Multi-sensor Data Privacy Protection with Adaptive Privacy Budget for IoT Systems

Xinyi Liu*, Ye Zheng*, Zhengxiong Li†, Yidan Hu*
*Department of Cybersecurity, Rochester Institute of Technology, Rochester, NY 14623 USA
†Department of Computer Science and Engineering, University of Colorado Denver, Denver, CO 80217 USA
xl9714@rit.edu, ye.zheng@mail.rit.edu, zhengxiong.li@ucdenver.edu, yidan.hu@rit.edu

2024 IEEE Conference on Communications and Network Security (CNS) | 979-8-3503-7596-1/24/$31.00 ©2024 IEEE | DOI: 10.1109/CNS62487.2024.10735696

***Abstract*—In the era of pervasive sensing and data-driven decision-making, the Internet of Things (IoT) has become ubiquitous, with sensors serving as the fundamental building blocks of IoT devices. However, sensor readings may contain sensitive personal information or be used to infer such information, raising significant privacy concerns. Local Differential Privacy (LDP) has become the de facto standard for numerical data privacy protection. To safeguard sensor readings in IoT systems, existing LDP solutions distribute the privacy budget evenly across multiple sensors for random perturbation. Unfortunately, this approach inevitably introduces excessive noise, significantly reducing the quality of IoT services.**

To address this deficiency, we propose impact-aware multi-sensor data privacy protection (IMapp) to provide rigorous privacy protection for sensor readings while maintaining high-quality IoT services. IMapp leverages the fact that sensor readings from different types of sensors have varied impacts on IoT services, adaptively distributing the privacy budget across multiple sensors according to their impacts. This approach enhances the quality of IoT services while ensuring guaranteed privacy protection. Additionally, IMapp incorporates a novel LDP mechanism that ensures rigorous privacy protection for sensors with arbitrary bounded domains. Theoretical analysis and evaluation results from three collected real datasets demonstrate that IMapp achieves the same level of multi-sensor data privacy as the existing solution while improving data fusion accuracy by up to two orders of magnitude.

***Index Terms*—Multi-sensor Data privacy, Local Differential Privacy, IoT Systems**

I. Introduction

The Internet of Things (IoT) has become omnipresent and widely adopted across various applications, including smart cities, smart homes, vehicle automation, and wearable computing [1], fundamentally altering how people interact with the physical world. At the core of this transformation are sensors, which serve as the foundational components of IoT devices. However, sensor readings often contain, or can be used to infer sensitive personal information, raising significant privacy concerns [2], [3]. In particular, even zero-permission motion sensors, such as accelerometers, gyroscopes, and magnetometers, can be exploited to infer users' sensitive information [3], [4], [5], highlighting the inadequacy of traditional access control methods in securing sensor data privacy. Examples of privacy risks associated with releasing data from zero-permission motion sensors include location inference [4], user input inference in health applications [5], and unauthorized access to permission-protected private information [6]. These situations underscore the critical need for effective mechanisms to ensure privacy protection for sensor data immediately after generation (i.e., before being released to anyone), particularly for data generated by zero-permission motion sensors.

Efforts to protect raw sensor data privacy can be broadly categorized into three approaches. First, cryptographic methods are commonly employed for secure sensor data protection [7], [8], [9], [10]. While these solutions offer robust privacy guarantees, they involve complexities in key management and incur higher computational costs. For instance, homomorphic cryptosystems (e.g., [7], [8], [10]) can be particularly costly for data aggregation, which may reduce their practicality. Furthermore, this paper focuses more on motion sensor data, which does not directly contain sensitive personal information and is thus less sensitive, making cryptographic approaches less cost-effective in this specific context. The second category involves sensor data anonymization techniques, including traditional k-anonymity, l-diversity, and emerging machine learning-based methods [11], [12], [13], [14], [15]. Despite their utility, these techniques remain vulnerable to re-identification attacks and do not guarantee rigorous privacy. The third line of solutions embraces Differential Privacy (DP) [16], a de facto paradigm for numerical data privacy. These solutions (e.g., [3]) adapt existing DP mechanisms, originally designed for numerical data, to achieve multi-sensor privacy protection. However, these approaches assume the existence of a trusted platform (e.g., data collector or server) for individual data privacy protection. This assumption renders the solution impractical, as the platform cannot be fully trusted, and immediate privacy protection is required before the data release.

More recently, Local Differential Privacy (LDP) mechanisms have been adopted to guarantee multi-sensor data privacy without the need for a trusted platform by distributing the privacy budget evenly for random perturbation, such as [17], [18]. Unfortunately, these approaches inevitably introduce excessive noise. As more sensors are added to an IoT system, each sensor receives a smaller portion of the privacy budget, resulting in increased noise and reduced quality of IoT services. Additionally, these solutions often directly apply existing LDP mechanisms initially designed for numerical data with unbounded or specific bounded domains, which makes them ineffective or even inapplicable for multi-sensor data privacy across various domains. Thus, there is a pressing need to design

advanced mechanisms to guarantee multi-sensor data privacy while maintaining the high quality of IoT services.

In this paper, we propose impact-aware multi-sensor data privacy protection (IMapp) to provide rigorous privacy protection while maintaining high-quality IoT services. We observe that sensor readings from different sensors have varied impacts on IoT services, meaning that changes in these readings affect service quality differently. Moreover, for a specific IoT service, the impact of sensor readings from a particular sensor on service quality remains consistent. For example, gyroscope data consistently has a more significant impact on orientation-based IoT services than accelerometer data. IMapp leverages these observations to estimate the heterogeneous impacts on a given IoT service in advance and then adaptively allocate the privacy budget across multiple sensors based on these impacts, thereby improving IoT service quality. Additionally, by adhering to the composition rule of the LDP mechanism [16], IMapp provides the same level of multi-sensor privacy protection for IoT systems as existing solutions like those in [17], [18]. IMapp operates in two phases: offline impact estimation, performed as a preliminary step, and online impact-aware data perturbation, which introduces noise to raw data during processing (see Fig. 1). Our contributions are summarized as follows:

- To the best of our knowledge, we are the first to provide rigorous data privacy protection under LDP for IoT systems involving multiple sensors with different domains.
- We propose the IMapp, a novel mechanism that maintains high-quality IoT services while ensuring LDP for multiple sensor data by adaptively distributing privacy budget based on their impact on IoT services. IMapp contains an efficient offline impact estimation method that enables adaptive assignment of the privacy budget. IMapp also incorporates a perturbation mechanism to guarantee privacy for numerical sensor data with various domains.
- We theoretically analyze IMapp's data privacy and thoroughly evaluate it using real datasets collected from three mobile devices. The results demonstrate that our solution outperforms the existing approach by reducing the service quality loss by up to two orders of magnitude.

