

Differentially Private Publication of Electricity Time Series Data in Smart Grids

Sina Shaham
University of Southern California
Los Angeles, USA
sshaham@usc.edu

Bhaskar Krishnamachari
University of Southern California
Los Angeles, USA
bkrishna@usc.edu

Gabriel Ghinita
Hamad Bin Khalifa University
Ar-Rayyan, Qatar
gghinita@hbku.edu.qa

Cyrus Shahabi
University of Southern California
Los Angeles, USA
shahabi@usc.edu

ABSTRACT

Smart grids are a valuable data source to study consumer behavior and guide energy policy decisions. In particular, time-series of power consumption over geographical areas are essential in deciding the optimal placement of expensive resources (e.g., transformers, storage elements) and their activation schedules. However, publication of such data raises significant privacy issues, as it may reveal sensitive details about personal habits and lifestyles. Differential privacy (DP) is well-suited for sanitization of individual data, but current DP techniques for time series lead to significant loss in utility, due to the existence of temporal correlation between data readings. We introduce *STPT* (*Spatio-Temporal Private Timeseries*), a novel method for DP-compliant publication of electricity consumption data that analyzes spatio-temporal attributes and captures both micro and macro patterns by leveraging RNNs. Additionally, it employs a partitioning method for releasing electricity consumption time series based on identified patterns. We demonstrate through extensive experiments, on both real-world and synthetic datasets, that STPT significantly outperforms existing benchmarks, providing a well-balanced trade-off between data utility and user privacy.

KEYWORDS

Differential Privacy, Time Series, Smart Grids

1 INTRODUCTION

Analysis of electricity consumption data plays a critical role in planning power grid infrastructures for smart cities. Such data are represented as time series of geo-tagged data, and serve as a critical information source, providing detailed knowledge to policymakers about energy usage trends. Such data can help identify where/when consumption surges occur, and decide where to place expensive equipment such as transformers; or in the case of renewable energy

sources, it can help decide where to place storage elements, and when to release their capacity back into the grid.

Despite its advantages, analysis of electricity consumption time series raises significant privacy concerns. The data may reveal personal habits and lifestyles, such as individuals' daily routines, working hours, etc., leading to privacy violations. Moreover, the risk of third-party exploitation by marketers and advertisers poses a threat of unwanted privacy intrusions, as consumers may be targeted based on their specific energy usage behavior.

Prevailing approaches for protecting electricity time series information rely on the powerful Differential Privacy (DP) model [1]. DP achieves privacy by adding noise to the data, thereby minimizing the likelihood of re-identification. However, when dealing with time series data, the existence of temporal correlations leads to increased re-identification risk over time, causing DP to add excessive noise to offset this risk, lowering data utility [2, 3].

We propose a model that jointly takes into account both time and space attributes of electricity consumption data. Our *Spatio-Temporal Private Timeseries* (STPT) algorithm trains a Recurrent Neural Network (RNN) to identify spatio-temporal electricity consumption patterns. The patterns are subsequently used to partition the time series into a spatio-temporal histogram that is used by DP mechanisms to sanitize and release the data. A key innovation of STPT is the incorporation of spatial distribution alongside temporal sequencing. We start with a low-granularity aggregation of time series data to identify macro consumption trends, followed by several increasingly-higher granularity aggregations to discern micro trends. This dual-focus on both macro and micro trends allows for a nuanced representation of consumption patterns, enhancing STPT's ability to preserve data utility while enforcing DP. Note that, while RNNs have been used before for geo-tagged time series, the focus of prior work is on trajectory forecasting [4]. Our approach is significantly different as consumer locations are static (e.g., households), and the purpose of RNN is to estimate future power consumption at each location.

Our specific contributions include:

- We introduce a novel method for modeling and representing electricity time series data, which takes into account both spatial and temporal properties.
- We propose STPT, an innovative algorithm that integrates a unique approach for training RNNs across both time and space dimensions on differentially private data.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

EDBT, March 25–28, 2025, Barcelona

© 2018 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-XXXX-X/18/06
<https://doi.org/XXXXXXX.XXXXXXX>

- We design a customized technique for STPT that clusters electricity consumption data across time and space, thereby improving data utility when applying DP-compliant mechanisms for sanitization of time series.
- We perform an extensive experimental evaluation on both real-world and synthetic datasets, demonstrating STPT’s notable improvements in data utility compared to existing benchmarks¹.

Section 2 introduces foundational concepts and the system architecture. Section 3 outlines our method for electricity data modeling, following by the description of the STPT algorithm in Section 4. We provide an extensive experimental evaluation in comparison with several benchmarks in Section 5. Section 6 reviews related work. Section 7 concludes with future work directions. Proofs of theorems are provided in the appendix.

2 PRELIMINARIES

Consider a two-dimensional map that encloses a set of N households $\mathcal{U} = \{u_1, \dots, u_N\}$. We denote the electricity consumption for user i at time t by $x_{i,t}$ (we use the term household and power grid user interchangeably). Each household meter sends its electricity reading to an aggregator at regular intervals $\Delta \times t$ ($t = 1, \dots, T$) where $\Delta \in \mathbb{R}$. The dataset of meter readings is denoted as:

$$\mathcal{D} = (x_{i,t})_{i=1,\dots,N;t=1,\dots,T} \quad (1)$$

The goal is to release the dataset \mathcal{D} according to the requirements of DP, thus preventing an adversary from inferring the consumption patterns of any individual user. We start our discussion by explaining the system model commonly used for the publication of DP electricity consumption time series, followed by an illustration of the foundational concepts related to DP. A summary of notations used throughout the paper is provided in Appendix A.

2.1 System Model

Figure 1 depicts the system architecture which adheres to the industry-standard model for publishing electricity data, and consists of households, a data aggregator, and data recipients which perform analyses on the released consumption data. The specific functions of each party are detailed below.

- **Households** equipped with smart meters are generators of data and are considered to be trustworthy in the system model. The electricity consumption of users is recorded hourly using their meter and sent to the data aggregator.
- **Data Aggregator** is a trusted party that collects the time series generated by users and publishes their aggregated data in a privacy-preserving way. The sanitization process is done based on DP, preventing adversaries from inferring any individual-level consumption pattern.
- **Data Recipients** leverage the private aggregated data for diverse applications, from forecasting to planning. Their objective is to utilize consumption values over specific spatial regions and time periods. The recipients are considered to be honest but curious, and they may attempt to infer individual user consumption from aggregated data. Individual

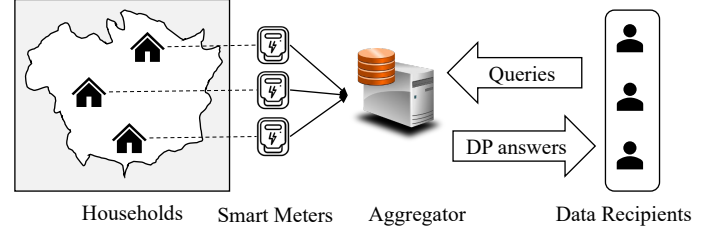


Figure 1: System model

consumption details must be protected, as they can be used to infer sensitive details about users, such as activity patterns, lifestyle habits, etc.

2.2 Differential Privacy

Two databases \mathcal{D} and \mathcal{D}' are called *neighboring* or *sibling* if they differ in a single record t , i.e., $\mathcal{D}' = \mathcal{D} \cup \{t\}$ or $\mathcal{D}' = \mathcal{D} \setminus \{t\}$.

