Privacy-preserving mHealth Data Release with Pattern Consistency

Mohammad Hadian[1], Xiaohui Liang[1], Thamer Altuwaiyan[1], and Mohamed M E A Mahmoud[2]

[1]Department of Computer Science, University of Massachusetts Boston

[2]Department of Electrical and Engineering, Tennessee Technological University

Email: {mshadian,xiaohui,thamerfa}@cs.umb.edu; mmahmoud@tntech.edu

Abstract

Mobile healthcare system integrating wearable sensing and wireless communication technologies continuously monitors the users' health status. However, the mHealth system raises a severe privacy concern as the data it collects are private information, such as heart rate and blood pressure. In this paper, we propose an efficient and privacy-preserving mHealth data release approach for the statistic data with the objectives to preserve the unique patterns in the original data bins. The proposed approach adopts the bucket partition algorithm and the differential privacy algorithm for privacy preservation. A customized bucket partition algorithm is proposed to combine the database value bins into buckets according to certain conditions and parameters such that the patterns are preserved. The differential privacy algorithm is then applied to the buckets to prevent an attacker from being able to identify the small changes at the original data. We prove that the proposed approach achieves differential privacy. We also show the accuracy of the proposed approach through extensive simulations on real data. Real experiments show that our partitioning algorithm outperforms the state-of-the-art in preserving the patterns of the original data by a factor of 1.75.

I. INTRODUCTION

The mobile healthcare (mHealth) system with emerging wearable devices and wireless communications has removed geographical and distance related barriers of healthcare and has made the

continuously-collected health data available anytime and anywhere [1]–[4]. However, the mHealth system raises a severe privacy concern to its users because the mHealth data usually contain private information [5], such as heart rate and blood pressure, at a highly fine-grained level, which may be maliciously used to derive patients' sensitive information, such as daily activities and health conditions. Keeping privacy of the data is the most important challenge of the emerging mHealth system [3], [6].

Differential Privacy (DP) [7] has been studied to help release data with privacy protection and accuracy assurance. DP can simply be explained as the mechanism of randomizing the results of the given query by adding some noise, commonly generated through Laplace distribution, to the original results based on a privacy budget ϵ to preserve the privacy of the individual records in the database. DP can guarantee mathematically-provable and measurable privacy preservation because of its precise definition and proof of privacy protection [8]. DP has been applied to release the data of the histogram in Fig. 1 (left) [9], [10]. With DP, it can be guaranteed that an attacker cannot distinguish the original histogram and the histogram with any bin plus 1 or minus 1. However, the aggregate mHealth data are much different from histogram bins from three perspectives (shown in Fig. 1): i) The bin values usually represent real values but not counts; ii) The bins can be generated at a high frequency, such as heart rate being continuously monitored by wearable devices at a rate of one per minute; and iii) The bins may be interpreted with special diagnosis needs and some particular pattern consistency is required. All these distinctive features make the existing DP mechanisms [10], [11] inefficient when applied to the mHealth data.
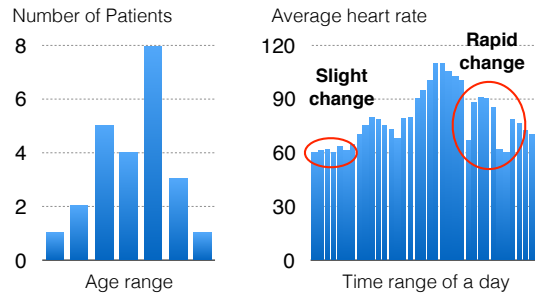

Fig. 1. Histogram and mHealth data bins

In this paper, we focus on an efficient and privacy-preserving data release approach specially designed for the mHealth data. We observe that the pattern consistency is particularly important for the mHealth data analysis. If the pattern of the original data is removed for the privacy concern, we consider such privacy-preserving approach is unacceptable in the mHealth data analysis. Here, we particularly study two pattern consistency requirements: i) if the difference of the values of two bins are larger than a threshold, the relationship of the outputs corresponding to the two bins must be preserved; and ii) if the difference of the values of two adjacent bins are larger than a threshold, the relationship of the

outputs must be preserved. Naturally, the difference threshold for the adjacent bins is smaller than the first one to distinguish between rapid and slight changes. The motivation behind these requirements is to enable healthcare providers to observe the patients' health conditions from the preserved patterns. Specifically, we propose a novel differential privacy mechanism to release the aggregate query results of the mHealth data as shown in Fig. 1 (right). To preserve the defined pattern, the DP mechanism consists of two stages: the first stage is the private partitioning of the database bins into DP-compliant buckets, and the second stage is the noise addition to the average value in each bucket for data release to achieve DP. Specifically, the contributions of this paper can be summarized as follows:

- First, we study the patterns occurring in the mHealth data and observe that there are two distinguishable patterns of "rapid change" and "slight change" in these data, which the current privacy preserving techniques are unable to distinguish and preserve them in the final result. The slight change is referred to the minor changes that happen over the time, while the rapid changes happen between two adjacent data bins.