The remainder of this paper is organized as follows: related work is discussed in Section II. Section III formulates the target problem and introduces background knowledge. Proposed IMapp is introduced in Section IV. We evaluate our solution and report the experimental results in Section V. We conclude our work in Section VI.

II. Related Work

A. Sensor Data Privacy Protection

Our work on multi-sensor data privacy in IoT systems is closely related to sensor data privacy protection, which can be categorized into three lines of approaches: cryptography, anonymization, and perturbation.

Cryptography is widely applied to protect mobile sensor data privacy, especially in crowdsensing systems [7], [8], [9], [10]. For instance, Miao et al. [7], [8] and Xiong et al. [10] applied homomorphic cryptosystems to achieve privacy-preserving data aggregation, while Xu et al. [9] utilized symmetric encryption for secure communication with cloud servers. Cryptography is primarily focused on protecting data sharing and can maintain high accuracy and security. However, it has several limitations. Encryption methods require robust key management and depend on the security of the chosen algorithm. Even though homomorphic cryptosystems support computation over encrypted data, their computational cost is relatively high.

Anonymization methods have also been employed to protect sensor data privacy. Liu et al. [19] proposed a k-anonymity method that clusters the records with similar quasi-identifiers. Many current approaches apply machine learning techniques to encode the data into a time-frequency domain and filter out sensitive features from the domain [11], [12], [13], [14], [15]. Introducing machine learning into traditional anonymization methods significantly decreases the risks of re-identification attacks. However, it also increases computational complexity due to the need for training models. Additionally, a major challenge with anonymization methods is that it is difficult to fully eliminate re-identification attacks while maintaining high utility, especially as the volume of data increases [20].

Perturbation techniques are another commonly employed way to enhance privacy protection. Abdallah et al. [3] analyzed privacy vulnerabilities and applied differential privacy (DP) at the hardware level to sensor data, adding a layer of protection to raw sensor data. Zheng et al. [17] introduced a method that aggregates similar fog-based IoT data subsets and applies LDP to these grouped subsets for enhanced privacy. Marchioro et al. [18] used LDP on IoT data within crowdsourcing platforms, where each privacy budget is allocated specifically to individual users to conserve the overall privacy budget. Zhao et al. [21] developed new LDP mechanisms and integrated them with federated learning to mitigate privacy threats while reducing communication costs between vehicles and cloud servers in crowdsourcing Internet of Vehicles (IoV) applications. Gao et al. [22] demonstrated another application of LDP-based federated learning over IoT sensing data. These perturbation techniques add noise to raw data, preventing the compromise of sensitive information. However, current approaches often treat all sensors as a single unit, applying the same level of perturbation across the board. This uniform approach can lead to excessive noise and diminished data utility, as individual sensors may contribute differently to the application.

B. Multi-Dimensional Numerical Data Perturbation

Our work is also closely related to protecting multi-dimensional numerical data privacy in DP. The first line of work aims to enable privacy-preserving multi-dimensional data release using DP [16]. For example, Xu et al. [23] present a differentially private algorithm, DPPro, for high-dimensional data release via random projection. In particular, DPPro projects data from a high-dimensional space to a randomly chosen lower-dimensional subspace to suppress the introduced noise and significantly improve data utility. Zhu et al. [24] highlighted the

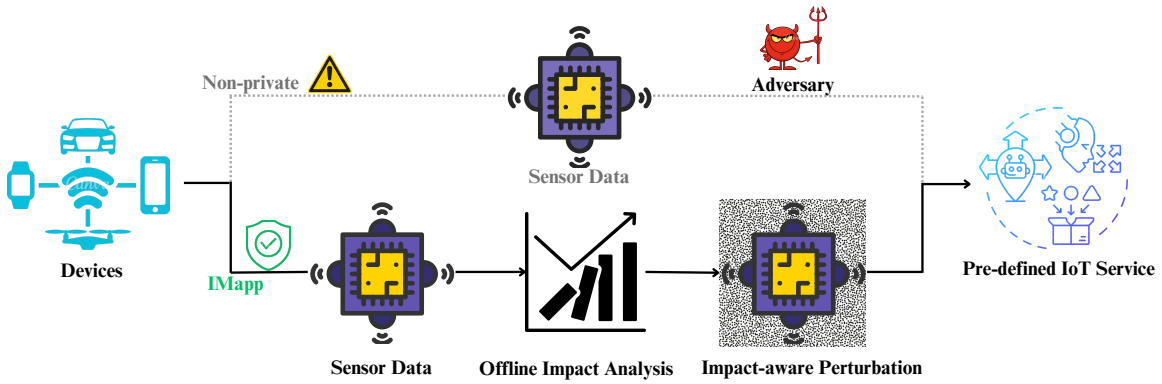


Figure 1: Overview of IMapp for multi-sensor privacy protection

importance of understanding correlated sensitivity in datasets and leveraged the correlation among different dimensions to identify the sensitivity for more effective privacy budgets across multi-dimensions, thereby reducing unnecessary noise addition and preserving data utility. Other examples include [25], [26] for numerical data privacy using DP. However, all the work assumes that a trusted data collector can be responsible for providing data privacy protection. In contrast, in our problem, we assume that the data collector, i.e., the platform, cannot be fully trusted, making this work inapplicable.

Another line of work focuses on LDP mechanisms for multi-dimensional privacy protection without needing a trusted platform. Wang et al. [27] propose the Piecewise Mechanism to provide rigorous privacy protection for numerical data with domain $[-1, 1]$ and then extend it for multi-dimension numerical data privacy protection by carefully selecting part of the multi-dimensional data for obfuscation and reports. However, an IoT system with multiple sensors has various data domains, and all sensor readings are required for services to function effectively. These two facts make the perturbation mechanism for a specific data domain inadequate, and selectively obfuscating only part of the multi-dimensional data is not feasible.

III. Preliminaries

This section first formulates the problem and then reviews the notion of LDP and Piecewise Mechanism for numerical privacy protection.

A. Problem Formulation

We consider an IoT system (e.g., AR/VR system or fitness monitoring system) that involves a user and an IoT service platform (e.g., Apps in mobile devices) provided by the third-party service provider. The user is equipped with multiple sensors, such as an accelerometer, gyroscope, and magnetometer. Let $S = \{S_1, S_2, \cdots, S_d\}$ be the sensors involved in the system, where d is the number of sensors. Each sensor S_i generates a numerical value $x_i \in \mathcal{X}_i$ at a specific time, where $\mathcal{X}_i$ is the domain of data from S_i, e.g. $[0, 2\pi]$ for gyroscope. When an IoT user wants to enjoy the IoT service, the user must provide the platform with sensor readings. However, directly sharing the sensor data would pose significant privacy and/or security risks [5], [6]. As a result, instead of submitting the original sensor readings $\mathbf{x} = \{x_1, x_2, \cdots, x_d\}$, the user would randomly perturb his/her sensor readings using a random perturbation mechanism, denoted as $\mathcal{M}$, and submits the perturbed sensor readings, $\mathbf{y} = \mathcal{M}(\mathbf{x})$, to the platform.