Definition 1 (ϵ -Differential Privacy[1]). A randomized mechanism \mathcal{A} provides ϵ -DP if for any pair of neighbor datasets \mathcal{D} and \mathcal{D}' , and any $a \in \text{Range}(\mathcal{A})$,

$$\frac{\Pr(\mathcal{A}(\mathcal{D}) = a)}{\Pr(\mathcal{A}(\mathcal{D}') = a)} \leq e^\epsilon \quad (2)$$

Parameter ϵ is referred to as privacy budget. ϵ -DP requires that the output obtained by executing mechanism \mathcal{A} does not significantly change by adding or removing one record in the database. Thus, an adversary is not able to infer with significant probability whether an individual’s record was included or not in the database.

Aside from the amount of privacy budget, another factor that plays a critical role in achieving ϵ -DP is the concept of *sensitivity*, which captures the maximal difference achieved in the output by adding or removing a single record from the database.

Definition 2 (L_1 -Sensitivity[5]). Given sibling datasets $\mathcal{D}, \mathcal{D}'$ the L_1 -sensitivity of a set $g = \{g_1, \dots, g_m\}$ of real-valued functions is:

$$s = \max_{\mathcal{D}, \mathcal{D}'} \sum_{i=1}^m |g_i(\mathcal{D}) - g_i(\mathcal{D}')| \quad (3)$$

A widely-used mechanism to achieve ϵ -DP is called Laplace mechanism. This approach adds to the output of every query function noise drawn from Laplace distribution $\text{Lap}(b)$ with scale b and mean 0, where b depends on sensitivity and privacy budget.

$$\text{Lap}(x|b) = \frac{1}{2b} e^{-|x|/b} \text{ where } b = \frac{s}{\epsilon} \quad (4)$$

To simplify notation, we denote Laplace noise by $\text{Lap}(\frac{s}{\epsilon})$.

In our work, we make extensive use of the following three essential results in differential privacy:

THEOREM 1 (SEQUENTIAL COMPOSITION [6]). *Let A_1 and A_2 be two DP mechanisms that provide ϵ_1 - and ϵ_2 -differential privacy, respectively. Then, applying in sequence A_1 and A_2 over the dataset \mathcal{D} achieves $(\epsilon_1 + \epsilon_2)$ -differential privacy.*

¹Codes and datasets are publicly available online at the following link: <https://github.com/ANRGUSC/pars/>

THEOREM 2 (PARALLEL COMPOSITION [6]). Let A_1 and A_2 be two DP mechanisms that provide ϵ_1 - and ϵ_2 -differential privacy, respectively. Then, applying A_1 and A_2 over two disjoint partitions of the dataset \mathcal{D}_1 and \mathcal{D}_2 achieves $(\max(\epsilon_1, \epsilon_2))$ -differential privacy.

THEOREM 3 (POST-PROCESSING IMMUNITY[1]). Let A be an ϵ -differentially private mechanism and g be an arbitrary mapping from the set of possible output sequences O to an arbitrary set. Then, $g \circ A$ is ϵ -differentially private.

3 PROBLEM STATEMENT

3.1 Time-Series Representation

Consider a spatial grid of size $C_x \times C_y$ overlaid on a 2D map, dividing the spatial domain into smaller regions. Additionally, we divide the time dimension into a number of C_t equal-length intervals. The electricity consumption data is thus captured by a three-dimensional matrix C_{cons} called consumption matrix with $C_x \times C_y \times C_t$ elements. Each element c_{ijk} in this matrix represents the electricity consumption within the (i, j) region during the time interval from $\Delta \times k$ to $\Delta \times (k + 1)$, where Δ is the time resolution. For ease of analysis, especially when conducting sensitivity studies in relation to data publication under DP, we assume without loss of generality that $\Delta = 1$. This assumption implies that each data point in the time series corresponds to distinct time intervals, meaning that C_t is effectively the length of the time series ($C_t = T$).

3.2 Problem Formulation

Data recipients are interested in answering multi-dimensional *range queries* on top of the electricity consumption matrix.

Definition 3. (Range Query) A range query on the consumption matrix is a 3-orthotope with dimensions denoted as $d_1 \times d_2 \times d_3$, where d_i represents a continuous interval in dimension i .

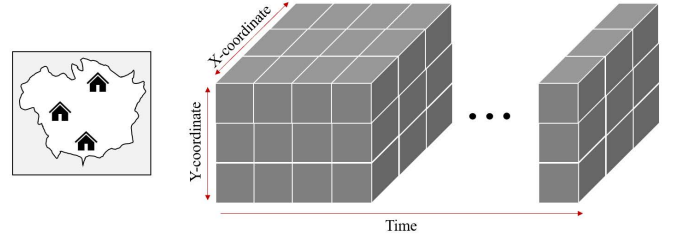
To evaluate the accuracy of results, we use the *Mean Relative Error (MRE)* metric. For a query q with the true aggregated consumption p and noisy consumption value \bar{p} , MRE is calculated as

$$MRE(q) = \frac{|p - \bar{p}|}{p} \times 100 \quad (5)$$

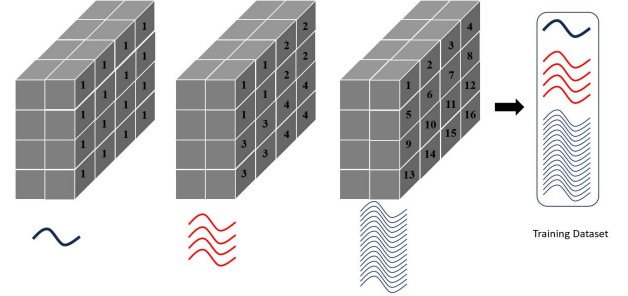
PROBLEM 1. Given a consumption matrix denoted by C_{cons} , generate a ϵ -DP matrix $C_{\text{sanitized}}$ such that average MRE subject to range queries is minimized.

3.3 A Simple Strategy

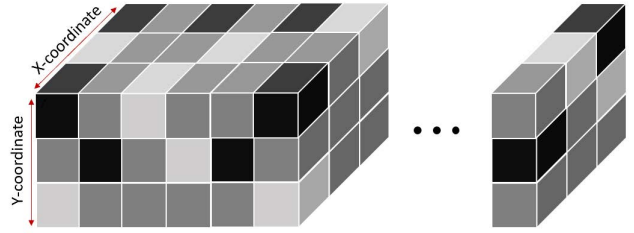
One simple strategy to publish the electricity consumption matrix is the *Identity* algorithm [7]. This algorithm was initially designed for population histograms, and works by adding independent Laplace noise to every matrix cell. When applying this technique to the consumption matrix, it is essential to note that time series have temporal correlations. As a result, every snapshot of time should have its distinct allocated privacy budget, according to the sequential composition theorem (Theorem 1). Conversely, since at each timestamp the spatial grid creates disjoint partitions of the map, parallel composition applies within each time interval (Theorem 2). The following important result emerges which quantifies the sensitivity of a query on each cell of the electricity consumption matrix.



(a) Electricity consumption matrix.



(b) Generated time series for training the RNN unit.



(c) Clustering hypercube cells for sanitization purposes.

Figure 2: Consumption matrix in different stages.

