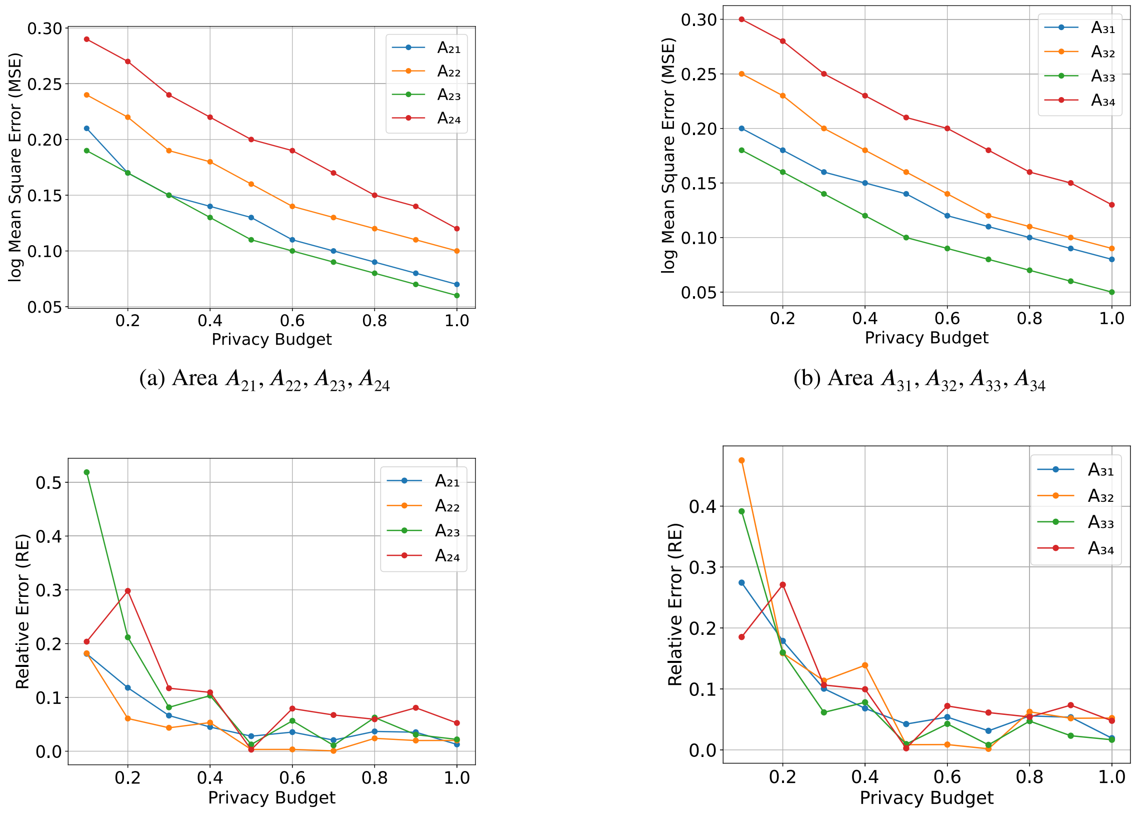
6. Experimental evaluation

In this section, we discuss the experimental setup and evaluate the archived results.

6.1. Experimental setup

We used Python 3.9 to implement the differential privacy algorithm based on the Quad-tree index. The implementation was done on a 3.61 GHz Intel Core i7-12700K CPU with 64 GB of RAM. The smart contracts are made using Solidity [53] version 0.8.0. Furthermore, we employed a private Ethereum blockchain for increased control over network participants and reduced transaction costs.

To measure the accuracy of the proposed method, we use three metrics: *Mean Square Error* (MSE) [54], *Relative Error* (RE) [55], and



(a) Area $A_{21}, A_{22}, A_{23}, A_{24}$

(b) Area $A_{31}, A_{32}, A_{33}, A_{34}$

(c) Area $A_{21}, A_{22}, A_{23}, A_{24}$

(d) Area $A_{31}, A_{32}, A_{33}, A_{34}$

Fig. 6. MSE (Figs (a) and (b)), and RE (Figs. (c) and (d)) for areas $A_{l_1,l_2,\ldots,1}$-$A_{l_1,l_2,\ldots,4}$ with DP budget $\epsilon \in (0,1]$. Figs. (a) and (b) illustrate the mean estimation accuracy of our method. Figs. (c) and (d) illustrate the frequency estimation results of our method (discussion in 6.2.2 and 6.2.3). We select the division of A_2 and A_3 as they have the highest number of users.

