Hindawi
Security and Communication Networks
Volume 2021, Article ID 6665984, 10 pages
https://doi.org/10.1155/2021/6665984

Research Article

Differentially Private Autocorrelation Time-Series Data Publishing Based on Sliding Window

Jing Zhao, Shubo Liu, Xingxing Xiong, and Zhaohui Cai

School of Computer Science, Wuhan University, Wuhan 430072, China

Correspondence should be addressed to Shubo Liu; liu.shubo@whu.edu.cn

Received 10 December 2020; Revised 31 March 2021; Accepted 10 April 2021; Published 24 April 2021

Academic Editor: Mamoun Alazab

Privacy protection is one of the major obstacles for data sharing. Time-series data have the characteristics of autocorrelation, continuity, and large scale. Current research on time-series data publication mainly ignores the correlation of time-series data and the lack of privacy protection. In this paper, we study the problem of correlated time-series data publication and propose a sliding window-based autocorrelation time-series data publication algorithm, called SW-ATS. Instead of using global sensitivity in the traditional differential privacy mechanisms, we proposed periodic sensitivity to provide a stronger degree of privacy guarantee. SW-ATS introduces a sliding window mechanism, with the correlation between the noise-adding sequence and the original time-series data guaranteed by sequence indistinguishability, to protect the privacy of the latest data. We prove that SW-ATS satisfies ε-differential privacy. Compared with the state-of-the-art algorithm, SW-ATS is superior in reducing the error rate of MAE which is about 25%, improving the utility of data, and providing stronger privacy protection.

1. Introduction

Time-series data are a set of sequential, large, and continuous data sequences. In general, time-series data can be regarded as a dynamic dataset that grows infinitely over time. Using the correlation between data values to analyze and mine time-series data can bring considerable benefits to government, enterprises, and social public services. For example, in this outbreak of COVID-19, monitoring and analyzing the patient's physical condition can effectively treat the disease and control the spread of the epidemic. The navigation software needs to count the total amount of traffic in a specific time range of each road to calculate the best route to the destination.

The above example illustrates the importance of publishing time-series data for knowledge discovery and acquisition. However, if the curator does not adopt appropriate privacy protection technology and publish the data directly, it will leak personal sensitive information and violate citizens' privacy.

Traditional data publishing mainly uses anonymous technology, such as k-anonymity [1] model and its derivative model [2, 3] for privacy protection. However, these methods are strongly dependent on the attacker's background knowledge assumptions and cannot provide an effective and rigorous method to prove its privacy protection level. Some studies [4, 5] adopt the technology of combining blockchain and artificial intelligence (AI) to protect the privacy of data, but the technology will bring into low efficiency, and once there may be some vulnerabilities, it will confront some risks of significant attack. Differential privacy [6] is a strict and provable privacy protection technology, which can protect users' sensitive information from leaking their privacy [7]. By adding random noise, it limits the impact of any record on the released statistical results to blur the existence of the record in the dataset, and such users' privacy will be fundamentally protected. This model is widely used for the release of various data in many application scenarios [8]. For the privacy leakage problem in time-series data publication, the existing work can also be solved by using the differential privacy model. Dwork et al. [9] achieved event-level differential privacy in the scenario of continuous statistics data publication. In order to reduce the noise added in the original time-series data, Chan et al. [10] proposed to adopt a

binary tree-based divide-and-conquer method to decompose and store time-series data.

Motivations and Contributions. Traditional differential privacy is widely used for data publication, while Kifer et al. [11] pointed out that it still encountered the risk of leaking personal privacy for publishing correlated time-series data. The current publication methods of differential privacy on correlated time-series data mainly include the methods of establishing correlation models, such as covariance matrix [12] and Markov [13, 14], and data transformation, e.g., the Fourier transform [15] and discrete wavelet transform (DWT). The abovementioned methods of differential privacy focus on the publication of independent and identically distributed (IID) data, which will lead to the following problems:

> Insufficient privacy protection: adding independent identically distributed noise to the correlated data will cause the attacker to filter out the noise through filtering attacks and other methods, thus causing the user's privacy to be disclosed.
>
> Low data utility: since IID noise is added to the correlated data, it will lead to the reduction of privacy protection level. In order to maintain the same level of differential privacy protection, more noise needs to be added, resulting in a sharp decrease in the utility of published data.

These issues indicate that the current methods of differential privacy are not suitable for processing time-series data with correlation. Although Wang et al. [16] proposed the CTS-DP method to resist filtering attacks by adding noise consistent with the correlation of the original data, it ignored the periodicity of time-series data and failed to provide adequate privacy protection. It also does not apply to the publication of dynamic data. Compared to the existing work, our main contributions are summarized as follows.