THEOREM 4. The sensitivity of range queries of size $1 \times 1 \times 1$ on the electricity consumption matrix C_{cons} is given by $\max_{i,t} x_{i,t}$.

THEOREM 5. The consumption matrix follows sequential decomposition in time and parallel decomposition in space.

The Identity algorithm allocates an equal amount of privacy budget to each time slice. Therefore, if ϵ_{tot} represents the entire budget allocated for sanitization, the budget for each time slice amounts to $\epsilon_{\text{tot}}/C_t$. Then, each cell of the matrix is sanitized by the addition of Laplace noise with sensitivity 1 and privacy budget $\epsilon_{\text{tot}}/C_t$ given that the time series are normalized in advance.

4 STPT ALGORITHM

4.1 Overview

The STPT algorithm starts by generating two matrices C_{cons} and C_{norm} out of the collected time series from different neighborhoods of the map. C_{cons} denotes the consumption matrix based on the actual values of the time series, whereas C_{norm} is its normalized counterpart. We employ min-max normalization at a global level.

Under this scheme, the normalized consumption for user i at time j is computed as follows:

$$x_{i,j} := \frac{x_{i,j} - \min_{i,t} x_{i,t}}{\max_{i,t} x_{i,t} - \min_{i,t} x_{i,t}} \quad (6)$$

The STPT algorithm conducts two sequential core procedures to generate the DP consumption matrix, namely the Pattern Recognition Step, followed by the Sanitization Step. The workflow of the approach is shown in Figure 3.

Pattern Recognition Step. The primary aim of pattern recognition is to create a sanitized version C_{pattern} of the normalized consumption matrix C_{norm} . The core objective is to achieve rough estimations of the normalized time series values while utilizing a small amount of privacy budget. The choice of using C_{norm} over C_{cons} is strategic, as it helps in bounding the sensitivity of the cells during the sanitization process. The generation of sanitized estimated values in C_{pattern} involves using a short segment of the time series, T_{train} , to predict future consumption while preserving privacy. The training data are sanitized through a novel hierarchical method, considering both time and space dimensions. The sanitized data are used to train a RNN, which is responsible for estimating the remaining values in C_{pattern} . The total privacy budget allocated for this phase is denoted by $\epsilon_{\text{pattern}}$.

Sanitization Step. This algorithm's primary objective is to perform an intelligent partitioning of the matrix, based on the private estimates in C_{pattern} , and then to sanitize and release the values of C_{cons} . Since the estimates in C_{pattern} are private, the resulting matrix partitioning is also privacy-preserving, being derived from private data. The partitioning approach for the consumption matrix, both temporally and spatially, is predicated on the principle of homogeneity. This principle, which contributes to enhanced data utility, aims to group cells with similar values into the same partition. Post partitioning, the true values in each partition, extracted from C_{cons} , are aggregated and sanitized. The final output of this procedure is the matrix $C_{\text{sanitized}}$, representing the differentially private version of C_{cons} . The privacy budget for the sanitization algorithm is denoted as $\epsilon_{\text{sanitize}}$. This leads to the total amount of privacy budget of ϵ_{tot} for the STPT algorithm where,

$$\epsilon_{\text{tot}} = \epsilon_{\text{sanitize}} + \epsilon_{\text{pattern}} \quad (7)$$

Therefore, STPT publishes a ϵ_{tot} -differential private version of the original consumption matrix (C_{cons}). The pseudocode for STPT is provided in Algorithm 1.

4.2 Pattern Recognition

The goal of the pattern recognition phase in the STPT algorithm is to effectively use a designated privacy budget, $\epsilon_{\text{pattern}}$, to develop a method for privately generating approximate estimates for cells within the normalized consumption matrix. The primary means to accomplish this is through the private training of an RNN unit. The input comprises of household time series data along with their corresponding geographic locations on the map. The pattern recognition process is outlined in Section 3. It involves generating the consumption matrix C_{cons} from the time series data and creating C_{norm} , the consumption matrix based on normalized time series.

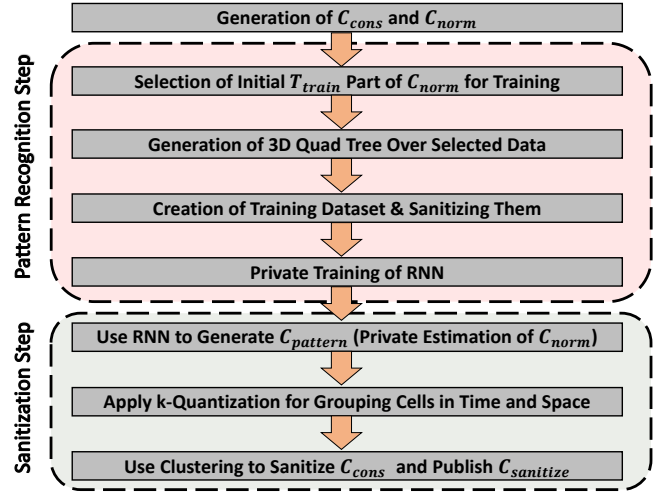


Figure 3: Workflow of Proposed Approach

Next, the initial time segment (T_{train}) of the consumption matrix C_{norm} is allocated for training, resulting in a matrix dimensionality of $C_x \times C_y \times C_t[0 : T_{\text{train}}]$. The notation $C_t[0 : T_{\text{train}}]$ indicates the selection of indices from 0 to T_{train} on the time dimension. A critical challenge is determining an efficient training method for the RNN, ensuring it comprehensively learns both micro and macro trends across neighborhoods, while minimizing the amount of privacy budget utilized for training. One straightforward training method for the RNN model involves the sanitization strategy described in Section 3.3. By adopting this method, every time snapshot is allocated a budget of $\epsilon_{\text{sanitize}}/T_{\text{train}}$, and each matrix entry undergoes Laplace noise perturbation with a sensitivity of one and budget $\epsilon_{\text{sanitize}}/T_{\text{train}}$, which translates to $\text{Lap}(1/(\epsilon_{\text{sanitize}}/T_{\text{train}}))$ given that time series are normalized.

Despite its feasibility, this method introduces excessive noise into the training data, impacting model accuracy. We introduce an approach centered on the generation of a spatio-temporal quadtree (lines 5 to 12 in Algorithm 1). Assuming $C_x < C_y$, the process initiates by segmenting time into $\log_2(C_x) + 1$ levels, resulting in a time span of T'_{train} for each interval, derived as follows:

$$T'_{\text{train}} = \lceil T_{\text{train}} / (\log_2(C_x) + 1) \rceil \quad (8)$$

The matrix corresponding to the i^{th} interval is $C_x \times C_y \times C_z[i * T'_{\text{train}} : (i + 1) * T'_{\text{train}}]$, corresponding to the quadtree's i^{th} level. In the first segment of the matrix corresponding to the tree's root, all cells are presumed to be part of a unified neighborhood. However, in the subsequent sub-matrix (depth 1), the previous matrix's neighborhoods are subdivided into four distinct quadrants. Given that quadtrees are data-independent index structures, we do not need to expend privacy budget to determine the division points of the space. Once the spatio-temporal partitioning of the training matrix is completed, there exist 4^i neighborhoods at each level i . The next step of the algorithm is generating a single time series representing each neighborhood. The representative time series is generated by element-wise averaging of all time series in the neighborhood

over the time allocated for that level of the tree. Consider a neighborhood at the i^{th} level of the tree, and without loss of generality, suppose times series 1, ..., j fall in this neighborhood. For each point in time t lying in the interval $i \cdot T'_{\text{train}} : (i+1) \cdot T'_{\text{train}}$ the value in the representative time series is the average of all consumption of users in that neighborhood and that specific time, calculated as:

$$x_{\text{rep},t} = \frac{1}{j} \sum_{i=1}^j x_{i,t} \quad (9)$$

We emphasize that the time series created are stacked, and not sequential. To produce training data for subsequent training phases, a time window is swept across each time series individually to generate the training data. An illustration of the method is exemplified in Figure 2b. As observed, we utilize a $4 \times 4 \times 6$ matrix for the training process. The entire duration of training is segmented into 3 parts, which translates to a duration of $6/(\log_2(4) + 1)$ for each part. This involves the creation of 3 submatrices, each having dimensions of $4 \times 4 \times 2$. The root node of the quadtree includes only one neighborhood, indicated by the cells' number, which results in one distinct time series shown beneath. Depths 1 and 2 in the tree align with 4 and 16 time series, respectively. Altogether, 21 time series are employed for the next step.