Blockchain Cumulative Task Load. MSE is used to evaluate mean estimation, RE is used for frequency estimation. The calculation Equations are shown as follows:

$$MSE = \frac{1}{h}\sum_{i=1}^{h}(m_i - m_i^*)^2 \tag{9}$$

where, h is Quad-tree deep, m_i and m_i^* are real and estimated means value.

$$RE = MF_{i\in\mathbf{H}}(\frac{|f_i - f_i^*|}{f_i}) \tag{10}$$

where, MF is a statistical function named Median Function, f_i and f_i^* are real and estimated frequency value. In general, because the value of MSE is small, the logarithm of MSE is used to show the experimental results, e.g., $Log(MSE)$.

6.2. Results and discussions

Now we discuss the achieved results in the following four subsections.

6.2.1. Users' location distribution comparison

In order to protect user privacy while anonymized location data can be collected and analyzed to optimize logistics route planning, delivery time estimation, etc., thereby improving service efficiency and user experience. This article uses EUA data sets[1] collected from real-world data sources to implement the user location data collection and privacy protection scheme proposed in this paper. The EUA dataset exclusively encompasses data from the Australia region. Within this dataset, the 'edge-servers' folder contains information pertaining to the locations of edge servers, while the "users" folder comprises datasets related to user locations. It is important to note that the utilization of the "users" folder in the context of UAV logistics scenarios is specifically for simulation experiments. These experiments aim to validate the feasibility and effectiveness of the proposed framework.

In the analysis of the experimental results Figs. 7 and 8, we observed a clear difference between the real location distribution of users and the location distribution after privacy protection. The real location distribution map shows that users are mainly concentrated in the diagonal area of the map, and the distribution is relatively dense. The user density in these areas is high, showing an obvious clustering trend. In contrast, the location map after privacy protection shows a more uniform distribution pattern. In order to protect user privacy, real locations are randomized to a certain extent so that users appear to be spread over a wider area. This processing reduces the clustering of users in a specific area, which effectively hides the actual range of activities and habits of users. Overall, privacy measures effectively change the distribution of users' locations, reducing the possibility of associating personal information with a specific location, but at the same time may have some impact on location-based services and data analysis.

6.2.2. Mean square error (MSE)

As illustrated in Figs. 6 (a) and (b), as the privacy budget increases, the MSE tends to decrease. This is because a higher privacy budget allows for more noise to be added during the data analysis process, providing stronger privacy guarantees and reducing the risk of overfitting or memorization of individual data points. As a result, the model may generalize better and produce more accurate predictions, leading to lower MSE values.

[1] https://github.com/swinedge/eua-dataset

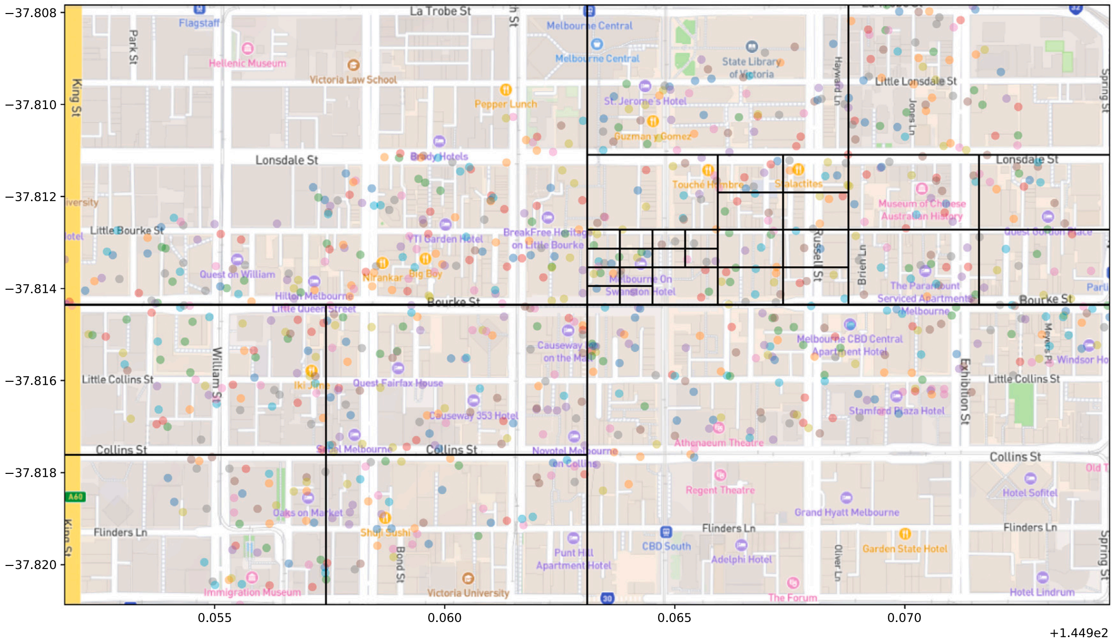


Fig. 7. Real user location distribution map.