Upon receiving the perturbed sensor readings, $\mathbf{y}$, the platform would analyze the received $\mathbf{y}$ to provide specific IoT services. Let $f(\cdot)$ be a specific mechanism/procedure adopted by the platform to provide IoT service. The quality loss of the IoT services is indicated by $\delta = dis(f(\mathbf{y}), f(\mathbf{x}))$, where $dis(\cdot)$ is a distance function. In particular, considering that sensor fusion for sensor readings combination is a building block for many specific IoT services, such as sensor fusion in AR and VR systems for precise tracking of head and body movements. To ease the presentation, we use $f(\cdot)$ as a specific sensor fusion function hereafter.

We assume the platform is honest but curious: It faithfully carries out system operations to provide IoT services but is interested in inferring the original sensor readings. We also assume that the sensor readings generated by the equipment sensors are correct and stable without spoofing. We seek to design a novel randomized mechanism $\mathcal{M}$ to provide rigorous privacy protection while maintaining high-quality IoT services. In particular, the designed mechanism, $\mathcal{M}$, is expected to have the following nice properties:

- *Guaranteed privacy protection*: The designed randomized perturbation mechanism $\mathcal{M}$ should satisfy ϵ-LDP to provide guaranteed privacy protection. In particular, the probability of any particular output is nearly the same regardless of a possible sensor reading, making it difficult for the platform to infer true sensor reading from the reported obfuscated sensor reading.
- *High-quality of IoT services*: The platform can still provide IoT services with high quality. In particular, the quality loss of the IoT services δ is not very large.
- *Computation efficiency*: Our approach must be computationally efficient, only incurring low computation costs.

B. Local Differential Privacy

Definition 1 (**Local Differential Privacy (LDP) [28]**). *A randomized mechanism $\mathcal{M} : \mathcal{X} \to \mathsf{Range}(\mathcal{M})$ satisfies ε-LDP if and only if*

$$\frac{P(\mathcal{M}(x) = y)}{P(\mathcal{M}(x') = y)} \leq e^{\epsilon}, \tag{1}$$

for any inputs $x, x' \in \mathcal{X}$ and any output $y \in \mathsf{Range}(\mathcal{M})$, where $\mathsf{Range}(\mathcal{M})$ is the output range of $\mathcal{M}$.

When $\mathsf{Range}(\mathcal{M})$ is a continuous domain, the probability $P(\cdot)$ is replaced by probability density $pdf(\cdot)$. Here ϵ is a parameter controlling the level of privacy protection commonly referred to as *privacy budget*. The smaller the ϵ, the stronger the privacy protection, and vice versa. Intuitively, ϵ-LDP means that by observing the output y, the data collector (e.g., a platform) cannot infer whether the input is x or x' with high confidence, which provides users submitting sensitive data with plausible deniability. For example, $\epsilon = 0$ requires $\mathcal{M}$ maps two arbitrary inputs to any output y with the same probability, thus the output contains no distribution information of the input, making any inference from y powerless.

LDP has a nice composition property, which is detailed as follows:

Theorem 1 (**Composition Rule [16]**). *Suppose that $\mathcal{M}_1$, $\mathcal{M}_2$..., $\mathcal{M}_d$ is a set of randomization mechanisms. Each $\mathcal{M}_i : \mathcal{X}_i \to \mathsf{Range}(\mathcal{M}_i)$ satisfies ϵ_i-LDP for all $i \in \{1, 2, \dots d\}$. Their sequential combination $\mathcal{M} = (\mathcal{M}_1, \mathcal{M}_2, ..., \mathcal{M}_d) : \mathcal{X}_i^d \to \mathsf{Range}(\mathcal{M}_i)^d$, satisfies $\sum_{i=1}^{d} \epsilon_i$-LDP.*

The composition rule indicates that the privacy level of a multi-sensor system is the sum of the privacy budgets of all sensors.

C. Review of Piecewise Mechanism

Piecewise Mechanism (PM) is a state-of-the-art LDP method for numerical data privacy protection [27]. Specifically, denote $C = \frac{e^{\epsilon/2}+1}{e^{\epsilon/2}-1}$, for any numerical value $t_i \in [-1, 1]$, PM outputs the perturbed value $t_i^* \in [-C, C]$ according to the following probability density function:

$$pdf(t_i^* = x | t_i) = \begin{cases} p, & \text{if } x \in [l, r] \\ pe^{-\epsilon_i}, & \text{if } x \in [-C, l) \cup (r, C] \end{cases} \tag{2}$$

where $p = \frac{e^{\epsilon} - e^{\epsilon/2}}{2e^{\epsilon/2}+2}$, $l = \frac{C+1}{2} \cdot t_i - \frac{C-1}{2}$, and $r = l + C - 1$.

PM can be easily extended to multi-dimensional privacy protection by perturbing each dimension value with privacy $\frac{\epsilon}{d}$, where d is the number of dimensions that need to be perturbed [27].

IV. Perturbation Mechanism Design

In this section, we first introduce a baseline approach for rigorous privacy protection, followed by an overview of our solution for improving IoT service while providing the same level of privacy protection. Finally, we detail the design of our solution.

A. A Baseline Approach

We first introduce a baseline approach, a state-of-the-art multi-sensor data privacy protection mechanism under LDP [18]. Let $\mathbf{x} = \{x_1, \cdots, x_d\}$ be the d sensor readings, with ϵ as the total privacy budget. The baseline solution mainly consists of two primary steps. Firstly, the privacy budget ϵ is evenly divided across the d sensor readings. Next, each sensor reading x_i is randomly perturbed using a privacy budget of $\frac{\epsilon}{d}$ by applying an existing LDP mechanism initially designed for numerical data privacy protection, such as the representative Piecewise Mechanism [27] adopted in [18].

The baseline approach could provide rigorous privacy protection for multiple privacy protections. Specifically, if the randomized perturbation mechanism used for privacy protection for each specific sensor reading satisfies $\frac{\epsilon}{d}$-LDP, the mechanism satisfies ϵ-LDP according to the composition rule. However, the strong privacy guarantee comes from the sacrifice of the data utility. In particular, for a large d, the privacy budget assigned to each sensor reading would be pretty small and inevitably introduce a lot of noise for each sensor reading.