Once partitioning is complete, the 4^i resulting time series at each level i undergo sanitization. The key advantage of hierarchical partitioning lies in bounding sensitivity. As outlined in Theorem 6, the sensitivity of the time series at depth i is $1/4^{\log_2(C_x)-i}$. The underlying principle suggests that macro trends can be captured with heightened precision since the sensitivity of the time series is reduced, allowing for a smaller amount of Laplace noise during sanitization. The stacked sanitized time series corresponds to list res in Algorithm 1. After finalizing the time series, training data for the RNN is produced by sweeping a time window across the time series and organizing them into batches. This training data is then employed for training an RNN. Subsequently, the RNN is utilized to create private estimations of the matrix C_{norm} in C_{pattern} .

THEOREM 6. *The sensitivity of a cell at depth i of the matrix is $1/4^{\log_2(C_x)-i}$.*

4.3 Sanitization Algorithm

The output of pattern recognition is the matrix C_{pattern} , with dimensions $C_x \times C_y \times C_t$. Each element of this matrix is created using a differentially private approach. These elements are sanitized estimates of normalized time series, providing an idea of consumption patterns rather than actual consumption amounts. The purpose of the sanitization algorithm is not only to reveal these patterns but also to provide sanitized consumption values.

The sanitization algorithm of STPT (lines 15 to 22 in Algorithm 1) starts by developing a non-overlapping partitioning of the matrix C_{pattern} . The developed partitioning's objective is to group cells with similar values together. For this purpose, we use a k -quantization of matrix C_{pattern} to generate clusters. The formal definition of k -quantization is provided in Definition 4.

Definition 4 (k -Quantization). Let C_{pattern} be a 3-dimensional matrix with elements $c_{i,j,k}$, where i, j, k are the indices of the matrix,

Algorithm 1 STPT

Input: \mathcal{D} , $\epsilon_{\text{pattern}}$, $\epsilon_{\text{sanitize}}$, Quad Tree Depth ($depth$), Window Size (ws), Training Time (T_{train}), Quantization Level (k).

Output: Sanitized Consumption Matrix

- 1: $C_{\text{cons}} \leftarrow$ Create Consumption Matrix
- 2: $C_{\text{norm}} \leftarrow$ Min-Max Normalize C_{cons}
- 3: Select T_{train} Data Points from C_{norm}
- 4: $res \leftarrow []$ ▷ Initialize empty list for time series
- 5: **for** $d \in [0, \dots, depth]$ **do**
- 6: $Temp \leftarrow$ Select time interval $[i \cdot T_{\text{train}} : (i+1) \cdot T_{\text{train}}]$
- 7: Divide x and y Axes of $Temp$ into 2^d Creating 4^d Neighborhoods
- 8: **for** each neighborhood **do**
- 9: Compute Representative Time Series (Eq. 9)
- 10: Sanitize Time Series with Budget $\epsilon_{\text{pattern}}/T_{\text{train}}$ and Sensitivity $1/4^{\log_2(C_x)-i}$
- 11: Append Sanitized Series to res
- 12: Prepare Training Data from res Based on ws
- 13: Train RNN
- 14: Generate C_{pattern} Using RNN
- 15: $\mathcal{P} \leftarrow k$ -Quantize C_{pattern}
- 16: **for** each partition $P_i \in \mathcal{P}$ **do**
- 17: $f(P_i) \leftarrow$ Sum Values in C_{cons} for P_i
- 18: $s \leftarrow$ Compute Sensitivity of P_i
- 19: Sanitize $f(P_i)$ Using s and Budget (Eq. 11)
- 20: **for** each cell $c \in P_i$ **do**
- 21: Update c in C_{cons} to $f(P_i)/|P_i|$
- 22: **return** Sanitized C_{cons} , i.e., $C_{\text{sanitized}}$

and let k be a positive integer representing the number of quantization levels. The k -quantization of C_{pattern} is a process defined as follows:

- (1) **Determine Range:** Identify the minimum $\min(C_{\text{pattern}})$ and maximum $\max(C_{\text{pattern}})$ values within the matrix C_{pattern} .
- (2) **Establish Quantization Buckets:** Divide the range

$$[\min(C_{\text{pattern}}), \max(C_{\text{pattern}})]$$

into k equal intervals or 'buckets', each representing a quantization level.

- (3) **Quantize Matrix Values:** For each element $c_{i,j,k}$ in the matrix C_{pattern} , assign it to a quantization level based on which bucket its value falls into. This assignment is represented as a function $Q(c_{i,j,k})$ that maps the value of $c_{i,j,k}$ to one of the k quantization levels.

The output is a quantized 3-dimensional matrix where each element is represented by one of the k quantization levels, effectively reducing the original range of values in C_{pattern} to k distinct values.

The k -Quantization of the matrix leads to generation of k non-overlapping clusters of C_{pattern} and subsequently C_{cons} . We use this non-overlapping partitioning of the matrix as a basis for sanitizing and releasing the electricity data C_{cons} . Once partitioning is completed, the values in each partition are aggregated and sanitized based on the Laplace mechanism. The accumulated value in each partition is then uniformly distributed across its corresponding

cells. More formally, denote the set of generated non-overlapping partitions by $\mathcal{P} = \{P_1, P_2, \dots, P_k\}$ where each P_i is a set of cells. Note that, partitions are generated based on C_{pattern} which is safe to release. The partitions are then used for matrix C_{cons} , i.e. to compute the sanitized consumption values. A partition's cells are not necessarily continuous and may be scattered across the matrix. To sanitize and publish the electricity consumption values, the corresponding values in each partition are added and sanitized based on Laplace noise to achieve differential privacy, as follows:

$$f(P_i) = \sum_{c \in P_i} f(c) + \text{Lap}(s/\epsilon), \quad (10)$$

where c denotes a cell and the function $f(\cdot)$ returns the added value of all cells in the partition. Once the sanitized value of each partition is generated, it is uniformly distributed among its cells. Therefore, for all $c \in P_i$ its value is updated to $f(P_i)/|P_i|$ in the sanitized matrix $C_{\text{sanitized}}$.

A critical aspect is the allocation of privacy budget across quadtree levels. Theorem 7 establishes that the sensitivity of each partition is equal to the maximum number of cells contained within a single xy -axis *pillar* of the consumption matrix, where a pillar refers to all cells that have the same x and y coordinates. This theorem provides a foundational understanding of how sensitivity is distributed across the partitions, and guides the privacy budget allocation.