First, because time-series data exhibit periodic changes and have strong autocorrelation, even if a single record in the dataset is deleted, an attacker can infer information about missing records from other correlated records. We propose periodic sensitivity to replace the global sensitivity in traditional differential privacy to avoid this situation and provide a stronger degree of privacy protection under the same privacy budget. Second, based on the periodic sensitivity, we propose a sliding window mechanism to process infinitely growing and correlated time-series data. Third, we theoretically proved that our proposed correlated time-series data publication algorithm based on sliding window (SW-ATS) satisfies differential privacy. And compared with the state-of-the-art method, the experimental results show that SW-ATS can reduce more errors and provide stronger privacy protection.

2. Related Work

In the early research on differential privacy data publication, most literature studies assume that the data are independent. At present, the research on differential privacy on correlated data is still relatively limited. Because the main research obstacle of correlated differential privacy is that correlated records can provide additional information for attackers, while traditional mechanisms can hardly model it. In this case, meeting the definition of differential privacy is a complex task. Kifer et al. [11] proposed for the first time that differential privacy would reduce privacy guarantees on correlated datasets if the correlation between data is not considered. For example, suppose that a record r has an impact on a group of records. Even if the record r is deleted from the dataset, the relevant information of r can be derived from this group of records. In this case, the traditional differential privacy cannot provide enough privacy protection. Chen et al. [17] treated social networks as correlated datasets and solve the problem of insufficient privacy protection by multiplying the global sensitivity by the number of correlated records. However, this method introduces too much noise, making the utility of datasets decline sharply.

In the research of correlated time-series data, Cao et al. [18] used internal coupling and internal coupling behavior functions to model related information and used these functions in the association framework to express the degree of association between behaviors. They proposed a hidden Markov detection model to detect abnormal transaction behavior based on grouping. They defined a time interval and assumed that behaviors falling within the same interval are related behaviors. Song et al. [19] proposed a hybrid coupling framework, which uses some special attributes to identify the relationship between records. Zhang et al. [20] proposed a related network traffic classification algorithm, using IP address to identify network traffic correlated records. Zhou et al. [21] mapped correlated records to an undirected graph and proposed a multi-instance learning algorithm.

Wang et al. [16] proposed the concept of sequence indistinguishability and proved that the correlations between the original time series and the time series after adding noise are consistent; then, the added noise meets the differential privacy. The differential time-series data publication algorithm CTS-DP proposed by them adds correlated noise to ensure the correlation of added noise. Zhu et al. [12] defined correlation sensitivity. They considered the correlation between records and proposed an effective related differential privacy solution, CIM (correlated iteration mechanism). CIM uses the covariance matrix to describe the correlation between sequences and uses the covariance matrix as the weight to calculate the sensitivity function. Experimental results show that this solution is superior to traditional differential privacy in terms of the mean squared error in response to large batches of queries. This also shows that the correlated differential privacy can successfully protect privacy while maintaining the practicality of the data.

Some scholars convert the correlated time-series data to another independent domain for processing while retaining the main characteristics of the original sequence. Rastogi et al. [15] proposed a Fourier transform (FPA) method to solve this problem. In FPA, the discrete Fourier transform (DFT) is used to convert the correlated data into an independent Fourier domain. Approximately reconstruct the

DFT coefficients of the original sequence. To overcome the shortcomings of FPA when applied to short-term and nonstationary sequences, discrete wavelet transform (DWT) was proposed in [22, 23]. DWT extends the range of FPA and retains more features of the sequence. Although there are difficulties in ensuring differential privacy, the literature [24–26] uses principal component analysis (PCA) to extract the features of the dataset to another dimension, and the disturbance data published can be applied to some common statistical learning applications. Table 1 provides a summary of recent studies in correlated time-series data publication of differential privacy.

Summary. Currently, on the issue of differential privacy correlated time-series data, some methods add independent noise on the correlated time-series data, which is easy to be attacked. The other methods add correlated noise but ignore the periodic changes of time-series data, resulting in insufficient privacy protection. What is more, the current method can only be applied to the publication of static data. This article attempts to solve the following problems:

How to dynamically publish correlated time-series data?

How to deal with the lack of privacy intensity due to the periodic changes of correlated time-series data?

3. Preliminary Knowledge

3.1. Differential Privacy. Dwork et al. [11] proposed the differential privacy model for the first time, which is a strong privacy protection framework. By limiting the influence of the change of a single record in the dataset on the query results, the attacker cannot accurately obtain the sensitive information in the record even if he knows all the record information except a certain record.