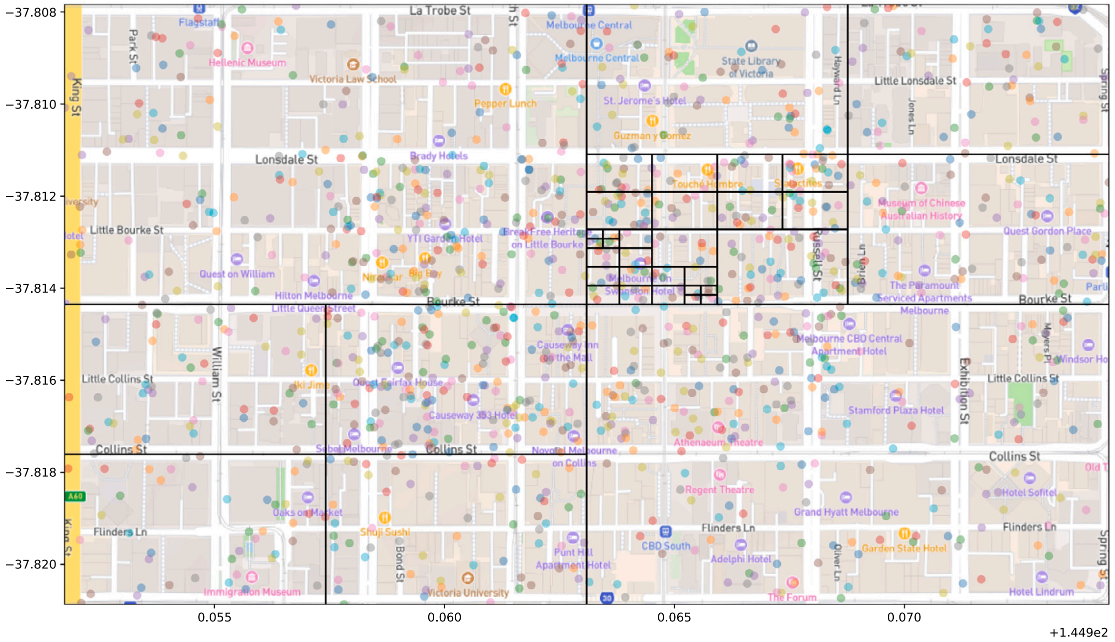


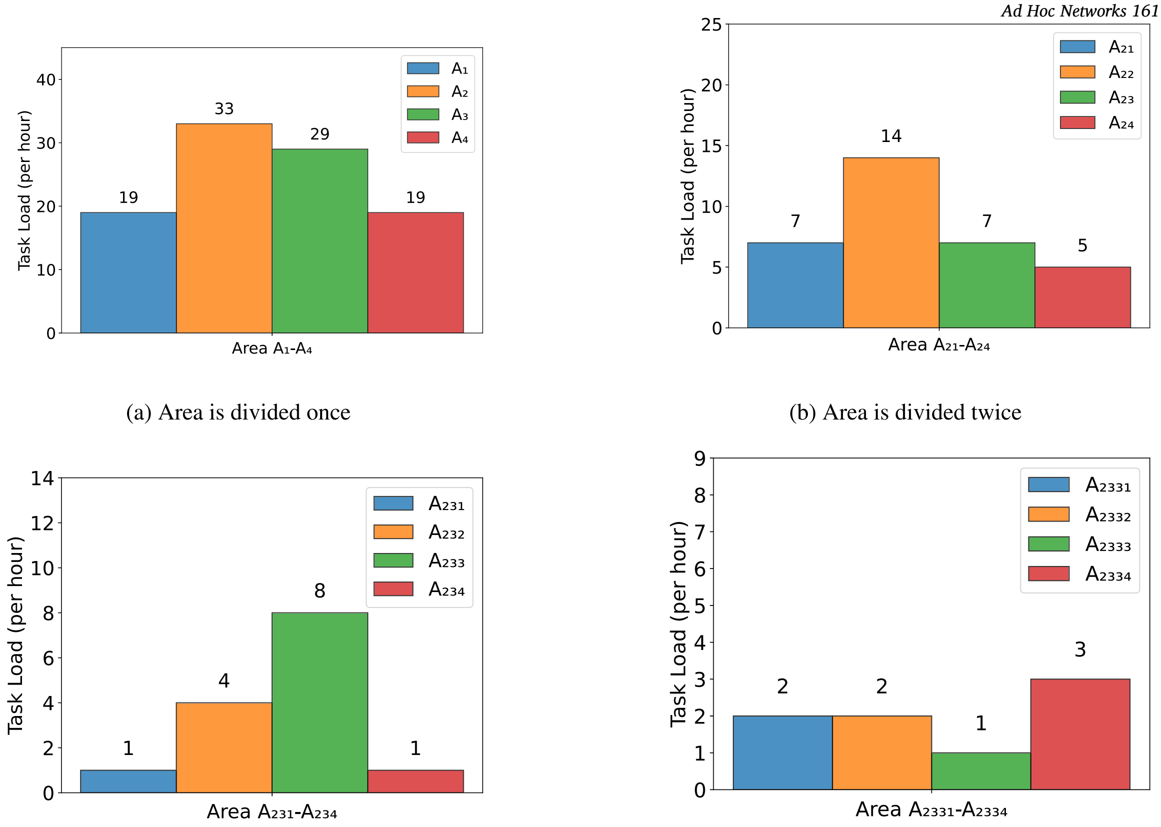
Fig. 8. Privacy-protected user location distribution map.