THEOREM 7. *Let $P_i \in \mathcal{P}$ be a partition in the consumption matrix. The sensitivity of P_i is the maximum number of cells it contains in any of the xy -axis pillars.*

Armed with this knowledge, the optimal assignment of privacy budget to each partition can be derived as follows. Let us denote the sensitivity of partition P_i and allocated budget to this partition by s_i and ϵ_i , respectively. The optimal assignment of privacy budget to partitions can be formulated and solved by convex optimization as shown in Theorem 8.

THEOREM 8. *Given a non-overlapping partitioning of the consumption matrix $\mathcal{P} = \{P_1, \dots, P_m\}$ and the sensitivity of these partitions $S = \{s_1, \dots, s_m\}$, the optimal allocation of the privacy budget to a partition P_i is derived by the following equation:*

$$\epsilon_i = \frac{\epsilon_{\text{sanitize}} \times s_i^{\frac{2}{3}}}{\sum_{i=1}^m s_i^{\frac{2}{3}}}, \quad (11)$$

where $\epsilon_{\text{sanitize}}$ represents the total sanitization budget.

5 EXPERIMENTAL EVALUATION

5.1 Experimental Setup

Datasets & Spatial Distribution. We conducted our experiments using four publicly accessible datasets, each under two distinct spatial distributions, resulting in a total of eight datasets. The statistics of these datasets are illustrated in Figure 6 and detailed in Table 2 of the Appendix C.

- **CER [8]:** The dataset released by the Commission for Energy Regulation (CER) in Ireland originates from the Electricity Smart Metering Customer Behavior Trials carried out between 2009 and 2010. This project involved over 5,000 households and businesses and was focused on assessing

the impact of smart meters on electricity consumption patterns. The objective was to gain insights for conducting a cost-benefit analysis regarding the country-wide adoption of smart meters. The anonymized data collected from these trials has been made accessible online for public research purposes.

- **California, Michigan, and Texas Datasets [9]:** The datasets serve as digital twins representing residential energy usage within each state's residential sector. They are identified by the state's acronym and concentrate on the electricity consumption of the first five counties in alphabetical order for each state. For instance, the CA dataset includes data from Alameda, Alpine, Amador Butte, and Calaveras counties. These datasets provide hourly household electricity time series data from September to December 2014.

The privacy concerns regarding household-level electricity consumption have limited the availability of publicly accessible datasets, with no geotagged datasets currently accessible online [10]. Therefore, to account for the distribution of users, we perform our experiments by distributing households in two settings: Uniform and Normal. A grid with granularity of 32×32 is overlaid on the map, and the households are distributed over the space according to one of the two distributions. The center of the normal distribution is selected randomly over the map, and the households are located with the standard deviation equal to one-third of the grid size. The experiment is repeated 10 times and the average result is shown to ensure repeatability of the experiments. Therefore, in total, the experiments are conducted on four datasets which are referred to as CA-Uniform, MI-Uniform, TX-Uniform, CER-Uniform, CA-Normal, MI-Normal, TX-Normal, and CER-Normal.

Benchmarks. We compare the performance of our approach with the available state-of-the-art approaches detailed below.

- **FAST.** The framework proposed in [2] is a widely adopted approach focused on exploiting the Kalman Filter for lowering utility loss while sanitizing time series.
- **Fourier Perturbation Algorithm.** The methodology initially introduced in [11] and subsequently refined through sensitivity evaluations in [3], involves processing a time series with a specified integer k . The procedure begins by executing a Fourier transform on the time series, followed by the selection and sanitizing of the top k primary Fourier coefficients. After sanitizing these coefficients, the inverse Fourier transform is applied, and DP time series are generated. We implement the algorithm in two settings where $k = 10$ and $k = 20$, denoted in the experiments as Fourier-10 and Fourier-20, respectively.
- **Wavelet Perturbation Algorithm.** By substituting the Fourier transform with the discrete Haar wavelet transform, Lyu et al. [12] introduced the wavelet perturbation algorithm for creating DP time series. This method, akin to the Fourier technique, requires an integer k , which signifies the number of coefficients to be used and sanitized. We denote this algorithm as Wavelet and implement it in two distinct scenarios: one with $k = 10$ and the other with $k = 20$.

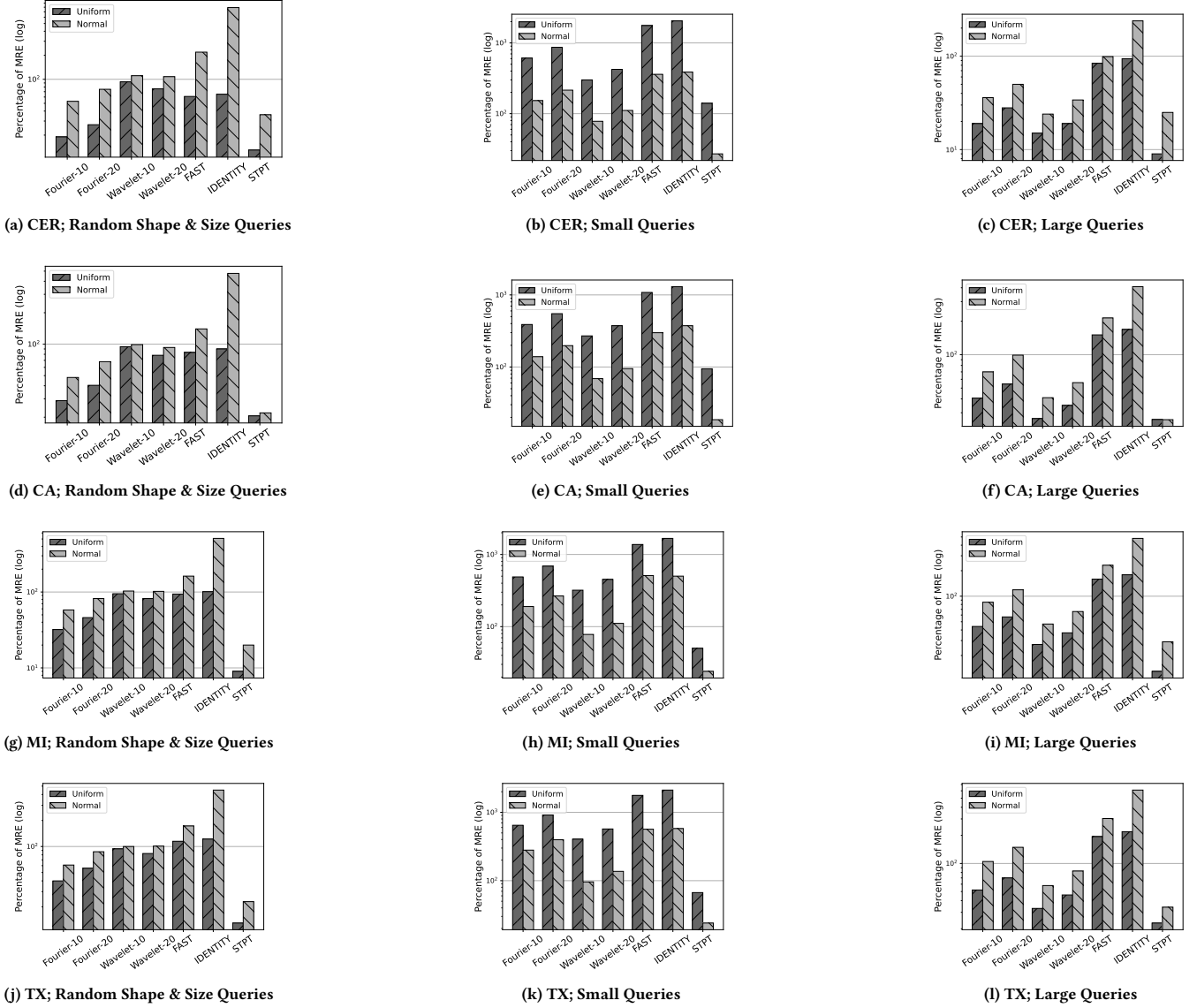


Figure 4: Assessment of Algorithm Performance Across Datasets and Query Types.