B. Design Rationale and Overview

Providing d sensor readings is essential to enhancing privacy in IoT systems. We propose an impact-aware multi-sensor data privacy protection (IMapp) method to generate obfuscated d sensor readings. This approach guarantees privacy protection while significantly improving the quality of the resulting IoT services. IMapp is designed based on the following key observations:

First, we observe that data from different types of sensors have varying impacts on IoT service quality. For example, slight alterations to gyroscope data can significantly affect orientation-based IoT services, such as the view angle in a VR system, while changing accelerometer values have less impact.

According to the Local Differential Privacy (LDP) composition rule, we also observed that a mechanism could provide a consistent overall privacy guarantee regardless of the specific privacy budget assigned to each sensor reading as long as the total privacy budget remains constant. As a result, we can allocate larger privacy budgets to sensor readings with a more significant impact while assigning smaller budgets to those with a lesser impact. This approach improves IoT service quality while maintaining the same level of privacy protection as the baseline approach.

Additionally, we observe that the procedure for analyzing multiple sensor readings, denoted by $f(\cdot)$, to support IoT services is predetermined and remains constant for a specific platform. Consequently, the impact of each sensor reading on IoT service quality is also constant, allowing us to compute their impacts before system use. As illustrated in Fig. 1, our solution works in two phases: sensor impact estimation and randomized perturbation design.

In what follows, we first introduce the method for calculating the impact of heterogeneous sensors on a specific predefined IoT service. Next, we detail how to distribute the privacy

budget across multiple sensor readings according to their impacts. Finally, we present a comprehensive design for privacy-preserving sensor data protection, incorporating adaptive privacy budgets.

C. Sensor Impact Estimation

Note that methods for analyzing multi-sensor readings in IoT services are often proprietary and predefined. We treat the entire analysis procedure as a black-box function, denoted by $f(\cdot)$. For example, in data fusion, $f(\cdot)$ takes multi-sensor data as inputs and outputs the sensor fusion results.

Let $\mathbf{x}_{-i}$ represent a possible combination of sensor readings from all sensors, excluding those from sensor S_i. We define the impact of the data from S_i on $f(\cdot)$ as:

$$\Delta_i = \max_{x_i, x_i', \mathbf{x}_{-i}} \frac{dis(f(x_i, \mathbf{x}_{-i}), f(x_i', \mathbf{x}_{-i}))}{x_i - x_i'}, \tag{3}$$

where x_i and x_i' are two possible sensor readings from S_i. The function $dis(\cdot)$ represents a distance metric, with Euclidean distance used as an example in this paper.

Intuitively, a brute-force approach can calculate Δ_i for each $i \in \{1, 2, \ldots, d\}$. Specifically, this method involves discretizing the continuous numerical domain into numerous possible points for each sensor. Subsequently, for each sensor S_i, we enumerate all possible combinations of $(x_i, x_i', \mathbf{x}_{-i})$ to determine the impact Δ_i. Although this brute-force method can provide an accurate impact estimation for each sensor, it is computationally expensive and impractical for real-world applications, particularly when a system involves many sensors and each sensor's domain is large. To address these challenges, we propose a more efficient approach for Δ_i estimation by 1) reducing the continuous numerical domains, and 2) designing a distribution-aware sampling method instead of relying on discretization-based enumeration.

1) Reducing Continuous Domains: We observe that certain ranges of sensor readings rarely occur in real-world scenarios. For example, the accelerometers in phones are designed to measure a wide range of accelerations, including those far beyond what humans can naturally produce without external forces or mechanical assistance. Thus, although the accelerometer's domain is large, e.g., $[-157, 157]$, actual readings during typical human activities are limited. We leverage the observation to narrow the domain of interest for impact calculation.

Let $D_i = [l_i, r_i]$ be the domain of sensor readings from sensor S_i. We reduce the original domain D_i to a new sub-range D_i' by focusing on the values typically generated by human activities. Specifically, we conduct various human activities offline to simulate typical usage of the IoT system and collect a large number of sensor readings, denoted as D_{off}. Let l_i' and r_i' be the minimal and maximal sensor readings from sensor S_i in the collected dataset. We then define the reduced domain D_i' as $[l_i', r_i']$ for impact calculation.

2) Distribution-aware Sampling: Ideally, given the D_{off} within the subdomains $\bigcup_{i=1}^{d} D_i'$, we could estimate a multi-dimensional data distribution across the d sensor readings and then repeatedly generate random samples from the multi-dimensional data distribution for impact estimation. However, obtaining the d-dimensional distribution with enough accuracy is difficult, especially when d is large. Fortunately, given the large D_{off}, it is relatively easy to obtain the marginal probability distribution of sensor readings for each sensor as well as the conditional probability distributions $P(X_j|X_i)$ for any two sensors S_i and S_j, where $i, j \in \{1, \cdots, d\}$ and $i \neq j$. We then use those marginal and conditional distributions to generate sample points sequentially for d sensors. Specifically, we randomly generate a sample x_1 from $P(X_1)$ for the first sensor S_1. Next, the subsequent samples x_i $(i > 1)$ are added to the sequence based on the conditional probability distribution.

Distribution estimation. We first estimate the marginal distribution of sensor readings, denoted by $P(X_i)$, for each sensor S_i. We split the focused subdomain D_i' into n_i intervals with equal size, $\{p_{i,1}, p_{i,2}, \cdots, p_{i,n_i}\}$. Next, we count the number of samples in D_{off} for each interval. Finally, we convert those counts to frequencies to roughly estimate the marginal distribution $P(X_i)$.

Similarly, we also estimate the joint probability distribution for the pair of sensor readings from any two sensors S_i and S_j. Let $\{p_{j,1}, p_{j,2}, \cdots, p_{j,n_j}\}$ be the n_j intervals from D_j'. We first count the number of samples for each pair of intervals $(p_{i,k}, p_{j,\tau})$ in D_{off}, where $k \in \{1, 2, \cdots n_i\}$ and $\tau \in \{1, 2, \cdots n_j\}$, and then covert those counts to frequencies to have the joint probability distribution $P(X_i, X_j)$. Then the conditional probability distributions can be calculated by

$$P(X_i = x_i | X_j = x_j) = \frac{P(X_i = x_i, X_j = x_j)}{P(X_j = x_j)} \tag{4}$$

for $i, j \in \{1, \cdots, d\}$ and $i \neq j$.

Sequential sampling. We randomly shuffle the d sensors and individually generate a sample for each sensor. Without loss of generality, we assume the sampling order is from S_1 to S_d. We randomly draw a sample x_1 from the distribution $P(X_1)$ for the first sensor S_1. For the subsequent sensor S_t, where $t > 1$, we randomly draw its sample according to the conditional distribution $P(X_i|x_j)$, where x_j is the previously added sample. Algorithm 1 shows this sampling procedure.