Definition 1 (ε-differential privacy [26]). Consider two neighboring datasets, D and $D\prime$. For each output $O \subseteq \text{range}(A)$ of a neighboring dataset, if the random algorithm A satisfies

$$\Pr(A(D) \in O) \leq e^{\varepsilon} \times \Pr(A(D') \in O), \tag{1}$$

then the algorithm A satisfies ε-differential privacy.

Definition 2 (Global sensitivity [28]). Suppose there is a query function $f: D \longrightarrow R^d$, which takes a dataset D as input and outputs a d-dimensional real vector R. For any neighboring datasets, D and D', the global sensitivity of the function f is defined as

$$GS_f = \max_{D,D'} \| f(D) - f(D') \|_1. \tag{2}$$

Definition 3 (Laplace mechanism [28]). Given a dataset D and a function $f: D \longrightarrow R^d$ with sensitivity GS_f. The random algorithm,

$$M(D) = f(D) + \text{Lap}\left(\frac{GS_f}{\varepsilon}\right), \tag{3}$$

provides ε-differential privacy protection.

Theorem 1. *parallel combinatorial properties [29]). With a random algorithm sequence $A_1, A_2, \ldots, A_n$ and the random processes of any two algorithms that are independent of each other, the privacy protection budget is $\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n$, respectively. For disjoint datasets $D_1, D_2, \ldots, D_n$, the combined algorithm $A(A_1(D_1), A_2(D_2), \ldots, A_n(D_n))$composed of these algorithms provides $(\max \varepsilon_i)$–differential privacy protection.*

3.2. Problem Definition. Time-series data are a set of sequential, large, and continuous data sequences. In general, time-series data can be regarded as a dynamic dataset that grows infinitely over time. For example, Table 2 shows the blood glucose data collected by different users within one month of time-series data.

Considering the following scenarios, user A wants to query the average value of blood glucose data within the range of T_1-T_2; user B wants to query the number of people whose blood pressure is greater than 140 mmHg at time T_3 ... The goal of this article is to use differential privacy technology to publish correlated time-series data, and users can obtain meaningful query results under the premise that personal privacy in the database is not leaked. The curator aggregates the time-series data of all users and divides it into $D_n = \{D_1, \ldots, D_n\}$ subdatasets according to the data attributes. Each subdataset D_i is divided into $S_{i,m} = \{S_{i,1}, \ldots, S_{i,m}\}$ pieces of disjoint time-series data according to the user dimension. The curator finally publishes all data on the premise of satisfying differential privacy and responds to user queries as shown in Figure 1.

For any piece of time-series data X, it can be treated as a short-term stationary sequence, and its autocorrelation can be expressed using an autocorrelation function.

Definition 4 (Autocorrelation function [30]). The correlation of time-series data can be expressed by the autocorrelation function. For the original time-series data X, the autocorrelation function can be expressed as

$$R_{XX}(\tau) = N_0 \delta(\tau). \tag{4}$$

Among them, N_0 represents the power spectral density of X and $\delta(\tau)$ represents the impulse function.

Definition 5 (Sequence indistinguishability [16]). If the original time-series data X and the noise sequence Z to be released have the same normalized autocorrelation functions, that is,

$$R_Z(\tau) = R_{XX}(\tau), \tag{5}$$

then the noise sequence and the original sequence are indistinguishable to the attacker, and the attacker cannot

Table 1: Summary of literature survey.

Algorithm	Advantage	Limitation
Pufferfish [27]	The algorithm takes into account the correlation between data	Does not satisfy differential privacy
PCA [24–26], DFT [15], and DWT [22,23]	Under the premise of keeping the main characteristics of the sequence unchanged, the correlation time series is transformed into another independent domain for processing	Independent noise is added and the sequence correlation is destroyed to some extent
CIM [12]	Literature [12] proposed correlated sensitivity to reduce noise and utilized a correlation coefficient matrix to describe the correlation of a series	It is only applicable to the publication of histogram statistics
CTS-DP [16]	The correlation noise is added to the original time-series data	Dynamic data cannot be processed and privacy protection is inadequate

Table 2: User blood glucose monthly statistics (mg/dl).

User	t_1	t_2	t_3	...	t_n
James	186	203	196	...	260
Mary	140	132	129	...	148
Jane	167	152	198	...	176
...	...	...	...	...	...
Tom	188	239	197	...	204

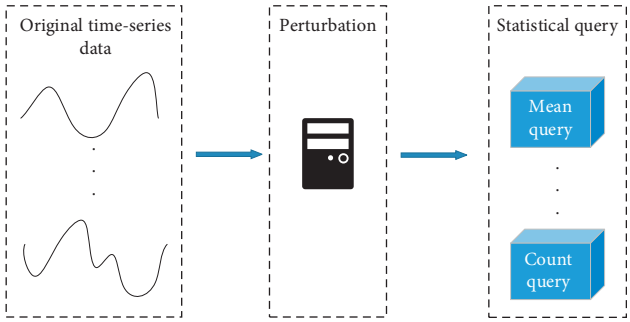

Figure 1: Data publication scenario.

simply use knowledge about the correlation of the original sequence to launch the attack.