- Second, we propose an efficient and privacy-preserving mHealth data release mechanism. A new pattern-preserving partitioning algorithm is developed to preserve the rapid changes from the slight changes in the released data.

- Third, we prove that the proposed mechanism achieves $\epsilon-$differential privacy. We further obtain the accuracy analysis results through an extensive simulations using real data set. The results show that our algorithm achieves pattern consistency while other existing mechanisms cannot.

II. PRELIMINARIES

A. Histogram and privacy

Histograms, are commonly used to aggregate information to represent statistical data, as shown in Fig. 1 (left). The statistical data usually are considered to be privacy-preserving because the attacker is unable to derive the information about a single data record. However, if the database allows the attacker to run multiple queries on histograms without any constraints, the attacker is able to find out a newly-added or deleted data record by simply running the same query before and after the operation. For example, in Fig. 1 (left), the attacker queries the number of patients for all the age ranges of n and $(n+1)$ consecutive days. Then, the attacker is able to derive the age of a patient admitted on $(n+1)$-th day, which is not intended to be disclosed.

B. Differential privacy

Differential privacy (DP) mechanisms are recently proposed to address the above privacy problem. The idea is to tolerate a small change in the database and prevent the attacker from being able to tell the change. Here we detail the DP definitions.

DP places a bound (controlled by privacy budget ϵ) on the difference in the probability of algorithm outputs for any two neighboring databases. For any given database instance I, let $nbrs(I)$ denote the set of neighboring databases which differ from I in at most one record; i.e., if $I' \in nbrs(I)$, then $|(I - I') \cup (I' - I)| = 1$.

Definition 1. A randomized algorithm $\mathcal{A}$ is $\epsilon-$Differentially Private if for all instances I, any $I' \in nbrs(I)$, and any subset of outputs $S \subseteq Range(\mathcal{A})$, the following holds:

$$Pr[\mathcal{A}(I) \in S] \leq exp(\epsilon) \times Pr[\mathcal{A}(I') \in S]$$

Definition 2 (Sensitivity of a query)**.** Given a sequence of counting queries Q, the sensitivity of Q, denoted as ΔQ, is:

$$\Delta Q = max||Q(I) - Q(I')||_1$$

where $I' \in nbrs(I)$.

Laplace mechanism: We use $Lap(b)$ to denote the Laplace probability distribution with mean 0 and scale b. The Laplace mechanism is commonly used to achieves DP by adding Laplace noise to a query output.

Definition 3. Let Q be a query sequence of length d and $\mathcal{Z}$ be a d-length vector of random variables where $\mathcal{Z}_i \sim Lap(\Delta Q_i/\epsilon)$. The Laplace mechanism $\tilde{Q}$ is defined as:

$$\tilde{Q}(I) = Q(I) + \mathcal{Z}$$

The randomized algorithm $\tilde{Q}$ is ϵ-differentially private.

Differential private algorithm has two properties.
- Sequential composition: If there are n independent algorithms $\mathcal{A}_1, ..., \mathcal{A}_n$, whose privacy budgets are $\epsilon_1, ..., \epsilon_n$ respectively, any function $\mathcal{K}$ of them: $\mathcal{K}(\mathcal{A}_1, ..., \mathcal{A}_n)$ is $\sum_{i=1}^{n} \epsilon_i$-differentially private.
- Parallel composition: If the previous mechanisms are computed on disjoint subsets of the private dataset then the function $\mathcal{K}$ would be $max\{\epsilon_i\}$-differentially private.

III. Proposed scheme

In this section, we introduce the proposed privacy-preserving data release scheme.

A. Scheme overview

The overview of our scheme is shown in Fig. 2. Consider a scenario where the user has a wearable watch to monitor her health data continuously. The watch uploads the data to the database through smartphones and wireless communications. The database is always updated with the latest health data. A querier can send query to the database and obtain the statistic information about the user. However, the responses should not disclose any single data record to the querier. There are four steps for the query process.
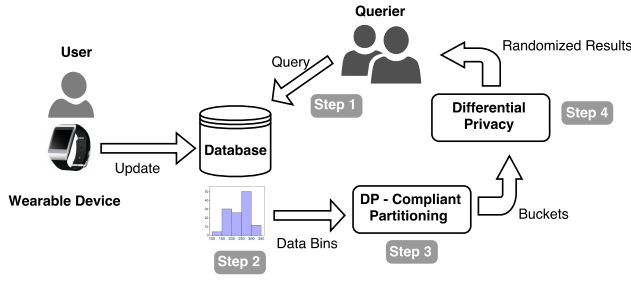

Fig. 2. System Model

Step 1: Query generation: querier sends the query sequence Q to the database. The querier in this case could be the personal healthcare provider of the user or any other party (e.g. doctor, hospital, etc.) who needs to monitor the health conditions of the user. The sequence of queries can be average heart rate of the users for every 10 minute of her daily life over a week or month. This query can help the doctor to monitor the general pattern of changes in her patient's heart rate on a very fine-grained basis.