Algorithm 1: Distribution-aware Sampling

```
Input: {P(X_1), ..., P(X_d)}, and {P(X_i|X_j) : i, j ∈ {1, ..., d}, i ≠ j};
Output: Sampled results Y;
Y ← ∅;                                  ▷ Initialize the sampled results
Randomly draw x_1 from P(X_1);
Y ← Y ∪ x_1;
while 1 < t ≤ d do
    Randomly draw x_t based from P(X_t|x_{t-1});
    Y ← Y ∪ x_t;
    t ← t + 1;                          ▷ The next sensor
end while
return Y;
```

3) Impact Estimation: We now estimate the impact Δ_i for each sensor S_i. Suppose we have repeatedly sampled n sets of points from the distribution-aware sampling, in which each set contains sensor readings $\{x_1, \cdots, x_d\}$.

Algorithm 2: Sensor Impact Estimation

```
1  Input: Offline sensor readings D_off
2  Output: Sensor impact Δ_1, Δ_2, ··· Δ_d
3  for i in {1, ..., d} do
4      Redefine D'_i based on D_off and divide it into
         intervals {p_{i,1}, p_{i,2}, ..., p_{i,n_i}};
5      Estimate P(X_i) from D_off via counting;
6      for j in {i+1, ··· , d} do
7          Estimate P(X_i, X_j) from D_off via counting;
8          Calculate P(X_i|X_j) using Eq. (4);
9      end for
10 end for
11 while repeat n times do
12     Y ← Algorithm 1;  ▷ Disttibution-aware sampling
13     for i in {1, ..., d} do
14         while repeat r times do
15             Randomly draw x'_i from P(X_i);
16             Compute EI_{i,i'} using Eq. (5) with Y;
17         end while
18         Compute EI_i using Eq. (6);
19     end for
20 end while
21 Calculate each Δ_i using Eq. (7);
22 return Δ_1, Δ_2, ... Δ_d;
```

For each data point consisting of d sensor readings, the following process is iterated r times. First, a perturbed value x'_i is randomly drawn from the marginal distribution $P(X_i)$. For each x'_i selected, the corresponding element impact is calculated using the formula:

$$EI_{i,i'} = \frac{dis(f(x_i, \mathbf{x}_{-i}), f(x'_i, \mathbf{x}_{-i}))}{x_i - x'_i}, \tag{5}$$

where $(x_i, \mathbf{x}_{-i})$ denotes the selected point from the distribution-aware sampling, and $\mathbf{x}_{-i} = \{x_1, \cdots, x_i, \cdots, x_d\} \setminus x_i$ represents all sensor readings excluding x_i. The term $(x'_i, \mathbf{x}_{-i})$ refers to the point after replacing x_i with x'_i.

For each data point x_i, the element impact $EI_{i,i'}$ is estimated r times, and the average element impact $\overline{EI}_i$ is computed as follows:

$$\overline{EI}_i = \frac{\sum_{j=1}^{r} EI_{i,i'}}{r}. \tag{6}$$

Note that we have n sets of sensor readings, thus n data points of x_i. To estimate the impact in Eq. (3), we need to take the maximum among them. Formally, let $\overline{EI}_i^k$ $(1 \le k \le n)$ be the averaged element impact from Eq. (6), the impact of sensor S_i is then determined by:

$$\Delta_i = \max\{\overline{EI}_i^1, \overline{EI}_i^2, \ldots \overline{EI}_i^n\}. \tag{7}$$

We summarize the above procedure in Algorithm 2.

Computation Complexity. We now analyze the computation complexity of the proposed impact estimation algorithm. As we can see in Algorithm 2, generating sampling Y (line 12) and computing the element impact $EI_{i,i'}$ (line 16) have the highest computation cost, which is $O(nd)$ and $O(nmr)$, respectively. Thus, the overall computation cost of Algorithm 2 is $\max\{O(nd), O(nmr)\}$, which is affordable. Notably, the impact estimation could be done offline, which makes our solution very practical.

D. Impact-aware Perturbation

This subsection details the procedures of impact-aware perturbation, comprising two online phases: *Budget Allocation* and *Data Perturbation* with the assigned budget.

1) Budget Allocation: Given a privacy budget ϵ, we now allocate the ϵ according to the computed sensor impacts Δ_i. Specifically, we first normalize those impacts, $\Delta_1, \Delta_2, \cdots, \Delta_d$, from Algorithm 2, which is given by

$$\Delta'_i = \frac{\Delta_i}{\sum_{j=1}^{d} \Delta_j} \tag{8}$$

Next, we compute the assigned privacy budget ϵ_i for the sensor reading from sensor S_i by

$$\epsilon_i = \epsilon \times \Delta'_i \,. \tag{9}$$

2) Data perturbation: Then, we perturb each sensor reading x_i with the assigned privacy budget ϵ_i. We employ the Piecewise Mechanism [27], which was particularly designed for local differential privacy protection on a bounded numerical domain. However, the original Piecewise Mechanism takes the input within domain $[-1, 1]$. We extend it to arbitrary bounded domain D_i by a *mapping-perturbation-remapping* procedure.

Mapping. This step maps the sensor reading's domain D_i to $[-1, 1]$. Denote $D_i = [l_i, r_i]$ and $g(x_i) : D_i \to [-1, 1]$ is the domain mapping function, then

$$g(x_i) = \frac{2}{r_i - l_i} \cdot x_i - \frac{l_i + r_i}{r_i - l_i} \tag{10}$$

linearly maps each $x_i \in D_i$ to $[-1, 1]$. Then, we can perturb $g(x_i)$ using the Piecewise Mechanism.

Perturbation. Given privacy budget ϵ_i, Piecewise Mechanism perturbs the mapped data $g(x_i)$ to y_i according to the following sampling distribution:

$$pdf(y_i = t | g(x_i)) = \begin{cases} p, & \text{if } t \in [l, r] \\ pe^{-\epsilon_i}, & \text{if } t \in [-C, l) \cup (r, C] \end{cases} \tag{11}$$

where $C = \frac{e^{\epsilon_i/2}+1}{e^{\epsilon_i/2}-1}$ and

$$\begin{aligned} p &= \frac{e^{\epsilon_i} - e^{\epsilon_i/2}}{2e^{\epsilon_i/2} + 2}, \\ l &= \frac{C+1}{2} \cdot g(x_i) - \frac{C-1}{2}, \\ r &= l + C - 1. \end{aligned} \tag{12}$$

This mechanism outputs perturbed $y_i \in [-C, C]$. We then map it back to the sensor's reading domain D_i by remapping.