4. Correlated Time-Series Data Publishing Algorithm Based on Sliding Window

In real life, time-series data are a dynamic dataset with infinite growth over time. Therefore, on the basis of the CTS-DP algorithm, this paper uses the sliding window mechanism for any length of time-series data to realize the continuous publication of time-series data under the premise of satisfying differential privacy. In order to solve the problem of insufficient privacy protection in the CTS-DP algorithm, we propose periodic sensitivity instead of global sensitivity to achieve greater privacy protection.

4.1. Sliding Window Model. Define time-series data $X = \{D_1, D_2, \ldots, D_t\}$, where D_t represents the data value at time t. The sliding window model is used to model the time-series data X, each sliding window is defined as w_i, and the sliding window size is w. The data contained in each sliding window is $X_{W_i} = \{D_i, D_{i+1}, \ldots, D_{i+w-1}\}$, and the data to be published after processing by the algorithm is $\tilde{X}_{W_i} = \{\tilde{D}_i, \tilde{D}_{i+1}, \ldots, \tilde{D}_{i+w-1}\}$.

The sliding window in time-series data refers to specifying an interval on the time-series data, which contains the latest data. The purpose is to limit the infinite data stream and obtain data characteristics. With the arrival of new data, the data in the sliding window are processed after the amount of data reaches the set sliding window size. Then slide the window forward and wait for the next set of data. Figure 2 shows the process of publishing time-series data using the sliding window model.

Differential privacy protection under time-series data is divided into two levels: the event level and the user level [9]. The former protects every event in the time-series data sequence, while the latter protects all user behaviors. This paper is aimed at the privacy protection of the event level, protecting each event in the time-series data sequence.

4.2. The Sampling Period of Time-Series Data. Time-series data usually have a strong characteristic of periodic change. According to the characteristic of timing data showing a periodic change, the sampling period of the timing data can be determined. For example, the blood glucose of normal people remains in a constant range before three meals a day and before bedtime. Usually, the sampling frequency of health data within a day is taken as a period. Taking the blood glucose data as an example, the blood glucose data are sampled four times a day, and then the sampling period of blood glucose data is $T = 4$. For some data that can only obtain a single statistical value in a day, such as the number of steps, the sampling frequency of the data within a week or month can be used as the period, that is, $T = 7$ or $T = 30$.

4.3. Periodic Sensitivity. Since the time-series data have strong periodic changes, if the global sensitivity is still adopted at this time, it will indeed increase the risk of privacy leakage.

For example, someone's blood pressure surged recently due to staying up late. If users query the blood pressure value of a day at this time, they will have a higher probability to infer the other approaching blood pressure samples. Therefore, in order to ensure that the data are not leaked, it is necessary to delete all the sampling data before and after approaching this blood pressure value. At this time, if the global sensitivity is still sampled to generate Laplacian noise,

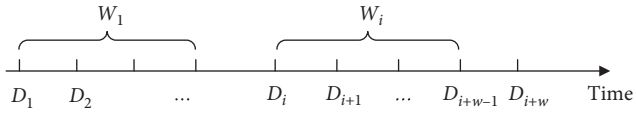

Figure 2: Time-series data publication under sliding window.

it is obviously unable to better protect the data from leakage. Based on this, this paper proposes periodic sensitivity to replace global sensitivity to provide stronger privacy protection.

Definition 6 (periodic sensitivity). According to the attribute N of time-series data, determine the sampling period T of this attribute, and then the periodic sensitivity is defined as

$$PS_q = \max\left\|Q(X) - Q\left(X^{-T_i}\right)\right\|, \; i \in \left[1, \frac{|X|}{|T|}\right]. \tag{6}$$

Among them, X represents a piece of time-series data of attribute N, Q represents the query function, $-T^i$means removing all data in the i-th sampling period, and $|T|$ represents the number of sampled data points in a period.

4.4. Algorithm Design. The SW-ATS algorithm can iteratively process and publish the existing data (static data) in the database, and the recently arrived data (dynamic data) can be processed and published after the data volume meets the sliding window size. Or adjust the size of the sliding window to the size of the newly added data before publishing. The establishment process of the SW-ATS algorithm is shown in Algorithm 1.