- *Identity*. The approach illustrated in Subsection 3.3 is an adaptation of the original approach for the publication of time series and will be used as a comparison benchmark in our experiments.

Query Types. As discussed in the problem formulation of Section 3.2, analysts are interested in range queries which are 3-orthotopes with dimensions $d_1 \times d_2 \times d_3$, indicating the consumption on a map region over a particular time range. For this purpose, we use small ($1 \times 1 \times 1$) and large queries ($10 \times 10 \times 10$) as well as queries with random shape and size. For each of the three categories, we generated 300 randomly generated queries over the consumption matrix, calculated the MRE, and reported the average result.

Hyper-parameters Setting. Provided in Appendix D.
Hardware and Software Setup. Provided in Appendix E.

5.2 Comparison with Benchmarks

Figure 4 illustrates the performance of algorithms when subjected to queries of differing shapes and sizes. Each row in the figure is dedicated to one of four datasets: CA-Uniform, CER-Uniform, CA-Normal, and CER-Normal. Within each row, the leftmost figure depicts the performance for randomly shaped and sized queries generated over the consumption matrix. The center figure shows results for smaller queries, and the rightmost figure displays the

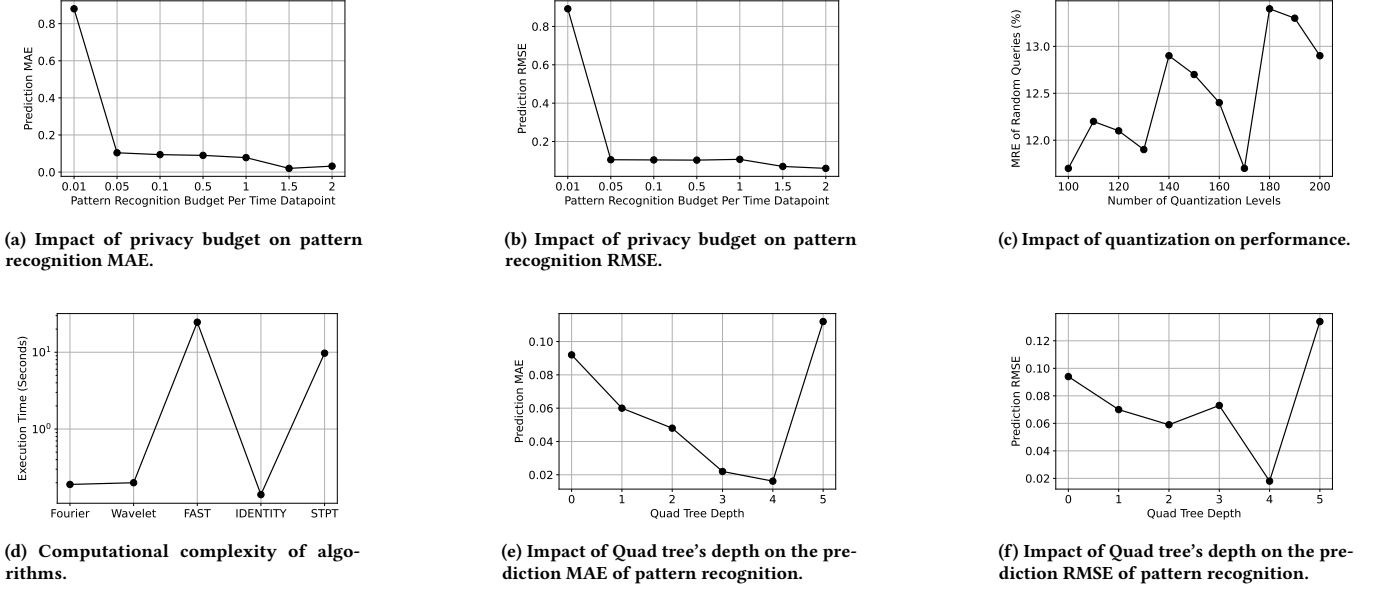


Figure 5: Detailed Analysis of STPT.

performance for larger queries. As can be seen, significant improvements have been made by STPT across the datasets in either distribution. As an example, for queries with random shapes and sizes, the STPT algorithm exhibited percentage-wise improvements of 60, 31, 54, and 32 in the Uniform setting for each respective dataset. Notably, the performance enhancement of the algorithms is more pronounced for smaller-sized queries. This result is desirable, indicating that more precise information about the consumption matrix can be conveyed with minimal loss of utility.

As anticipated, the IDENTITY algorithm generally shows the least accuracy among the baseline algorithms. However, it surpasses some of the more recent algorithms in scenarios where the data exhibit a more uniform shape, as seen in the first and second rows. An unexpected outcome of our experiments is the relative performance of Wavelet and Fourier transformations. Although Wavelet transformation was introduced at a later stage than the Fourier approach, the Fourier method demonstrates superior performance for queries of random shape and size.

Another notable observation is that, on average, all algorithms tend to perform worse with non-uniform data. This aligns with findings in [13], where a crucial determinant of performance is the homogeneity in data partitioning. Uniform data distribution contributes to higher homogeneity, and also decreases the uniformity error when estimating the size of random queries based on the sanitized partition counts.

5.3 STPT Detailed Evaluation

Privacy Budget. Figures 5a and 5b analyze how the allocation of privacy budget affects pattern recognition performance in the STPT algorithm. While the sanitization budget in the second step remains constant, the budget for pattern recognition varies. For enhanced clarity, the x-axis displays the amount of budget allocated to each

training datapoint of the RNN unit. The y-axis, meanwhile, indicates the MAE and RMSE of the RNN unit's predictions. As anticipated, an increase in the allocated budget enhances prediction accuracy, showcasing the privacy-utility trade-off. Notably, a significant improvement is observed when the privacy budget is increased from 0.01 to 0.05, suggesting that the minimal budget required for effective training lies within this range.

Quantization. Figure 5c illustrates the effect of the number of quantization levels on the performance of the STPT algorithm. The MRE metric is displayed on the y-axis for queries of varying shapes and sizes. Although there are fluctuations in the results, the general trend indicates that excessive increase in the number of quantization levels can negatively impact the effectiveness of STPT. This is expected, as many points in the cycle of a time series often exhibit similar values. Consequently, a high degree of quantization results in excessive partitioning and a reduction in the homogeneity that is captured in the data.