Step 2: Aggregate query generation: based on the queries, the database records would be aggregated into data bins. For instance the database may contain user's heart rate for each minute but the querier (i.e. doctor) may only be interested in monitoring the information on a 5 minute basis. Thus the data bins generated by the database aggregates information of 5 records into a single bin.

Step 3: DP-compliant partitioning: the bins resulted from previous step will be then partitioned into a set of buckets based on the values of each bin, structure of the data and user defined threshold values indicating the level of granularity required by the querier. The purpose of this partitioning is to achieve more efficiency over answering the queries and also it is proven that coarser granularity achieved by a proper bucketing will increase the accuracy of the randomization [10]. This process needs to use randomization which complies with differential privacy requirements because the structure of the buckets may reveal information and due to small changes in the database, private information in the database can be inferred. Also, because of general nature of aggregation and randomization, the patterns in the original data will usually be removed during this process. Our partitioning algorithm is proven to be differentially private and preserve the patterns of the original data.

Step 4: Randomizing Algorithm: the query results of buckets from previous step will be randomized by adding noise from Laplace distribution to preserve the privacy of the data. This step simply follows the data-independent Laplace mechanism which adds noise to the results regardless of the inputs. Then the results would be sent to the querier.

The first two steps are widely studied and the process suggested for them is fairly straightforward and standard, thus we skip further details about them and focus on the last two steps especially the third step as the main goal of this work.

B. DP-Compliant Partitioning

The partitioning of the data bins into the buckets has to be complying with differential privacy requirements because it has to be guaranteed that the bucketing is probabilistically the same for two neighboring databases, in other words, adding or removing a record from the database should be tolerated through a random variable. The process of partitioning itself follows a simple triple thresholded scheme:

1) Algorithm setup and initialization: The variable declarations and initializations should be done before the start of the algorithm process itself. The Min, Max are non-negative numbers used to hold the values of maximum and minimum of bins in each bucket. Integers $i, j, size$ are used to hold the indexes of current bin, current bucket and size of current bucket respectively. Also $current$ is used to hold the value of previous

bin (i.e. x_{i-1}). Three following threshold parameters are learned from public accessible information and are set based on user and querier's setup:

- T_D: this value bound the maximum possible difference in maximum and minimum values of a single bucket to insure the uniformity of each bucket. (i.e. the bin values can dramatically change if this threshold is too large.)

- T_L: the maximum possible length of buckets is bounded by T_L in order to evade creating over-sized buckets.

- T_R: this threshold which is naturally selected to be smaller than T_D, ensures that changes in the data are observed and differentiated from changes which generally will be captured by T_D, because the change in two adjacent bins may actually be smaller than the T_D threshold but since it has happened in a very small period of time, it has important information in it and has to be preserved and reflected for the doctor. For instance increasing heart rate during daily activities will be distinguished from changes during workout.

Due to the privacy requirement of the partitioning algorithm, it is necessary to randomize T_D and T_R threshold parameters. Thus, using the Laplace mechanism, with privacy budget ϵ_1 we produce random noises Y, Y' which will be added to T_D and T_R and form $\hat{T}_D$ to $\hat{T}_R$ respectively.

2) Partitioning process: The partitioning process is done using an efficient and simple process of scanning from beginning of the data domain to the end (i.e. first to the last bin) with possible single backtracks during scan. The process starts with placing the first bin into the fist bucket and continues to next bins. In case uniformity of the bins complies with threshold requirements, the bin would be added to the same bucket, otherwise a new bucket would be created and process continues with the new bucket. The first condition to check is the T_R threshold because of its smaller value (line 15). In case of this condition not satisfying, two single bin buckets (each containing one of the unbalanced bins) are required to be created. Based on the size of the current bucket, three different cases are considerable (lines 16, 27 and 31). Furthermore, after this condition the two remaining thresholds would be examined (line 38) and based on that condition, either a new bucket would be created or the current bucket would be enlarged. Creating new bucket is simply done by pushing the current bucket (i.e. b_j) into the result set (i.e. B) and incrementing j followed by resetting $size$ to 0.