Conversely, when the privacy budget is low, the noise added to the data is limited, which may result in less privacy protection but potentially higher accuracy in the analyzed data. However, low privacy budgets may also make the model more susceptible to overfitting, leading to higher MSE values.

6.2.3. Relative error (RE)

As illustrated in Figs. 6 (c) and (d), there exists an inverse relationship between the privacy budget and the RE. The privacy budget is a parameter that controls the level of privacy protection and is typically denoted as ε. On the one hand, a smaller privacy budget indicates higher privacy protection, allowing for less privacy leakage. On the other hand, the relative error is used to measure the accuracy of data after privacy protection and represents the degree of difference between the estimated value and the true value.

When the privacy budget is small ($0 < \varepsilon$), the LDP mechanism adds more noise to protect data privacy, leading to a larger RE. In other words, as the privacy budget decreases, data accuracy may decrease due to the increased noise that may introduce further errors. Conversely, when the privacy budget is large ($\varepsilon \leq 1$), the LDP mechanism adds less noise, allowing for higher data accuracy and reducing the RE. In this case, estimation results closer to the true value may be generated, but at the same time, the level of privacy protection may weaken.



(a) Area is divided once

(b) Area is divided twice

(c) Area is divided thrice

(d) Area is divided forth

Fig. 9. A Blockchain cumulative task load to evaluate the collection data utility (discussion in 6.2.4) of Areas: (a) A_1, A_2, A_3, A_4, (b) division of A_2 (highest number of task load 33), (c) division of A_{22} (highest number of task load 14), and (d) division of A_{233} (highest number of task load 8). The X axis shows *Area*, and the Y axis shows *Task Load (Per Hour)*.

6.2.4. Blockchain cumulative task load

The blockchain cumulative task load is essential in evaluating the utility of data collected across different areas, particularly in high-density urban environments like a central business district. By leveraging blockchain technology, this approach ensures the integrity and reliability of data, as blockchain's decentralized nature makes the data secure and transparent. The cumulative task load represents the total number of data analysis tasks completed, offering insights into the efficiency of data collection and its effectiveness in various contexts. By analyzing data at different granular levels, from broader areas to specific segments, this method not only assesses the volume of data generated but also correlates the data utility with the density of users in those areas.

Blockchain cumulative task load is the cumulative number of data analysis tasks completed. The task load is used to evaluate the utility of the collected data for different areas (recall Fig. 4 for the division of data collection areas). In Fig. 9, we illustrate the comparison of task load per hour for different areas (i.e., $A_{l_1,l_2,\ldots,1}$, $A_{l_1,l_2,\ldots,2}$, $A_{l_1,l_2,\ldots,3}$, and $A_{l_1,l_2,\ldots,4}$). We use real-time location data within a central business district (CBD) of a metropolitan city for our experiment . All areas (i.e., A_1, A_2, A_3, A_4 of Fig. 9(a)) are with the same size (in square kilometre) and only difference is the number of task load per hour. That is, Fig. 9(a) illustrates the blockchain cumulative task load of Areas A_1 to A_4. Next, we examine data utility with more granularity of a particular area, for example, Area A_2, as it has the highest task load completion. As such, Fig. 9(b) illustrates the blockchain cumulative task load of Area A_{21} to A_{24}, i.e., the highest task load of Area A_2 (where A_2 = $(A_{21} + A_{22} + A_{23} + A_{24})$). Similarly, Fig. 9(c) illustrates the blockchain cumulative task load of Area A_{231} to A_{234}, which is the highest task load of Area A_{22}. Finally, Fig. 9(d) illustrates the blockchain cumulative task load of Area A_{2331} to A_{2334}, which is the highest task load of Area A_{233}. Fundamentally, with the high density (with the number of users), our framework performed significantly well in terms of data utility. In other words, our framework can provide better data utility with an increased number of users.