Algorithm 1 shows the basic framework of SW-ATS. SW-ATS divides the original time series X into n subsequences according to the sliding window length L (line 1) and iteratively processes the subsequences in each sliding window (2~9 lines). First, calculate the autocorrelation function of the subsequence Sub_i (line 3) and periodic sensitivity (line 4); then generate 4 groups of white Gaussian noise (line 5) with the same length as the subsequence and the power spectral density of $\sqrt{2\lambda}$, where $\lambda = \Delta f/\varepsilon$ (the ratio of sensitivity and privacy budget) (line 4); four groups of Gaussian white noise are convolved with the impulse response to obtain four groups of Gaussian noise sequences with autocorrelation function $R_{G_i}(\tau) = \sqrt{R_{\mathrm{Sub}_i\mathrm{Sub}_i}(\tau)/8}$ (line 6); finally, Laplacian noise can be obtained by using the sum of the two Gaussian noise groups' squares minus the sum of the squares of the other two, sample from which at intervals of 1 can calculate the Laplacian noise of length L (line 7); by splicing all Laplace noise of length L and adding them to the original time series, the final noise-adding sequence $\widetilde{X}$ is gained and ready for publishing (lines 8 and 9).

For the newly added data, when the amount of data reaches the size of the sliding window, the sequence X_t is obtained, and steps 3–7 in Algorithm 1 is directly executed to obtain the sequence Sub_t'with noise. Then execute $\widetilde{X}_t = X_t + Sub_i$' and publish $\widetilde{X}_t$.

4.5. Privacy analysis

Theorem 2. *Algorithm SW-ATS satisfies ε-differential privacy.*

Proof. Literature [16] has proved that if the original time-series data and the noise sequence added to the time-series data meet Definition 5, then the published noise sequence meets ε-differential privacy.

$$\sup_{x^i, x_i^{'}, R_{XX}(\tau), S} \ln \frac{\Pr\left[M(X) \in S | X_i, R_{XX}(\tau)\right]}{\Pr\left[M(X\imath) \in S | X_i', R_{XX}(\tau)\right]} \le \varepsilon. \tag{7}$$

Therefore, according to Theorem 1, the algorithm SW-ATS satisfies ε-differential privacy. □

Theorem 3. *The noise sequence generated by the algorithm SW-ATS in each sliding window is correlated with the original sequence.*

Proof. Literature [16] has proved that if the autocorrelation function R_{G_i} satisfies $R_{G_i}(\tau) = \sqrt{R_{\mathrm{Sub}_i\mathrm{Sub}_i}(\tau)/8}$, then the autocorrelation function of the noise sequence calculated by $\mathrm{Sub}_i' = G_1'^2 + G_2'^2 - G_3'^2 - G_4'^2$ satisfies $R_{\widetilde{X}_t}(\tau) = R_{\mathrm{Sub}_i\mathrm{Sub}_i}(\tau)$, so the noise sequence and the original time series have correlation within the same sliding window. □

4.6. Time Complexity Analysis. In Algorithm 1, the time complexity of steps 3, 4, 5, and 6 is $O(L^2)$, $O(4L)$, $O(4)$, and $O(4L^2)$, respectively. Since the algorithm needs to iterate on each sliding window, the total computational complexity of the algorithm is

$$\begin{aligned} T(n) &= \frac{n}{L}\left(O\left(L^2\right) + O(4L) + O(4) + O\left(4L^2\right)\right), \\ &\approx O(nL), \end{aligned} \tag{8}$$

where L is the length of the sliding window. When the length of the sliding window is the same as the original sequence, the time complexity of the algorithm is $O(n^2)$. With the continuous increase of new data, only the latest data can be calculated, so for the recently arrived data, the time complexity is $O(L^2)$.

4.7. Utility Analysis. This paper uses the differential privacy utility definition proposed by Blum et al. [31] to perform utility analysis.

Definition 7 ((α, β)-accuracy [31]). For a query set Q, if for each query $Q_t \in \mathrm{Q}$ and the original dataset X, the privacy protection mechanism M can satisfy equation (9) with a probability $1 - \beta$, then M satisfies (α, β)-accuracy.

$$\max_{Q_t \in Q} \left|Q_t(x\prime) - Q_t(x)\right| \le \alpha. \tag{9}$$

For any query $Q_t \in \mathrm{Q}$, it is known that $\beta > 0$ holds, and the generalized Laplace mechanism satisfies