C. Randomizing Algorithm

After partitioning of the data into buckets, adding noise to the average value of bins in each bucket would simply satisfy DP as long as the added noise is generated with proper setting and noise scale. In

---
Algorithm 1 Private Partitioning Algorithm

1: **Input:** database D, T_D, T_L, T_R, ϵ_1

2: **Output:** a set of histogram buckets B

3: *Initialization*: Set $size = 0; i = 1; j = 1$, $B = \emptyset$,

4: $\hat{T}_D = T_D + Y, \hat{T}_R = T_R + Y'$ $\triangleright$ $Y, Y' \sim Lap(1/\epsilon_1)$

5: **while** $i \leq length(D)$ **do**

6:     $current = NULL;$

7:     **if** $(size == 0)$ **then**

8:         $b_j \leftarrow x_i$ $\triangleright$ Put bin x_i into bucket b_j

9:         $Min = Max = current = x_i;$

10:         $current = x_i; size + +; i + +;$ $\triangleright$ Goto next bin

11:     **end if**

12:     $Max = max(Max, x_i); Min = min(Min, x_i)$

13:     **if** $(current \neq NULL$ and $|current - x_i| > \hat{T}_R)$ **then**

14:         **if** $size == 0$ **then**

15:             **if** $(B[-1].length > 1)$ **then**

16:                 $last = B.pop(); b_j = last.pop()$

17:                 $B \leftarrow last; B \leftarrow b_j; j + +;$

18:                 $b_j \leftarrow x_i; B \leftarrow b_j;$

19:                 $j + +; current = x_i; i + +; size = 0;$

20:             **else** $\triangleright$ Last bucket is already single bin

21:                 $b_j \leftarrow x_i; B \leftarrow b_j$

22:                 $j + +; current = x_i; i + +; size = 0;$

23:             **end if**

24:         **else if** $size == 1$ **then**

25:             $B \leftarrow b_j; j + +; b_j \leftarrow x_i; B \leftarrow b_j; j + +;$

26:             $current = x_i; size = 0; i + +$

27:         **else if** $size > 1$ **then**

28:             $last = b_j.pop(); B \leftarrow b_j; j + +;$

29:             $b_j \leftarrow last; B \Leftarrow b_j; j + +;$

30:             $b_j \leftarrow x_i; B \leftarrow b_j; j + +;$

31:             $current = x_i; size = 0; i + +$

32:         **end if** $\triangleright$ $size$ check

33:     **end if** $\triangleright$ $\hat{T}_R$ check

34:     **if** $((Max - Min \leq \hat{T}_D)$ and $(size \leq T_L))$ **then**

35:         $b_j \leftarrow x_i$ $\triangleright$ Put bin x_i into bucket b_j

36:         $current = x_i; size + +; i + +;$ $\triangleright$ Goto next bin

37:     **else**

38:         $B \leftarrow b_j$ $\triangleright$ Done with this bucket

39:         $current = x_i; size = 0; j + +;$ 8 $\triangleright$ Goto next bucket

40:     **end if**

41:   **end while**

this step we use the Laplace distribution for generating noise. Considering the maximum possible change in values of bins in neighboring databases is α, since value of each bucket is the average value of its bins and having buckets with the $size = 1$ is considerable, the Laplace scale for this step has to be $b = \frac{\alpha}{\epsilon}$ for achieving $\epsilon-$DP.

IV. PROOF OF PRIVACY

This part shows the proof of privacy for Algorithm 1. Let d_0, d_1 be neighboring databases and $\mathcal{A}(d_0), \mathcal{A}(d_1)$ be the output of the algorithm on these databases; Max_{ik}, and Min_{ik} be the maximum and minimum value of bins in ith bucket b_i of d_k. Y and Y' are Laplace random variables and $f_y, f_{y'}$ are their density function, therefor $\hat{T}_D = T_D + Y$ and $\hat{T}_R = T_R + Y'$ are the randomized threshold values. To show that the algorithm is $\epsilon-$differentially private, we need to show that result of partitioning of d_0 and d_1 are probabilistically equivalent, thus it is sufficient to prove: $Pr(\mathcal{A}(d_0) = B) \leq e^\epsilon \times Pr(\mathcal{A}(d_1) = B)$. The parameters effective in partitioning results are the threshold values (i.e. T_L, $\hat{T}_D$ and $\hat{T}_R$), but T_L is not a random variable, therefor only $\hat{T}_D$ and $\hat{T}_R$ are effective in randomness of the result. Suppose the maximum difference in value of bins in two neighboring databases is bounded by α. This α can be learned from public accessible data based on the type of queries and maximum possible values for the under study filed of the record. For each bucket, we have $Max_i - Min_i < \hat{T}_D$ and $|current - x_i| < \hat{T}_R$. These two inequalities can be generalized to the whole database:

$$\frac{Pr(\mathcal{A}(d_0) = B)}{Pr(\mathcal{A}(d_1) = B)} \leq e^\epsilon \Leftrightarrow \mathcal{X} = \left(\frac{\prod_{b_i \in d_0} Pr(Max_{i0} - Min_{i0} < \hat{T}_D)}{\prod_{b_i \in d_1} Pr(Max_{i1} - Min_{i1} < \hat{T}_D)} \right.$$
$$\left. \times \frac{\prod_{b_i \in d_0} Pr(|current - x_{i0}| < \hat{T}_R)}{\prod_{b_i \in d_1} Pr(|current - x_{i1}| < \hat{T}_R)} \right) \leq e^\epsilon$$