This approach demonstrates a systematic method for evaluating blockchain's effectiveness in handling data analysis tasks, particularly in densely populated urban areas. The focus on areas with varying task loads allows for a nuanced understanding of how blockchain technology can be optimized for data utility in different settings.

7. Conclusion and future work

In this paper, we have presented a framework for collecting user location data in intelligent systems. For example, we used a use-case scenario of a UAV delivery system. We defined a Quad-tree partition rule (SMD rule) to adapt to the density of users in different areas, resulting in a matrix of user locations. LDP is then utilized to perturb the collected user location matrix and protect the privacy of user locations. In our proposed framework, blockchain served as a data aggregator to accumulate completed data analysis tasks for each area, enabling evaluation of the utility of the collected data. However, preventing the tampering of private data remains an open issue, which we leave for future work. In addition, future studies on data aggregation and analysis methods are needed to enhance the efficiency and availability of data sharing in smart city environments. Striking a balance between transparency and privacy in blockchain design is crucial. Implementing advanced cryptographic techniques, such as zero-knowledge proofs or privacy-focused consensus mechanisms like zk-SNARKs, can mitigate these concerns. These methods enable transaction verification without revealing sensitive details. As blockchain evolves, optimizing protocols for privacy while managing computational costs will be essential to ensure a secure and efficient decentralized system. We leave this for another future work.

CRediT authorship contribution statement

Aiting Yao: Writing – original draft, Validation, Methodology, Data curation, Conceptualization. **Shantanu Pal:** Writing – review & editing, Visualization, Validation, Supervision, Resources, Methodology, Investigation, Conceptualization. **Xuejun Li:** Writing – review & editing, Supervision, Project administration, Methodology, Investigation, Formal analysis, Conceptualization. **Zheng Zhang:** Writing – review & editing, Software, Resources, Investigation, Data curation. **Chengzu Dong:** Writing – review & editing, Visualization, Software, Resources, Methodology, Conceptualization. **Frank Jiang:** Writing – review & editing, Supervision, Resources, Project administration, Conceptualization. **Xiao Liu:** Writing – review & editing, Supervision, Project administration, Methodology, Investigation, Conceptualization.

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.

Acknowledgement

This work was supported by the National Natural Science Foundation of China Project (No. 62372004).

References

[1] P. McEnroe, S. Wang, M. Liyanage, A survey on the convergence of edge computing and AI for UAVs: Opportunities and challenges, IEEE Internet Things J. 9 (17) (2022) 15435–15459.

[2] H. Li, S. Zhu, A. Tolba, Z. Liu, W. Wen, A reliable delivery logistics system based on the collaboration of UAVs and vehicles, Sustainability 15 (17) (2023) 12720.

[3] C. Dong, J. Zhou, Q. An, F. Jiang, S. Chen, L. Pan, X. Liu, Optimizing performance in federated person re-identification through benchmark evaluation for blockchain-integrated smart UAV delivery systems, Drones 7 (7) (2023) 413.

[4] F. Betti Sorbelli, et al., UAV-based delivery systems: a systematic review, current trends, and research challenges, J. Auton. Transp. Syst. (2024).

[5] X. Wang, J. Ma, X. Liu, R.H. Deng, Y. Miao, D. Zhu, Z. Ma, Search me in the dark: Privacy-preserving boolean range query over encrypted spatial data, in: IEEE INFOCOM 2020-IEEE Conference on Computer Communications, IEEE, 2020, pp. 2253–2262.

[6] H. Hu, K. Xiong, G. Qu, Q. Ni, P. Fan, K. Letaief, AoI-minimal trajectory planning and data collection in UAV-assisted wireless powered IoT networks, IEEE Internet Things J. 8 (2) (2020) 1211–1223.

[7] T. Sheng, R. Jin, C. Yang, K. Qiu, M. Wang, J. Shi, J. Zhang, Y. Gao, Q. Wu, X. Zhou, et al., Unmanned aerial vehicle mediated drug delivery for first aid, Adv. Mater. 35 (10) (2023) 2208648.

[8] S. Shen, K. Zhang, Y. Zhou, S. Ci, Security in edge-assisted Internet of Things: challenges and solutions, Sci. China Inf. Sci. 63 (2020) 1–14.