Input: original time series X
Output: time series to be published after adding noise $\bar{X}$
(1) Read the original time series X and divide X into n subsequences $\mathrm{Sub}_1, \mathrm{Sub}_2, \ldots, \mathrm{Sub}_n$ using the sliding window length L, where $n = \lfloor |X|/L \rfloor$.
(2) **for** $i = 1$ **to** n:
(3) Calculate the autocorrelation function $R_{\mathrm{Sub}_i\mathrm{Sub}_i}(\tau)$ of the subsequence Sub_i.
(4) According to the query function q, calculate the periodic sensitivity PS_q of the time-series data X, where PS_q is computed by equation (5).
(5) Generate four IID Gauss white noise series G_1, G_2, G_3, G_4, which have the same length as $|\mathrm{Sub}_i|$. In addition, $G_i \sim N(0, \sqrt{2\lambda}), i \in \{1, 2, 3, 4\}$, where $\lambda = PS_q/\varepsilon$.
(6) Calculate $G_1' = G_1 \otimes h(\tau)$, $G_2' = G_2 \otimes h(\tau)$, $G_3' = G_3 \otimes h(\tau)$, and $G_4' = G_4 \otimes h(\tau)$, where $h(\tau) = \sqrt{R_{\mathrm{Sub}_i\mathrm{Sub}_i}(\tau)/16\pi N_0}$.
(7) $\mathrm{Sub}_i' = G_1'^2 + G_2'^2 - G_3'^2 - G_4'^2$.
(8) Splice Sub_i' at the end of Z.
(9) **end for**
(10) $\bar{X} = X + Z$
(11) **Return** $\bar{X}$

Algorithm 1: SW-ATS.

$(\alpha, 1 - \exp(-\alpha\varepsilon/S_{K,U}(f))/2)$ – accuracy with a probability of at least

$$1 - \frac{\exp\left(-\alpha\varepsilon/S_{K,U}(f)\right)}{2}. \tag{10}$$

Proof. Let E_n represent the error introduced by generalized Laplace noise, then

$$\Pr\left(\max_{Q_t \in Q} \left|Q_t(x') - Q_t(x)\right| > \alpha\right) = \Pr\left(E_n > \alpha\right). \tag{11}$$

Since $Q_t(D') = Q_t(D) + y_{K,U}$, where $y_{K,U} \sim GL(S_{K,U}(f)/\varepsilon, C_{K,U})$, according to the properties of the Laplace distribution, there is $\Pr(E_n > \alpha) = \int_{-\infty}^{\alpha} f(x)dx = 1 - \exp(-\alpha\varepsilon/S_{K,U}(f))/2$; then, $\Pr(E_n \le \alpha) = 1 - \Pr(E_n > \alpha) = \exp(-\alpha\varepsilon/S_{K,U}(f))/2$; given α, there is $\beta = \exp(-\alpha\varepsilon/S_{K,U}(f))/2$. □

5. Experimental Evaluation

This experiment uses MATLAB language to realize the correlated time-series differential privacy publishing algorithm based on sliding window. The experimental environment is Inter (R) Core (TM) i5 2.7 GHz, 4 GB memory, Windows 7 operating system. We used two real-world datasets in our evaluations as this has helped in illustrating the effectiveness of our approach in real-world applications.

Diabetes (http://archive.ics.uci.edu/ml/datasets/Diabetes). Diabetes dataset is a representative standard classification dataset in the UCI machine learning dataset. The records were obtained from two sources: an automatic electronic recording device and paper records. The automatic device had an internal clock to timestamp events, whereas the paper records only provided "logical time" slots (breakfast, lunch, dinner, and bedtime). For paper records, fixed times were assigned to breakfast (08 : 00), lunch (12 : 00), dinner (18 : 00), and bedtime (22 : 00).

Steps. The data are collected by teachers and students through smart bracelets and mobile phones. Table 3 shows some of the fields in the dataset, including start date, end date, and value. It means that the number of steps someone took during the period from 2019-05-14 10 : 37 : 07 to 2019-05-14 11 : 49 : 32 is 956 steps. Moreover, the start and end dates of each sampling are not fixed, indicating that the smart bracelet and mobile phones collect and count the number of steps in multiple periods within a day. After sorting out, the step data collected in each period of the day are merged to obtain the step data in the unit of day.

Metrics. In the experiment, to verify the effectiveness of the proposed algorithm in this paper, SW-ATS and CTS-DP algorithms are compared. In terms of data utility evaluation, the mean absolute error (MAE) was used to measure the effectiveness. MAE was defined as follows:

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N}\left|x_i' - x_i\right|, \tag{12}$$

where N represents the length of the time series, and the lower MAE means the better utility of data.

5.1. Experimental Results. Nowadays, CTS-DP is the state-of-the-art method to publish correlation time-series data. Therefore, we choose the CTS-DP algorithm as a comparison.

5.1.1. Impact of Sliding Window Size on Data Utility. Figure 3 shows a graph of the experimental results of the two algorithms under different sliding window sizes when the privacy budget ε is 1 and 0.5, respectively. In the Diabetes dataset, a peace of time series was randomly selected from the experiment for processing. Each algorithm was tested 1000 times, and the experimental results were averaged 1000 times. It can be seen that the result of SW-ATS is obviously better than that of CTS-

Table 3: Some fields of the Steps dataset.