Using the *sequential composition* property of DP, taking $\epsilon = \epsilon_1 + \epsilon_2$, we have:

$$\mathcal{X} = \left(\frac{\prod_{b_i \in d_0} Pr(Max_{i0} - Min_{i0} < \hat{T}_D)}{\prod_{b_i \in d_1} Pr(Max_{i1} - Min_{i1} < \hat{T}_D)} \leq e^{\epsilon_1} \right)$$
$$\times \left(\frac{\prod_{b_i \in d_0} Pr(|current - x_{i0}| < \hat{T}_R)}{\prod_{b_i \in d_1} Pr(|current - x_{i1}| < \hat{T}_R)} \right) \leq e^{\epsilon_2})$$

Thus if both parts of this equation holds, the algorithm is $\epsilon-$differentially private. We try to solve these inequalities (i.e. $\mathcal{X}_1$ and $\mathcal{X}_2$) in order to find the required Laplace distribution scale for satisfying $\epsilon-$DP. Suppose the changed record falls into bucket b_i (ith bucket of d_k). Altering d_0 or d_1 are equivalent, thus

we only consider one case. For the first inequality (i.e. $\mathcal{X}_1$), if the changed record is between minimum and maximum values of the bucket (i.e. $Min_{i0} \leq x_{i0} \leq Max_{i0}$), then the Max_{i0} and Min_{i0} will not change and $Pr(Max_{i0} - Min_{i0} < \hat{T}_D) = 1 \leq e^{\epsilon_1}$ for every ϵ_1. But if the changed value effects either Max_{i0} or Min_{i0}, we need to find the suitable Laplace scale ($b = s/\epsilon_1$) in order to have this change tolerated. The Max_{i0} and Min_{i0} are changes by α units and to the value of the $\mathcal{X}_1$, increasing Min_{i0} is equivalent to decreasing Max_{i0} and vice versa, thus we only consider change of Max_{i0}. We take $Y \sim Lap(s/\epsilon_1), t = Max_{i0} - Min_{i0}$ and $u = t - T_D$, then we consider two cases of changing Max_{i0}:

1) If $Max_{i0} \leftarrow Max_{i0} + \alpha$:

$$\mathcal{X}_1 = \frac{Pr(t + \alpha < \hat{T}_D)}{Pr(t < \hat{T}_D)} = \frac{Pr(Y > u + \alpha)}{Pr(Z > u)} < 1 \leq e^{\epsilon_1}$$

2) If $Max_{i0} \leftarrow Max_{i0} - \alpha$:

$$\mathcal{X}_1 = \frac{Pr(t - \alpha < \hat{T}_D)}{Pr(t < \hat{T}_D)} = \frac{Pr(Y > u - \alpha)}{Pr(Z > u)} = \frac{\int_{u-\alpha}^{+\infty} f_y(y)dy}{\int_{u}^{+\infty} f_y(y)dy} \leq e^{\epsilon_1}$$

We consider following cases in order to solve the above inequality:

• $u \geq \alpha : \mathcal{X}_1 = e^{\epsilon_1/s} \leq e^{\epsilon_1} \Rightarrow \epsilon_1/s \leq \epsilon_1 \Rightarrow 1/s \leq 1 \Rightarrow s \geq 1$

• $0 < u < \alpha :$

$$\mathcal{X}_1 = \frac{\frac{1}{2} + \int_{u-\alpha}^{+\infty} f_y(y)dy}{\int_{u}^{+\infty} f_y(y)dy} = \frac{2 - e^{\frac{u-\alpha}{b}}}{e^{\frac{-u}{b}}} \leq e^{\epsilon_1}$$

Taking $v = e^{\frac{u}{b}}$, then $\mathcal{X}_1 = 2v - e^{\frac{-\alpha}{b}} v^2 \leq e^{\epsilon_1} \Rightarrow s \geq \alpha$

• $u \leq 0 :$

$$\mathcal{X}_1 = \frac{\frac{1}{2} + \int_{u-\alpha}^{0} f_y(y)dy}{\frac{1}{2} + \int_{u}^{0} f_y(y)dy} = \frac{2 - e^{\frac{u-\alpha}{b}}}{2 - e^{\frac{u}{b}}} \leq e^{\epsilon_1}$$

$$\Leftrightarrow e^{\epsilon_1}(e^{u\epsilon_1})^{\frac{1}{s}} - [e^{(u-\alpha)\epsilon_1}]^{\frac{1}{s}} \leq 2e^{\epsilon_1} - 2$$

Taking $s = \alpha$, the inequality above holds. Thus for the first part of the equation (i.e. $\mathcal{X}_1$), the Laplace scale $b = \frac{\alpha}{\epsilon_1}$ is sufficient for DP.