[9] Y. Liu, R. Zhao, J. Kang, A. Yassine, D. Niyato, J. Peng, Towards communication-efficient and attack-resistant federated edge learning for industrial Internet of Things, ACM Trans. Int. Technol. (TOIT) 22 (3) (2021) 1–22.

[10] Y. Liu, Z. Na, Y. Zhang, X. Qin, B. Lin, Multi-UAV-assisted covert communications for secure content delivery in Internet of Things, Comput. Commun. 210 (2023) 138–146.

[11] J. Geiping, H. Bauermeister, H. Dröge, M. Moeller, Inverting gradients-how easy is it to break privacy in federated learning? Adv. Neural Inf. Process. Syst. 33 (2020) 16937–16947.

[12] K. Gai, Y. Wu, L. Zhu, M. Qiu, M. Shen, Privacy-preserving energy trading using consortium blockchain in smart grid, IEEE Trans. Ind. Inform. 15 (6) (2019) 3548–3558.

[13] S. Pal, M. Hitchens, V. Varadharajan, Towards the design of a trust management framework for the Internet of Things, in: 2019 13th International Conference on Sensing Technology, ICST, IEEE, 2019, pp. 1–7.

[14] A. Yao, F. Jiang, X. Li, C. Dong, J. Xu, Y. Xu, G. Li, X. Liu, A novel security framework for edge computing based uav delivery system, in: 2021 IEEE 20th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), IEEE, 2021, pp. 1031–1038.

[15] Z. Lu, N. Yu, X. Wang, Incentive mechanism and path planning for Unmanned Aerial Vehicle (UAV) hitching over traffic networks, Future Gener. Comput. Syst. 145 (2023) 521–535.

[16] C. Zhang, X. Liu, M. Xiang, A. Yao, X. Fan, G. Li, Fed4ReID: Federated learning with data augmentation for person re-identification service in edge computing, in: 2023 IEEE International Conference on Web Services, ICWS, IEEE, 2023, pp. 64–70.

[17] S. Pal, M. Hitchens, V. Varadharajan, Access control for Internet of Things—Enabled assistive technologies: An architecture, challenges and requirements, in: Assistive Technology for the Elderly, Elsevier, 2020, pp. 1–43.

[18] S. Pal, M. Hitchens, V. Varadharajan, Towards a secure access control architecture for the Internet of Things, in: 2017 IEEE 42nd Conference on Local Computer Networks, LCN, IEEE, 2017, pp. 219–222.

[19] H. Wang, H. Hong, L. Xiong, Z. Qin, Y. Hong, L-srr: Local differential privacy for location-based services with staircase randomized response, in: Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security, 2022, pp. 2809–2823.

[20] J. Jiang, G. Han, H. Wang, M. Guizani, A survey on location privacy protection in wireless sensor networks, J. Netw. Comput. Appl. 125 (2019) 93–114.

[21] Z. Wu, G. Li, S. Shen, X. Lian, E. Chen, G. Xu, Constructing dummy query sequences to protect location privacy and query privacy in location-based services, World Wide Web 24 (2021) 25–49.

[22] P.S. Saravanan, S. Ramani, V.R. Reddy, Y. Farhaoui, A novel approach of privacy protection of mobile users while using location-based services applications, Ad Hoc Netw. (2023) 103253.

[23] V. Schmitt, Z. Li, M. Poikela, R.P. Spang, S. Möller, What is your location privacy worth? Monetary valuation of different location types and privacy influencing factors, in: Proceedings of the 16th ACM Conference on Security and Privacy in Wireless and Mobile Networks, 2023, pp. 19–29.

[24] L. Zhang, Y. Qian, M. Ding, C. Ma, J. Li, S. Shaham, Location privacy preservation based on continuous queries for location-based services, in: IEEE INFOCOM 2019-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), IEEE, 2019, pp. 1–6.

[25] Y. Li, X. Tao, X. Zhang, J. Liu, J. Xu, Privacy-preserved federated learning for autonomous driving, IEEE Trans. Intell. Transp. Syst. 23 (7) (2021) 8423–8434.

[26] M. Savitha, M. Senthilkumar, A unique secure multimodal biometrics-based user anonymous authenticated key management protocol (SMUAAKAP) based on block chain mechanism for generic HIoTNs, Theoret. Comput. Sci. 941 (2023) 77–90.

[27] L. Ma, Q. Pei, H. Xiao, H. Li, Z. Li, K. Fan, Edge computing enhanced privacy preserving for location based services, in: IEEE INFOCOM 2019-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), IEEE, 2019, pp. 1–6.

[28] A. Fathalizadeh, V. Moghtadaiee, M. Alishahi, On the privacy protection of indoor location dataset using anonymization, Comput. Secur. 117 (2022) 102665.