Field	Sample
Start date	2019-05-14 10 : 37 : 07 + 0800
End date	2019-05-14 11 : 49 : 32 + 0800
Value	956

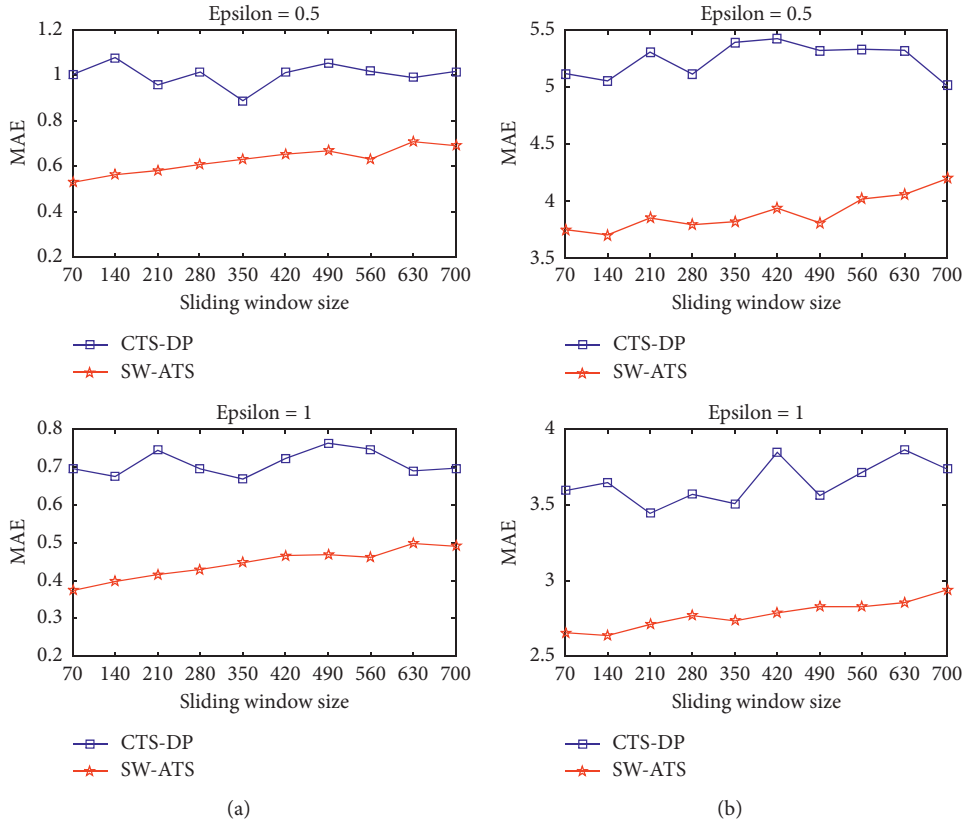

Figure 3: Utility comparison when the size of sliding window changes. (a) MAE (diabetes dataset). (b) MAE (steps dataset).

DP, and the average error is reduced by 37.5%. As the size of the sliding window continues to increase, the MAE of SW-ATS also increases. In the Steps dataset, the dataset was first divided into 7 intervals according to the number of steps (an interval less than or equal to 3000 steps and an interval greater than 21000), and then the number of people in each interval was counted every day to form 7 statistical time-series data. The experimental results also show that the results of SW-ATS are better than those of CTS-DP, and the average error is reduced by 24.9%. With the increasing size of the sliding window, the effect of SW-ATS keeps increasing, but it is always smaller than that of CTS-DP.

5.1.2. Impact of Epsilon on Data Utility. Figure 4 shows the comparison of the results of the two algorithms under different privacy budgets when the sliding window sizes are $5T = 35$ and $10T = 70$, respectively. With the increase of the privacy budget, the MAE of both algorithms is decreasing, and the algorithm SW-ATS proposed in this paper is always better than CTS-DP.

The average error of the algorithm SW-ATS in the Diabetes dataset is 25.1% less than that of CTS-DP, and the decrease in the average error in the Steps dataset is 12.5%.