For the second part (i.e. $\mathcal{X}_2$), we have to make sure the difference between changed record and its next (and previous) record will be tolerated through $\hat{T}_R$ and will not result in changing the buckets. Considering the differences between values of consecutive bins, increasing the larger value is equivalent to decreasing the smaller value and vice versa, thus considering only changes in one value (i.e. $x_{i0} \leftarrow x_{i0} + \alpha$ and $x_{i0} \leftarrow x_{i0} - \alpha$) is sufficient for the proof. We have:

$$\mathcal{X}_2 = \frac{Pr(|x_{(i-1)0} - x_{i0}| < \hat{T}_R)}{Pr(|x_{(i-1)1} - x_{i1}| < \hat{T}_R)} \times \frac{Pr(|x_{(i+1)0} - x_{i0}| < \hat{T}_R)}{Pr(|x_{(i+1)1} - x_{i1}| < \hat{T}_R)}$$

Also, relative change between $x_{(i-1)1}$ and x_{i1} is equivalent to change between $x_{(i+1)1}$ and x_{i1}, thus only considering one case is sufficient for the proof. Considering the large value in the adjacent bins as Max_{i0} and the smaller vales as Min_{oi} through the exact same argument as the argument for previous part, the proof of privacy for $s = \alpha \Rightarrow b = \frac{\alpha}{\epsilon_2}$ is immediate.

V. EVALUATION

In this section, we evaluate the performance of our newly-designed algorithm. We have conducted real experiments on captured heart rates from wearable devices attached to a user during two weeks. Heart rate usually ranges between 50 and 210, and thus we choose $\alpha = \frac{160}{14}$ as the sensitivity. Obviously, for a bigger time period of data collection, the required Laplace scale for achieving same privacy budget gets smaller as a result of less sensitivity due to aggregation over more data. The experiments are run on a Linux machine with an Intel Core i7 3.5 GHz processor and 8 GB of DDR3 memory.

A. Pattern preservation evaluation

We compare our algorithm and the state-of-the-art proposed by Li el al. [9] in terms of capturing the patterns of the original data and reflecting them in the final output. To do so we define pattern preservation percentage as $\frac{detected\_rapid\_changes}{all\_rapid\_changes\_in\_data} \times 100$ Our algorithm uses an additional threshold parameter T_R as maximum difference allowed for two adjacent bins to be combined into one bucket. In case this threshold is violated, our algorithm divides the two bins into two single-bin buckets in order to emphasis on this rapid change as opposed to Li's algorithm which does not capture rapid changes. In our designed experiment we used $T_L = 4, T_D = 30$ and $T_R = 15$ for our algorithm and for Li's algorithm we used two settings for their T_D to be as small as our T_R and also same value as our T_D, and have assigned the same value for their T_L as for ours. We tested both algorithms on a set of collected heart rate per minute data stored for two weeks. Fig. 3 is a sample result of partitioning and randomizing of both algorithms. The black arrows are rapid changes which are captured by our algorithm. The arrow marked by (1) is an example of failing Li's algorithm with $T_D = 30$ in capturing the rapid change and (2) is an example in which both Li's algorithms fail in doing so. As shown in Fig. 4 (a) our algorithm does better than Li's in preserving the pattern of original data with the same T_D threshold. In case of selecting Li's T_D to be as small as our T_R, obviously every rapid change in data is also captured as subset of $Max - Min$ factor which Li's algorithm is focused on, thus almost (because of the randomization effect) equal preservation percent is reasonable in this case.

The very important point is that our algorithm other than detecting the rapid change, using the idea of creating single bin buckets at the places of rapid change is preserving the change from being smoothed

out through the averaging in the bucket. Specifically, as shown in Fig. 4, our algorithm outperforms Li's (with the same T_D values) in percentage of preserved rapid changes with 70.81% compared to 40.1%. As mentioned Li's algorithm with $T_D = 15$ is able to perform almost equal to ours (68.36%) but may lose the found pattern by averaging bin values over their bucket. The values are computed by averaging over results of 1000 experiments.
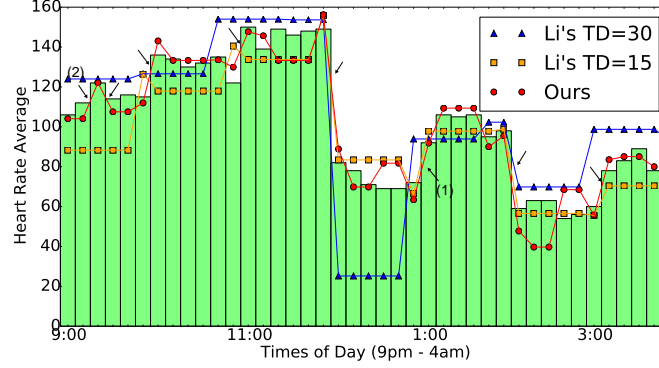