Computational Complexity. Figure 5d presents and compares the runtime of various algorithms. According to the figure, the execution time for all algorithms is remarkably small, typically spanning just a few seconds. Although the STPT algorithm shows a slight rise in computational complexity, it is crucial to note that a significant portion of this complexity stems from the initial training phase required for pattern recognition, which is a one-time process. Overall, all algorithms demonstrate comparable execution times in the order of seconds, indicating that computational complexity does not pose a significant hurdle to their performance.

Quad Tree Depth. The influence of varying tree depth on pattern recognition efficiency is showcased in Figures 5e and 5f. The aim is to explore how changes in tree depth affect the MAE and RMSE in the RNN unit. It's important to remember the balance between sensitivity and precision in time series produced at various depths.

At shallow depths, such as the root node, the noise effect on relative error is low, since real counts are high. As the depth increases, this trend inverts. The results reveal that augmenting the tree depth up to a certain point enhances performance, but beyond that, the diminishing number of training data points at each level restricts further performance gains. Consequently, opting for a medium tree length, despite its impact on micro trends, proves to be advantageous.

6 RELATED WORK

Private Publication of Time Series. The existing body of work on differentially-private publication of time series falls into two primary categories: data transformation and correlation analysis. In the former category, the main strategy involves converting the data into an alternative domain that exhibits lower sensitivity, or provides a condensed representation of the time series. After sanitization in this new domain, an inverse function is used to revert the data back to its original form for publication. Notable methods in this category include the Fourier transformation [3, 11] and the discrete Haar wavelet transform [12]. The latter category focuses on enhancing the utility of DP time series publications through improved leverage of inter-data correlations. This includes the concept of Pufferfish privacy, which employs a Bayesian Network to model correlations [14]; the use of Kalman Filters to reduce utility loss as explored in [2]; and the adoption of a first-order auto-regressive process for correlation modeling as presented in [15].

Private Publication of Multi-Dimensional Histograms. Our research closely aligns with the topic of publishing higher-dimensional histograms under DP. The study in [13] highlights the importance of data homogeneity in the private publication of histograms and introduces an algorithm called HTF (Homogeneous Tree Framework), designed to capture data homogeneity in order to reduce the effect of DP noise and thus improve utility. Another algorithm in this category is HDMM (High-Dimensional Matrix Mechanism) [16], which conceptualizes queries and data as vectors, and employs advanced optimization and inference methods for their resolution. DPCube [17] focuses on identifying and privately releasing dense sub-cubes. It allocates a portion of the privacy budget to derive noisy counts over a regular partitioning, which is subsequently refined into a standard kd-tree structure. The method then uses the remaining budget to acquire fresh noisy counts for the partitions, followed by an inference stage to rectify discrepancies between the two count sets. Other approaches, such as those in [18] and [19], concentrate on modifying the granularity of space to enhance the utility of data in the publication of sanitized datasets.

7 CONCLUSION

Our study addressed critical privacy challenges in publishing electricity consumption data, balancing protection concerns with data utility. Our proposed innovative solution, STPT, significantly improves DP-compliant data publication accuracy by integrating time series data with the spatial attributes of households. This unique approach utilizes the short-term and long-term memory capabilities of RNNs for sophisticated pattern recognition, capturing both micro and macro consumption patterns. Our extensive experiments with real-world and synthetic datasets demonstrate STPT’s superior performance in maintaining high data utility while ensuring

Table 1: Summary of Notations

Symbol	Description
\mathcal{D}	Time series data database
N	Number of households
\mathcal{U}	Set of households (or power grid users)
$x_{i,t}$	User i ’s consumption at time t
C_{cons}	Actual consumption matrix
C_{norm}	Normalized consumption matrix
C_{pattern}	Pattern estimate matrix
$C_{\text{sanitized}}$	Sanitized consumption matrix
ϵ_{tot}	Total privacy budget
$\epsilon_{\text{pattern}}$	Pattern recognition budget
$\epsilon_{\text{sanitize}}$	Sanitization budget
\mathcal{P}	Partition set
s_i	Partition i sensitivity
T_{train}	RNN training time

robust privacy protection, compared to existing methods. The suggested method, while primarily aimed at enhancing the utility of DP compliant publishing electricity data, is versatile and can be applied to various situations, including Wireless Sensor Networks and the publication of health-related data, offering potential for future research applications.

A TABLE OF NOTATIONS

Table 1 summarizes the notations used throughout the manuscript.

B PROOF OF THEOREMS

B.1 Proof of Theorem 4

Recall that the consumption matrix is constructed such that the time series resolution matches the time axis resolution. As a result, each matrix cell contains no more than a single data point of an individual household/user. Consequently, adding or removing a user from the data can alter the value in a matrix cell by at most $\max_{i,t} x_{i,t}$. If the time series are normalized to values between 0 and 1, then this sensitivity would be 1.

B.2 Proof of Theorem 5

The sequential decomposition in time is due to the correlation of time series over time. The parallel decomposition of the privacy budget over space is due to the fact that the time series of users are spatially bounded in the matrix and independent of the values in other cells.

B.3 Proof of Theorem 6

Consider a cell at time t corresponding to a sub-region at depth i of the tree and all users j falling in the sub-region. Let us denote the consumption of user i before and after the removal of an individual by $x_{i,t}$ and $x'_{i,t}$, respectively. The maximum change observed in the representative time series of the sub-region denoted by M at time t can be derived as,

$$\frac{|\sum_{i \in M} x_{i,t} - \sum_{i \in M} x'_{i,t}|}{4 \log_2(C_x) - i} = \frac{|x_{j,t} - x'_{j,t}|}{4 \log_2(C_x) - i} \leq \frac{1}{4 \log_2(C_x) - i} \quad (12)$$

Table 2: Electricity Consumption Data Summary

Dataset	Number of Households	Average Hourly Consumption (kWh)	STD of Hourly Consumption (kWh)	Maximum Hourly Consumption (kWh)	Sensitivity Clipping Factor
CER	5000	0.61	1.24	19.62	1.85
CA	250	0.38	1.13	33.54	1.51
MI	250	0.48	1.22	49.50	1.7
TX	250	0.55	1.63	68.86	2.18

In the above equation, index j denotes the datapoint corresponding to the user whose existence in the dataset is altered. Therefore, the addition or removal of an individual can change the value of the representative point by at most $\frac{1}{4\log_2(C_X)-i}$.

B.4 Proof of Theorem 7

Denote by s the maximum number of cells in a xy -axis pillar within the partition P_i . Given that the maximum cell count in each xy -axis pillar is bounded by p , and each pillar represents a unique time series, the addition or removal of an individual alters the cumulative values in the cluster by at most p . Hence, the sensitivity is characterized by this maximal change.