Remapping. Similar to the mapping procedure, domain $[-C, C]$ can be remapped to D_i via $h(y_i) : [-C, C] \to D_i$

$$h(y_i) = \frac{r_i - l_i}{2C} \cdot y_i + \frac{l_i + r_i}{2}. \tag{13}$$

Then, each perturbed data $h(y_i)$ is sent to perform the multi-sensor fusion.

Theorem 2. *Denote $\mathcal{M}_i$ by the Piecewise Mechanism with privacy budget ϵ_i, the mapping-perturbation-remapping procedure $\mathcal{M}_i^+ = h \circ \mathcal{M}_i \circ g$ satisfies ϵ_i-LDP.*

Proof. We prove the theorem according to the definition of ϵ-LDP. Specifically, we need to prove

$$\frac{pdf(\mathcal{M}_i^+(x_i) = v)}{pdf(\mathcal{M}_i^+(x_i') = v)} \leq e^{\epsilon_i} \tag{14}$$

holds for any x_i, x_i' pair and any v. Note that g, h are linear and invertible functions having no randomness, then

$$\begin{aligned} pdf(\mathcal{M}_i^+(x_i) = v) &= pdf(h \circ \mathcal{M}_i \circ g(x_i) = v) \\ &= pdf(\mathcal{M}_i \circ g(x_i) = h^{-1}(v)), \end{aligned} \tag{15}$$

which reduces the proof to considering $\mathcal{M}_i : [-1, 1] \to [-C, C]$. Piecewise Mechanism $\mathcal{M}_i$ with privacy budget ϵ_i satisfies ϵ_i-LDP [27], so $\mathcal{M}_i^+$ also satisfies ϵ_i-LDP. □

Theorem 3. *The proposed IMapp satisfies ϵ-LDP*

Proof. According to Theorem 2. The random perturbation mechanism $\mathcal{M}_i^+$ satisfies ϵ_i-LDP for each sensor reading privacy protection. Next, according to the composition rule introduced in Theorem 1, IMapp satisfies $\sum_{i=1}^{d} \epsilon_i$-LDP. Also, note that according to the budget allocation strategies introduced in Eqs. (8) and (9), we have $\sum_{i=1}^{d} \epsilon_i = \epsilon$. As a result, IMapp satisfies ϵ-LDP, and the theorem is proved. □

V. Evaluation

In this section, we evaluate the performance of the proposed IMapp using a collected real dataset.

A. Datasets

We evaluated our method by collecting three real multi-sensor datasets from three distinct smartphone models: the Samsung Galaxy S10 (SG 10), Google Pixel 6 (GP 6), and OnePlus 10 Pro (OP 10).

We first developed a user interface using Flutter [29] to collect data from the built-in sensors of mobile phones efficiently. This interface recorded sensor data during user movements while the application was in use, facilitating data collection for various IoT applications, such as motion tracking, navigation, and orientation estimation. Our dataset comprises measurements from the accelerometer, gyroscope, and magnetometer sensors. These zero-permission sensors were selected to provide a broad spectrum of hardware capabilities and sensor specifications, ensuring a thorough evaluation of our proposed privacy-preserving sensor fusion mechanism.

We summarize the collected datasets, including their sizes N and the domains for each sensor, in Table I. These values represent the number of multidimensional sensor readings collected and the range of sensor readings, respectively.

B. Experimental Settings

In this experiment, we consider two different data fusion functions as case studies: the Revised Madgwick Algorithm [30] and the Complementary Filter [31].

The first fusion function, Madgwick's algorithm, is designed to estimate orientation efficiently using inertial measurement units (IMUs). The revised version of this algorithm further improves accuracy by incorporating a gradient descent method to minimize the error between the measured and estimated orientations [30], [32]. The second, the Complementary Filter, is also widely used, especially in IMUs. It combines high-frequency data from gyroscopes with low-frequency data from accelerometers to produce more accurate orientation estimations [33].

To evaluate the impact of different sensors on these fusion functions, we performed 100,000 repetitions for each fusion function on each dataset during the impact analysis phase. We mainly consider 9 sensor variables in the experiments, with privacy budgets allocated based on the calculated impact from Algorithm 2.

We compare our solution, IMapp, with the baseline approach presented in [18]. We omit comparison with the existing solution in [17] because it employs the Laplacian mechanism, which [18] has already demonstrated to be less effective than the Piecewise Mechanism used in this paper.

We use the Mean Square Error (MSE) metric to quantitatively evaluate the performance of the IMapp mechanism. Specifically, we randomly select $T = 1,000$ entries as the ground truth set $\mathbf{x}$. Let $\mathbf{y}$ be the corresponding perturbed sensor reading vectors. The experimental MSE is given by

$$MSE = \frac{1}{T} \sum_{i=1}^{T} (f(\mathbf{y}) - f(\mathbf{x}))^2, \tag{16}$$

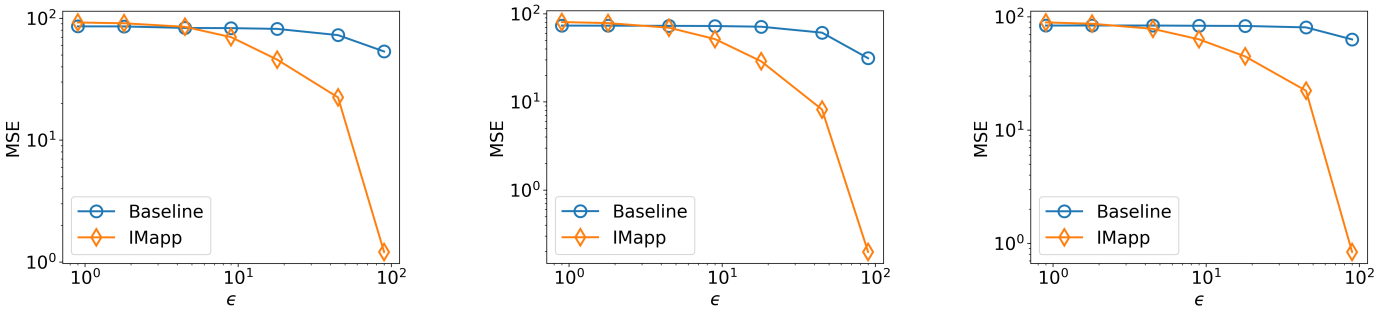
where $f(\cdot)$ is a specific data fusion function, such as the Revised Madgwick algorithm or the Complementary Filter used in this paper.

C. Experimental Results

1) Accuracy: Figs. 2a to 2c compare the MSE between the baseline and IMapp methods for the Revised Madgwick Algorithm [30] as the total privacy budget ϵ increases from 0.9 to 90. As ϵ increases, the MSE under both mechanisms decreases.