[29] J.W. Kim, K. Edemacu, J.S. Kim, Y.D. Chung, B. Jang, A survey of differential privacy-based techniques and their applicability to location-based services, Comput. Secur. 111 (2021) 102464.

[30] J. Smith, S.-F. Chang, Quad-tree segmentation for texture-based image query, in: Proceedings of the Second ACM International Conference on Multimedia, 1994, pp. 279–286.

[31] X. Xie, Z. Xiong, G. Zhou, G. Cai, On massive spatial data cloud storage and quad-tree index based on the Hbase, WIT Trans. Inf. Commun. Technol. 49 (2014) 691–698.

[32] Q. Ye, H. Hu, N. Li, X. Meng, H. Zheng, H. Yan, Beyond value perturbation: Local differential privacy in the temporal setting, in: IEEE INFOCOM 2021-IEEE Conference on Computer Communications, IEEE, 2021, pp. 1–10.

[33] A. Mitra, B. Bera, A.K. Das, Design and testbed experiments of public blockchain-based security framework for IoT-enabled drone-assisted wildlife monitoring, in: IEEE INFOCOM 2021-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), IEEE, 2021, pp. 1–6.

[34] C. Dong, Z. Xu, F. Jiang, S. Pal, C. Zhang, S. Chen, X. Liu, Bdfl: A blockchain-enabled fl framework for edge-based smart uav delivery systems, in: Proceedings of the Third International Symposium on Advanced Security on Software and Systems, 2023, pp. 1–11.

[35] H.J. Hadi, Y. Cao, K.U. Nisa, A.M. Jamil, Q. Ni, A comprehensive survey on security, privacy issues and emerging defence technologies for UAVs, J. Netw. Comput. Appl. 213 (2023) 103607.

[36] Q. Wang, R. Li, A weak consensus algorithm and its application to high-performance blockchain, in: IEEE INFOCOM 2021-IEEE Conference on Computer Communications, IEEE, 2021, pp. 1–10.

[37] V. Wylde, N. Rawindaran, J. Lawrence, R. Balasubramanian, E. Prakash, A. Jayal, I. Khan, C. Hewage, J. Platts, Cybersecurity, data privacy and blockchain: A review, SN Comput. Sci. 3 (2) (2022) 127.

[38] S. Pal, A. Hill, T. Rabehaja, M. Hitchens, A blockchain-based trust management framework with verifiable interactions, Comput. Netw. 200 (2021) 108506.

[39] H. Kim, J. Ben-Othman, L. Mokdad, On differential privacy-preserving movements of unmanned aerial vehicles, in: 2017 IEEE International Conference on Communications, ICC, IEEE, 2017, pp. 1–6.

[40] S. Enayati, D. Goeckel, A. Houmansadr, H. Pishro-Nik, Location privacy protection for UAVs in package delivery and IoT data collection, IEEE Internet Things J. (2023).

[41] A. Aljumah, T.A. Ahanger, I. Ullah, Heterogeneous blockchain-based secure framework for UAV data, Mathematics 11 (6) (2023) 1348.

[42] Z. Lv, L. Qiao, M.S. Hossain, B.J. Choi, Analysis of using blockchain to protect the privacy of drone big data, IEEE Netw. 35 (1) (2021) 44–49.

[43] B. Liu, W. Ni, R.P. Liu, Y.J. Guo, H. Zhu, Decentralized, privacy-preserving routing of cellular-connected unmanned aerial vehicles for joint goods delivery and sensing, IEEE Trans. Intell. Transp. Syst. (2023).

[44] X. Xu, H. Zhao, H. Yao, S. Wang, A blockchain-enabled energy-efficient data collection system for UAV-assisted IoT, IEEE Internet Things J. 8 (4) (2020) 2431–2443.

[45] H. Wang, H. Hong, L. Xiong, Z. Qin, Y. Hong, L-srr: Local differential privacy for location-based services with staircase randomized response, in: Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security, 2022, pp. 2809–2823.

[46] L. Zhu, H. Hong, M. Xie, J. Yu, DLPM: A dynamic location protection mechanism supporting continuous queries, Concurr. Comput.: Pract. Exper. 35 (19) (2023) e7495.

[47] C. Zhang, Y. Wang, W. Wang, H. Zhang, Z. Liu, X. Tong, Z. Cai, A personalized location privacy protection system in mobile crowdsourcing, IEEE Internet Things J. (2023).

[48] Z. Yao, L. Tan, J. Yi, L. Fu, Z. Zhang, X. Tan, J. Xie, K. She, P. Yang, W. Wu, et al., Sensitive data privacy protection of carrier in intelligent logistics system, Symmetry 16 (1) (2024) 68.