B.5 Proof of Theorem 8

The amount of noise added to each partition can be quantified using the variance of Laplace noise. Here, the goal is to distribute the privacy budget across partitions such that the total variance of applied noise is minimized. Equation 13 formulates this goal as a convex optimization problem.

$$\min_{\epsilon_1 \dots \epsilon_m} \sum_{i=1}^m s_i^2 / \epsilon_i^2 \quad (13)$$

$$\sum_{i=1}^m \epsilon_i = \epsilon_{\text{sanitize}}, \quad \epsilon_i > 0 \quad \forall i = 1 \dots m \quad (14)$$

Writing Karush-Kuhn-Tucker (KKT) [20] conditions, the optimal allocation of budget can be calculated as:

$$L(\epsilon_1, \dots, \epsilon_m, \lambda) = \sum_{i=1}^m s_i^2 / \epsilon_i^2 + \lambda \left(\sum_{i=1}^m \epsilon_i - \epsilon_{\text{sanitize}} \right) \quad (15)$$

$$\Rightarrow \frac{\partial L}{\partial \epsilon_i} = -\frac{2s_i^2}{\epsilon_i^3} + \lambda = 0 \quad (16)$$

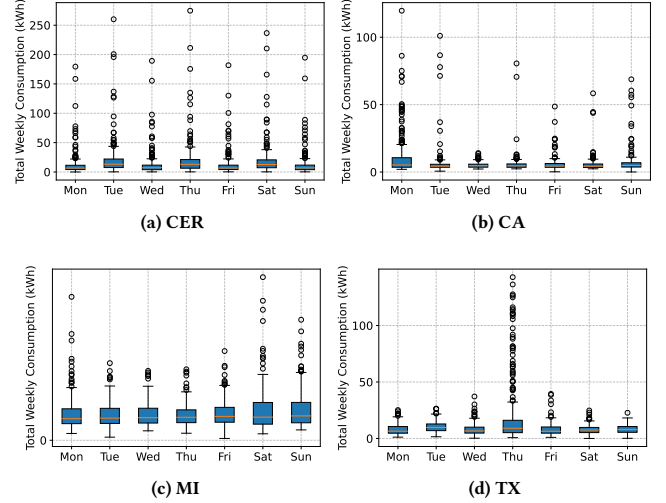
$$\Rightarrow \epsilon_i = \frac{2^{1/3} s_i^{2/3}}{\lambda^{1/3}}, \quad (17)$$

Substituting ϵ_i 's in the constraint equation, the optimal budget at the i -th level is derived as

$$\epsilon_i = \frac{\epsilon_{\text{sanitize}} \times s_i^{2/3}}{\sum_{i=1}^m s_i^{2/3}}. \quad (18)$$

C DATASET STATISTICS

Table 2 and Figure 6 summarize the statistics of datasets used in the experiments.

**Figure 6: Total Weekly Consumption per Week Day.**

D HYPER-PARAMETERS

The total privacy budget is set to $\epsilon_{\text{tot}} = 30$, with $\epsilon_{\text{pattern}} = 10$ allocated for pattern recognition in STPT, and $\epsilon_{\text{sanitize}} = 20$ for sanitization. The same privacy budget is utilized across all algorithms. For training in the STPT algorithm, 100 datapoints are used, resulting in a training matrix of $32 \times 32 \times 100$. The test involves 120 points, leading to a matrix of $32 \times 32 \times 120$. Consequently, the published consumption matrix has dimensions of $32 \times 32 \times 120$. The sensitivity clipping factor of the consumption matrix is provided in Table 2. The RNN unit comprises a self-attention mechanism and a GRU unit. Training was conducted over 20 epochs, with a batch size of 32. The time window is set to encompass 6 datapoints for predicting the next datapoint. The RMSProp optimizer is employed with a learning rate of $1e-3$. The embedding size and hidden dimension are set to 128 and 64, respectively.

E HARDWARE AND SOFTWARE SPECIFICATIONS

Experiments were ran on a cluster node equipped with an 18-core Intel i9-9980XE CPU, 125 GB of memory, and two 11 GB NVIDIA GeForce RTX 2080 Ti GPUs. Furthermore, all neural network models are implemented based on PyTorch version 1.13.0 with CUDA 11.7 using Python version 3.10.8.

REFERENCES

- [1] C. Dwork, "Differential privacy: A survey of results," in *International conference on theory and applications of models of computation*. Springer, 2008, pp. 1–19.
- [2] L. Fan and L. Xiong, "An adaptive approach to real-time aggregate monitoring with differential privacy," *IEEE Transactions on knowledge and data engineering*, vol. 26, no. 9, pp. 2094–2106, 2013.
- [3] F. Leukam Lako, P. Lajoie-Mazenc, and M. Laurent, "Privacy-preserving publication of time-series data in smart grid," *Security and Communication Networks*, vol. 2021, pp. 1–21, 2021.
- [4] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," in *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, 2018.
- [5] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory of cryptography conference*. Springer, 2006, pp. 265–284.
- [6] F. D. McSherry, "Privacy integrated queries: an extensible platform for privacy-preserving data analysis," in *Proceedings of the 2009 ACM SIGMOD International Conference on Management of data*, 2009, pp. 19–30.
- [7] J. Xu, Z. Zhang, X. Xiao, Y. Yang, G. Yu, and M. Winslett, "Differentially private histogram publication," *The VLDB journal*, vol. 22, pp. 797–822, 2013.
- [8] C. for Energy Regulation (CER), "Cer smart metering project - electricity customer behaviour trial, 2009-2010," 2012, [Dataset]. [Online]. Available: <https://www.ucd.ie/issda/data/commissionforenergyregulationcer/>
- [9] S. Thorve, Y. Y. Baek, S. Swarup, H. Mortveit, A. Marathe, A. Vullikanti, and M. Marathe, "High resolution synthetic residential energy use profiles for the united states," *Scientific Data*, vol. 10, no. 1, p. 76, 2023.
- [10] T. Babaei, H. Abdi, C. P. Lim, and S. Nahavandi, "A study and a directory of energy consumption data sets of buildings," *Energy and Buildings*, vol. 94, pp. 91–99, 2015.
- [11] V. Rastogi and S. Nath, "Differentially private aggregation of distributed time-series with transformation and encryption," in *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*, 2010, pp. 735–746.
- [12] L. Lyu, Y. W. Law, J. Jin, and M. Palaniswami, "Privacy-preserving aggregation of smart metering via transformation and encryption," in *2017 IEEE Trust-com/BigDataSE/ICSS*. IEEE, 2017, pp. 472–479.
- [13] S. Shaham, G. Ghinita, R. Ahuja, J. Krumm, and C. Shahabi, "Htf: homogeneous tree framework for differentially-private release of location data," in *Proceedings of the 29th International Conference on Advances in Geographic Information Systems*, 2021, pp. 184–194.
- [14] S. Song, Y. Wang, and K. Chaudhuri, "Pufferfish privacy mechanisms for correlated data," in *Proceedings of the 2017 ACM International Conference on Management of Data*, 2017, pp. 1291–1306.
- [15] X. Zhang, M. M. Khalili, and M. Liu, "Differentially private real-time release of sequential data," *ACM Transactions on Privacy and Security*, vol. 26, no. 1, pp. 1–29, 2022.
- [16] R. McKenna, G. Miklau, M. Hay, and A. Machanavajjhala, "Optimizing error of high-dimensional statistical queries under differential privacy," *Proc. VLDB Endow.*, vol. 11, no. 10, p. 1206–1219, 2018.
- [17] Y. Xiao, L. Xiong, L. Fan, S. Goryczka, and H. Li, "Dpcube: Differentially private histogram release through multidimensional partitioning," vol. 7, no. 3, p. 195–222, 2014.
- [18] W. Qardaji, W. Yang, and N. Li, "Differentially private grids for geospatial data," in *2013 IEEE 29th international conference on data engineering (ICDE)*. IEEE, 2013, pp. 757–768.
- [19] S. Shaham, G. Ghinita, and C. Shahabi, "Differentially-private publication of origin-destination matrices with intermediate stops," in *Proceedings of the 25th International Conference on Extending Database Technology (EDBT)*, 2022.
- [20] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.