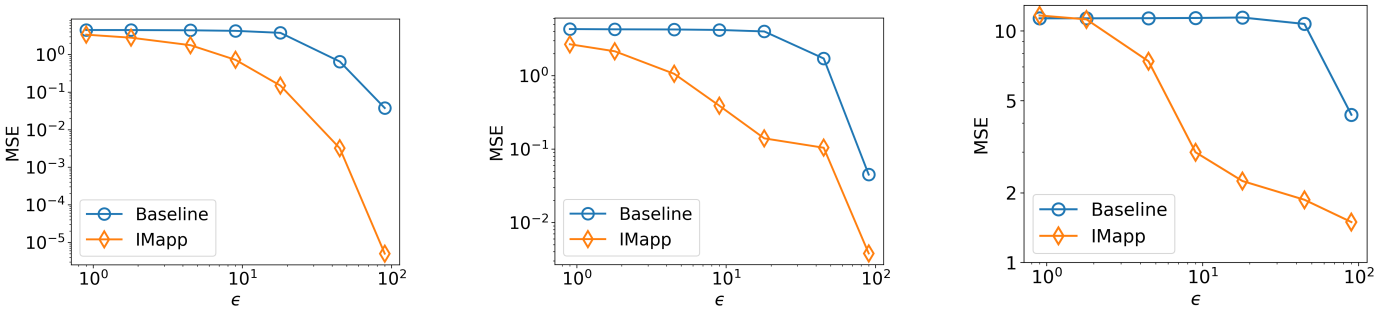
Table I: Dataset sizes and sensor domains for different devices

	N	Accelerometer	Gyroscope	Magnetometer
SG 10	$3,506$	$[-78, 78]$	$[-17.5, 17.5]$	$[-2,000, 2,000]$
GP 6	$3,571$	$[-157, 157]$	$[-34.9, 34.9]$	$[-3,198, 3,198]$
OP 10	$8,496$	$[-157, 157]$	$[-34.9, 34.9]$	$[-3,000, 3,000]$



(a) SG 10 (b) GP 6 (c) OP 10

Figure 2: MSE of two mechanisms for Revised Madgwick Algorithm across three datasets



(a) SG 10 (b) GP 6 (c) OP 10

Figure 3: MSE of two mechanisms for Complementary Filter across three datasets

This outcome is expected because a higher privacy budget results in obfuscated sensor readings closer to the original ones, thereby improving the accuracy of the fusion function when $\epsilon \leq 4.5$, the MSEs of the baseline and IMapp methods are comparable. However, as ϵ increases beyond 4.5, the MSE for IMapp declines rapidly, achieving more than ten times better results than those of the baseline approach. This is because, with a small ϵ (e.g., 0.9), the assigned privacy budget to each sensor reading is very low, making the obfuscated sensor readings very noisy and reducing the impact of different sensors on the fusion results. Conversely, with a larger ϵ, allocating a larger privacy budget to the sensor readings with higher impact significantly improves the accuracy of data fusion results, allowing IMapp to outperform the baseline approach by a large margin. Notably, from Fig. 2c for the dataset OP 10, we can see that when ϵ is large, e.g., $\epsilon = 90$, the proposed IMapp outperforms the baseline approach by reducing the MSE by up to two orders of magnitude.

Figs. 3a to 3c illustrate the MSE under the two approaches when the Complementary Filter is used for multi-sensor fusion. We can see that the MSEs under both schemes decrease as ϵ increases from 0.9 to 90 due to the same reason discussed in Fig. 2. We can also see that IMapp consistently outperforms the baseline approach with a smaller MSE for different data fusion functions and privacy budgets across three datasets. Moreover, IMapp can improve the accuracy of data fusion by up to two orders of magnitude with a relatively large privacy budget, demonstrating its superior performance.

These results demonstrate that IMapp can achieve a significantly reduced MSE while providing the same level of privacy protection as the baseline.

2) Efficiency: We evaluated the running time for impact evaluations across the three datasets and two fusion functions in Fig. 4, where each result is the accumulated running time after $100,000$ repetitions. The running time varies significantly between different fusion functions. For example, the Revised Madgwick Algorithm has a faster processing time, resulting in a significantly shorter running time for its impact evaluation than the Complementary Filter. This discrepancy is due to the computational complexity inherent in each fusion algorithm. Moreover, we would like to point out that even for the complementary filter, we can complete the varied sensor impact estimation within several minutes for $100,000$ repeated testing, which is very affordable. Additionally, it is also worth noting that sensor impact estimation can be conducted offline during the pre-processing stage. Since this process is carried out before real-time operations, its high computational cost does not affect the online multi-sensor data perturbation needed for real-time IoT services. This offline nature ensures that the efficiency of the impact evaluation process does not adversely impact overall system performance during operation.

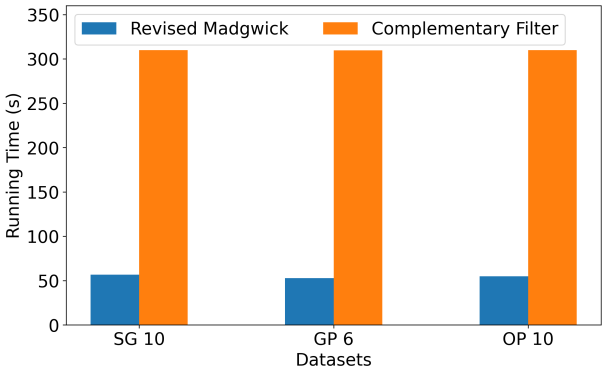


Figure 4: Running Time for two fusion functions across three datasets.

VI. Conclusion and Future works

Theoretical analysis and thorough performance evaluations confirm that IMapp can guarantee privacy protection, provide high-quality IoT services, and improve computation efficiency.

There are many directions to extend this work. Firstly, we aim to apply IMapp to support machine-learning-based IoT services with a balanced utility-privacy trade-off. Additionally, we plan to expand our solution to encompass multi-sensor data privacy protection over extended periods by leveraging temporal correlations.

Acknowledgements

This work was supported in part by the US National Science Foundation under grants CNS-2245689 (CRII) and 2426470, as well as the 2022 Meta Research Award for Privacy-Enhancing Technologies.

References

[1] D. Sehrawat and N. S. Gill, "Smart sensors: Analysis of different types of IoT sensors," in *IEEE 3rd International Conference on Trends in Electronics and Informatics (ICOEI)*, pp. 523–528, 2019.

[2] Y. Dong, Y. Hu, A. Aseeri, D. Li, and R. Zhang, "Location inference under temporal correlation," in *IEEE 32nd International Conference on Computer Communications and Networks (ICCCN)*, pp. 1–10, 2023.