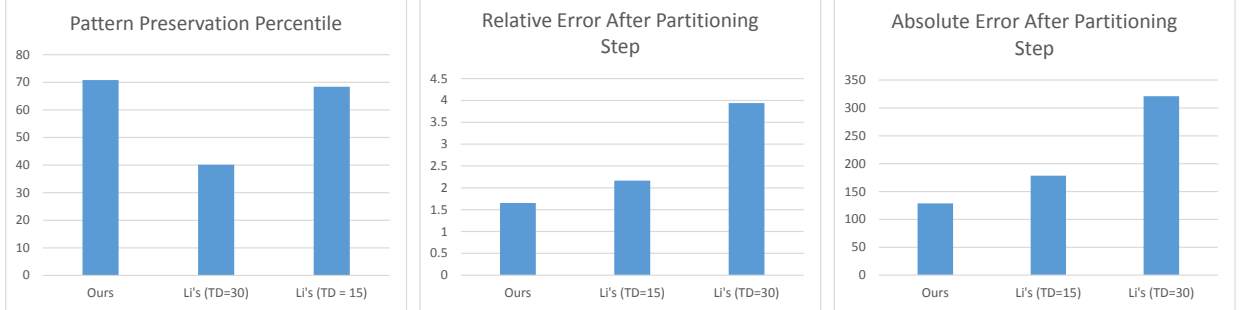Fig. 3. Pattern preservation results based on 10 minutes query sequence

B. Error analysis

Fig. 4. (a) Pattern preservation percentile. (b) & (c) Relative and absolute error comparison comparison

In this subsection we compare our algorithm with Li's in terms of absolute and relative error values in two steps and argue that preserving the pattern will result in higher accuracy (less error) because of the structural benefit gained in our algorithm. The reason is better uniformity between bins in a bucket which makes the averaging closest to the actual values. Obviously failing in dividing the unbalanced bin from others will result in higher error in averaging and leads to less accuracy at the end. It is important to mention that in the later steps of the scheme because of added significant Laplace noise (compared to error values) drawn from the same scale in both ours and Li's algorithms, only the errors in partitioning step are comparable in two algorithms. As shown in Fig. 4 (b, c) our algorithm outperforms Li's in both absolute and relative error metrics at partitioning step.

12

C. Time complexity

Due to the higher complexity of our algorithm than Li's, also considering the efficiency of their algorithm which does the partitioning in only one scan from the beginning to the end of the data (i.e. $O(n)$), a marginal increase in time complexity is acceptable as a trade-off for more accurate pattern preservation. Specifically our algorithm also scans the data only once, thus our algorithm has the same time complexity at Li's. The only overhead is in case of the T_R threshold violation in which there are several cases considerable and more computations necessary to handle the case. As average over 1000 experiments, Li's times for only partitioning and both steps are 0.734 ms and 1.012 ms and ours does it on 0.763 ms and 1.09 ms respectively.

VI. RELATED WORKS

Security and privacy in eHealth data has been widely studied in the literature [4]–[6] because of the private and sensitive nature of the date dealt with. The mechanism of mHealth [1] is a more recent extension to eHealth for revolutionizing the healthcare systems though mobile communications and has been highly attracted lately [12] and all privacy concerns raised in eHealth are applicable to mHelath. Our work is specifically focused on privacy preservation in mHealth data and considers specific requirements of it due to the sensitive nature of its data. While standard methods of secrecy hide the content of the message, covert communication in wireless environments [13], [14] and computer networks [15], [16] hides the existence of the communication.

Differential Privacy [7], [8], [17] is a newly emerged mechanism for private data publication with strong mathematical proof of privacy preservation. Dwork et al. [8] introduced the Laplace mechanism (LM) for generating noise, which is commonly used in the literature. LM uses a notion of sensitivity of the query (Δ) for finding the proper Laplace scale (b). In this work we use the same method of sensitivity and Laplace scale.Private partitioning of histograms under differential privacy has been widely studied. Blum et al. [18] have introduced a one-dimensional histogram, while Xiao et al. [19] have suggested a multi-dimensional one using a wavelet-based technique. Xu et al. [11] have classified and compared different existing approaches for histogram publishing that takes the structure of the bins into account. They have proposed two different algorithms for publication and have stated that the granularity of the partitioning has impact on accuracy of the results. A new partitioning algorithm achieving higher accuracy has been introduced in [10] by C. Li et al. and the performance of their method in partitioning and update is further improved by H. Li et al. in [9] but neither these works consider capturing the slight and rapid changes patterns in the original data which is main goal of our work.