[49] E. Alptekin, M.E. Gursoy, Building quadtrees for spatial data under local differential privacy, in: IFIP Annual Conference on Data and Applications Security and Privacy, Springer, 2023, pp. 22–39.

[50] 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.

[51] Y. Li, Y. Qin, H. Wang, K-nearest neighbor privacy protection query for distributed storage in location-based service, Wirel. Pers. Commun. 121 (2021) 1509–1532.

[52] F. Chen, J. Wang, C. Jiang, T. Xiang, Y. Yang, Blockchain based non-repudiable iot data trading: Simpler, faster, and cheaper, in: IEEE INFOCOM 2022-IEEE Conference on Computer Communications, IEEE, 2022, pp. 1958–1967.

[53] Ethereum Foundation, Solidity: Contract-oriented programming language, 2021, URL https://soliditylang.org/, (Accessed: May 2023).

[54] T. Wang, J. Blocki, N. Li, S. Jha, Locally differentially private protocols for frequency estimation, in: 26th USENIX Security Symposium (USENIX Security 17), 2017, pp. 729–745.

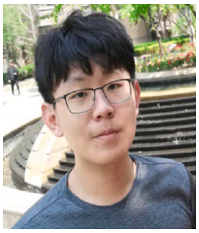
[55] T. Wang, N. Li, S. Jha, Locally differentially private frequent itemset mining, in: 2018 IEEE Symposium on Security and Privacy, SP, IEEE, 2018, pp. 127–143.

Aiting Yao received the bachelor's in the School of Mathematics and Statistics, Chaohu College, Hefei, Anhui and the master's in the School of Mathematics and Science, Anhui University, Hefei, Anhui, China in 2012 and 2020, respectively. She is currently pursuing the Ph.D. degree with the School of Computer Sciences and Technology, Anhui University, Hefei, Anhui, China. She is current research interests include mobile edge computing, privacy preserving, differential privacy, and federated learning.

Shantanu Pal is associated with the School of Information Technology, Deakin University, Melbourne, Australia. Shantanu has extensive research experience and knowledge in Internet of Things, big data and distributed smart applications, access control, trust management, blockchain technology, mobile and cloud computing, etc. He is a Senior Member of IEEE.

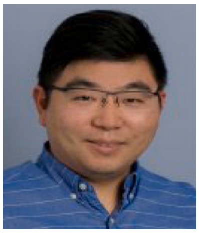
Xuejun Li received the Ph.D. degree in computer application technology from the School of Computer Science and Technology, Anhui University, Hefei, Anhui, China, in 2008. He is currently a Full Professor with the School of Computer Science and Technology, Anhui University, Hefei, Anhui, China. His major research interests include mobile edge computing, workflow systems, cloud computing, and intelligent software.

Zheng Zhang received the bachelor's in the School of Computer Sciences and Technology, Henan Polytechnic University, Jiaozuo, Henan, China in 2016-2020. He is currently pursuing a master's degree with the School of Computer Sciences and Technology, Anhui University, Hefei, Anhui. His current research interests include mobile edge computing, privacy preservation, and federated learning.



Chengzu Dong is a current Ph.D. candidate at Deakin University, he earned his first-class honors degree from Swinburne University of Technology. He has contributed to multiple high-quality conference papers and primarily focuses on research areas such as cybersecurity, machine learning, blockchain, unmanned aerial vehicle (UAV) delivery, and edge computing.

Frank Jiang is a mature and active mid-career researcher with over 15 years teaching experiences within Australia, holding a senior lectureship in cyber security at School of IT at Deakin University, Australia. He completed his Ph.D. degree in communication engineering and cyber security at University of Technology, Sydney (UTS). In 2011, he was awarded a highly prestigious UNSW Vice-Chancellor's Postdoctoral Fellowship (successful rate less than 4.5%) by University of New South Wales at the Australian Defence Force Academy for four years (2011-2014).



Xiao Liu received his Ph.D. degree in Computer Science and Software Engineering from the Faculty of Information and Communication Technologies at Swinburne University of Technology, Melbourne, Australia in 2011. He received his master's and bachelor's degrees from the School of Management, Hefei University of Technology, Hefei, China, in 2007 and 2004 respectively, all in Information Management and Information System. He is currently an Associate Professor at School of Information Technology, Deakin University, Melbourne, Australia. His research areas include Software Engineering, Distributed Computing and Service Computing, with special interests in workflow systems, cloud and edge computing, big data analytics, and human-centric software engineering. He is a Senior Member of IEEE and ACM.