[3] M. Abdallah and A. ElMougy, "Protection against side-channel attacks on multifusion zero-permission sensors using differential privacy," in *IEEE 44th LCN Symposium on Emerging Topics in Networking (LCN Symposium)*, pp. 92–99, 2019.

[4] S. Narain, T. D. Vo-Huu, K. Block, and G. Noubir, "Inferring user routes and locations using zero-permission mobile sensors," in *IEEE Symposium on Security and Privacy (SP)*, pp. 397–413, 2016.

[5] S. Zhang, Y. Liu, and M. Gowda, "I spy you: Eavesdropping continuous speech on smartphones via motion sensors," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 6, no. 4, 2023.

[6] K. Sun, C. Xia, S. Xu, and X. Zhang, "StealthyIMU: Stealing permission-protected private information from smartphone voice assistant using zero-permission sensors," in *NDSS*, 2023.

[7] C. Miao, L. Su, W. Jiang, Y. Li, and M. Tian, "A lightweight privacy-preserving truth discovery framework for mobile crowd sensing systems," in *IEEE INFOCOM'17*, pp. 1–9, 2017.

[8] C. Miao, W. Jiang, L. Su, Y. Li, S. Guo, Z. Qin, H. Xiao, J. Gao, and K. Ren, "Privacy-preserving truth discovery in crowd sensing systems," *ACM Transactions on Sensor Networks (TOSN)*, vol. 15, no. 1, 2019.

[9] G. Xu, H. Li, S. Liu, M. Wen, and R. Lu, "Efficient and privacy-preserving truth discovery in mobile crowd sensing systems," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 4, pp. 3854–3865, 2019.

[10] J. Xiong, R. Ma, L. Chen, Q. Li, X. Liu, and Z. Yao, "A personalized privacy protection framework for mobile crowdsensing in IIoT," *IEEE Transactions On Industrial Informatics*, vol. 16, no. 6, pp. 4231–4241, 2020.

[11] M. Malekzadeh, R. G. Clegg, A. Cavallaro, and H. Haddadi, "Mobile sensor data anonymization," *Proceedings of the International Conference on Internet of Things Design and Implementation (IoTDI)*, pp. 49–58, 2019.

[12] N. Debs, T. Jourdan, A. Moukadem, A. Boutet, and C. Frindel, "Motion sensor data anonymization by time-frequency filtering," in *28th European Signal Processing Conference (EUSIPCO)*, pp. 1707–1711, 2021.

[13] M. Malekzadeh, R. G. Clegg, A. Cavallaro, and H. Haddadi, "Privacy and utility preserving sensor-data transformations," *Pervasive and Mobile Computing*, 2020.

[14] O. Hajihassani, O. Ardakanian, and H. Khazaei, "Latent representation learning and manipulation for privacy-preserving sensor data analytics," in *IEEE Second Workshop on Machine Learning on Edge in Sensor Systems (SenSys-ML)*, pp. 7–12, 2020.

[15] P. Rouge, A. Moukadem, A. Dieterlen, A. Boutet, and C. Frindel, "Anonymizing motion sensor data through time-frequency domain," in *IEEE 31st International Workshop on Machine Learning for Signal Processing (MLSP)*, pp. 1–6, 2021.

[16] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Foundations and Trends in Theoretical Computer Science*, pp. 211–407, 2014.

[17] L. Zheng, T. Zhang, R. Qin, Y. Shen, and X. Mu, "Privacy-preserving subset aggregation with local differential privacy in fog-based IoT," in *Mobile Multimedia Communications*, pp. 399–412, 2021.

[18] T. Marchioro, A. Kazlouski, and E. P. Markatos, "Practical crowdsourcing of wearable IoT data with local differential privacy," in *Proceedings of the 8th ACM/IEEE Conference on Internet of Things Design and Implementation (IoTDI)*, pp. 275–287, 2023.

[19] F. Liu and T. Li, "A clustering k-anonymity privacy-preserving method for wearable IoT devices," *Security and Communication Networks*, 2018.

[20] T. Basso, R. Matsunaga, R. Moraes, and N. Antunes, "Challenges on anonymity, privacy, and big data," in *Seventh Latin-American Symposium on Dependable Computing (LADC)*, pp. 164–171, 2016.

[21] Y. Zhao, J. Zhao, M. Yang, T. Wang, N. Wang, L. Lyu, D. Niyato, and K.-Y. Lam, "Local differential privacy-based federated learning for internet of things," *IEEE Internet of Things Journal*, vol. 8, no. 11, pp. 8836–8853, 2021.

[22] J. Gao, M. Tang, T. Wang, and B. Campbell, "PFed-LDP: A personalized federated local differential privacy framework for IoT sensing data," in *Proceedings of the 20th ACM Conference on Embedded Networked Sensor Systems*, pp. 835–836, 2023.

[23] C. Xu, J. Ren, Y. Zhang, Z. Qin, and K. Ren, "DPPro: Differentially private high-dimensional data release via random projection," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 12, pp. 3081–3093, 2017.

[24] T. Zhu, P. Xiong, G. Li, and W. Zhou, "Correlated differential privacy: Hiding information in non-IID data set," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 2, pp. 229–242, 2015.

[25] Y. Hu and R. Zhang, "Differentially-private incentive mechanism for crowdsourced radio environment map construction," in *IEEE INFOCOM'19*, pp. 1594–1602, 2019.

[26] C. Ma, L. Yuan, L. Han, M. Ding, R. Bhaskar, and J. Li, "Data level privacy preserving: A stochastic perturbation approach based on differential privacy," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 4, pp. 3619–3631, 2023.

[27] N. Wang, X. Xiao, Y. Yang, J. Zhao, S. C. Hui, H. Shin, J. Shin, and G. Yu, "Collecting and analyzing multidimensional data with local differential privacy," *IEEE ICDE'19*, pp. 638–649, 2019.

[28] J. C. Duchi, M. I. Jordan, and M. J. Wainwright, "Local privacy and statistical minimax rates," in *IEEE Symposium on Foundations of Computer Science (FOCS)*, pp. 429–438, 2013.

[29] "Flutter." https://docs.flutter.dev/.

[30] xioTechnologies, "Fusion." https://github.com/xioTechnologies/Fusion.

[31] M. Garcia, "AHRS." https://github.com/Mayitzin/ahrs, 2021.

[32] S. O. H. Madgwick, *AHRS algorithms and calibration solutions to facilitate new applications using low-cost MEMS*. University of Bristol.

[33] W. T. Higgins, "A comparison of complementary and kalman filtering," *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-11, no. 3, pp. 321–325, 1975.