VII. CONCLUSION

In this paper, we proposed a novel private partitioning scheme for mHealth data under DP that preserves the patterns of the original data and reflects it to the results. We gave strict privacy proof to show this scheme is differentially private. Also through evaluation in real experiment we showed that the pattern of the original data is mostly reflected into the final results as opposed to the current state-of-the-art which is not able to properly detect and preserve the pattern. We showed our algorithm gains more accuracy due to its benefit from pattern preservation. In our future work, we will explore more characteristics of mHealth data in order to gain more utility for the querier, and specifically we will focus on different methods for improving the accuracy of pattern preservation and the results through relative error.

REFERENCES

[1] R. S. Istepanian, S. Laxminarayan, and C. S. Pattichis, "M-health: Emerging mobile health systems," *Topics in Biomedical Engineering International Book Series*, 2006.

[2] X. Liang, X. Li, M. Barua, L. Chen, R. Lu, X. Shen, and H. Luo, "Enable pervasive healthcare through continuous remote health monitoring," *IEEE Wireless Communications*, vol. 19, no. 6, pp. 10–18, 2012.

[3] R. Lu, X. Lin, and X. Shen, "Spoc: A secure and privacy-preserving opportunistic computing framework for mobile-healthcare emergency," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 3, pp. 614–624, 2013.

[4] W. R. Hersh, "Medical informatics: improving health care through information," *Journal of the American Medical Association*, vol. 288, no. 16, pp. 1955–1958, 2002.

[5] N. Dong, H. Jonker, and J. Pang, "Challenges in ehealth: From enabling to enforcing privacy," in *Foundations of Health Informatics Engineering and Systems*, 2011, pp. 195–206.

[6] T. Sahama, L. Simpson, and B. Lane, "Security and privacy in ehealth: Is it possible?" in *IEEE 15th International Conference on e-Health Networking, Applications and Services (Healthcom)*, 2013, pp. 249–253.

[7] C. Dwork, "Differential privacy," in *Encyclopedia of Cryptography and Security, 2nd Ed.*, 2011, pp. 338–340.

[8] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory of cryptography*. Springer, 2006, pp. 265–284.

[9] H. Li, Y. Dai, and X. Lin, "Efficient e-health data release with consistency guarantee under differential privacy," in *IEEE 17th International Conference on e-Health Networking, Applications and Services (Healthcom)*, 2015.

[10] C. Li, M. Hay, G. Miklau, and Y. Wang, "A data-and workload-aware algorithm for range queries under differential privacy," *Proceedings of the VLDB Endowment*, vol. 7, no. 5, pp. 341–352, 2014.

[11] J. Xu, Z. Zhang, X. Xiao, Y. Yang, G. Yu, and M. Winslett, "Differentially private histogram publication," *The VLDB Journal*, vol. 22, no. 6, pp. 797–822, 2013.

[12] M. Kay, J. Santos, and M. Takane, "mhealth: New horizons for health through mobile technologies," *World Health Organization*, pp. 66–71, 2011.

[13] R. Soltani, B. Bash, D. Goeckel, S. Guha, and D. Towsley, "Covert single-hop communication in a wireless network with distributed artificial noise generation," in *Communication, Control, and Computing (Allerton), 2014 52nd Annual Allerton Conference on*. IEEE, 2014, pp. 1078–1085.

[14] R. Soltani, D. Goeckel, D. Towsley, B. Bash, and S. Guha, "Covert wireless communication with artificial noise generation," *IEEE Transactions on Wireless Communications*, pp. 1–1, 2018.

[15] R. Soltani, D. Goeckel, D. Towsley, and A. Houmansadr, "Covert communications on poisson packet channels," in *Communication, Control, and Computing (Allerton), 2015 53rd Annual Allerton Conference on*.   IEEE, 2015, pp. 1046–1052.

[16] ——, "Covert communications on renewal packet channels," in *Communication, Control, and Computing (Allerton), 2016 54th Annual Allerton Conference on*.   IEEE, 2016, pp. 548–555.

[17] I. Dinur and K. Nissim, "Revealing information while preserving privacy," in *Proceedings of the twenty-second ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*.   ACM, 2003, pp. 202–210.

[18] A. Blum, K. Ligett, and A. Roth, "A learning theory approach to noninteractive database privacy," *Journal of the ACM (JACM)*, vol. 60, no. 2, p. 12, 2013.

[19] Y. Xiao, L. Xiong, and C. Yuan, "Differentially private data release through multidimensional partitioning," in *Secure Data Management*.   Springer, 2010, pp. 150–168.