5.1.3. Privacy Protection Strength Calculation. In this paper, we use the filtering-based attack method proposed by Xiong et al. [32] to calculate the privacy protection strength. The privacy protection strength after the attack is

$$\varepsilon' = \frac{\left(R^{-1}P\right)^2}{2m}\varepsilon^2, \tag{13}$$

where R is a vector, representing the autocorrelation function of the noise sequence. P is the cross-correlation function of the original sequence and the noisy sequence, and ε represents the privacy budget. The smaller the ε', the

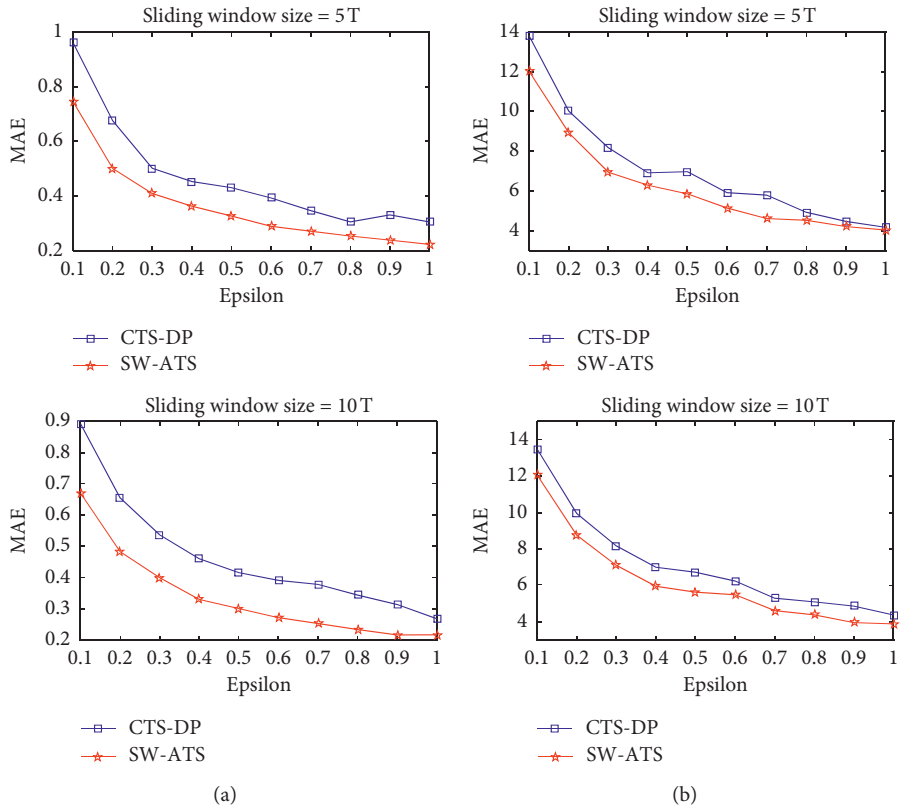

Figure 4: Utility comparison when epsilon changes. (a) MAE (diabetes dataset). (b) MAE (steps dataset).

higher the privacy protection strength. Figure 5 shows the comparison of the privacy protection strength of the two algorithms. It can be seen that on the two datasets, as the privacy budget continues to increase, the privacy protection strength of the two algorithms has a downward trend. However, the privacy protection strength of the SW-ATS algorithm is always higher than that of the CTS-DP. This proves that the periodic sensitivity proposed in this paper is effective and SW-ATS can protect the privacy of users to a greater extent from being leaked.

5.2. Experimental Conclusions. Each time CTS-DP releases data, it needs to process all the time-series data involved in the query. When new data arrive, CTS-DP needs to recalculate all the time-series data to be released and does a lot of unnecessary calculations. With the continuous growth of data flow, the calculation cost of the CTS-DP algorithm will become larger and larger and may cause the system to crash in extreme cases. The SW-ATS algorithm proposed in this paper introduces a sliding window mechanism on the basis of CTS-DP, which can both process the latest data and respond to queries with different time starting points and lengths. This reduces a lot of unnecessary calculations and greatly saves the system resources. The experimental results show that, under the sliding windows of different sizes, the error of SW-ATS is reduced by about 31% than that of CTS-DP, and under different privacy budgets, the error is reduced by about 19%.

6. Conclusions and Future Works

In this paper, we proposed a sliding window-based differential privacy publishing algorithm for autocorrelation time series, which is applied to the publishing of time-series data. We proved that SW-ATS satisfies ε-differential privacy. The experimental results show that the algorithm is significantly better than the comparison algorithm in the publishing of time-series data and can be applied to the publishing of dynamic data.

Although SW-ATS is effective, there are still some aspects to be improved in the future. One is that the periodic sensitivity depends on the sampling period of the timing data. Only when the time-series data have an obvious sampling period, SW-ATS can have a better protection effect. If the time-series data are sampled randomly, the privacy protection strength may not meet the expectations. At the same time, in order to calculate the periodic sensitivity, the length of the sliding window must be greater than three times the length of the sampling period. At present, the SW-ATS algorithm only considers the autocorrelation of single attribute and can only process the time-series data of a single attribute each time. The data of each attribute not only have self-correlation but also have a mutual correlation. It is the next research direction of this paper to consider the

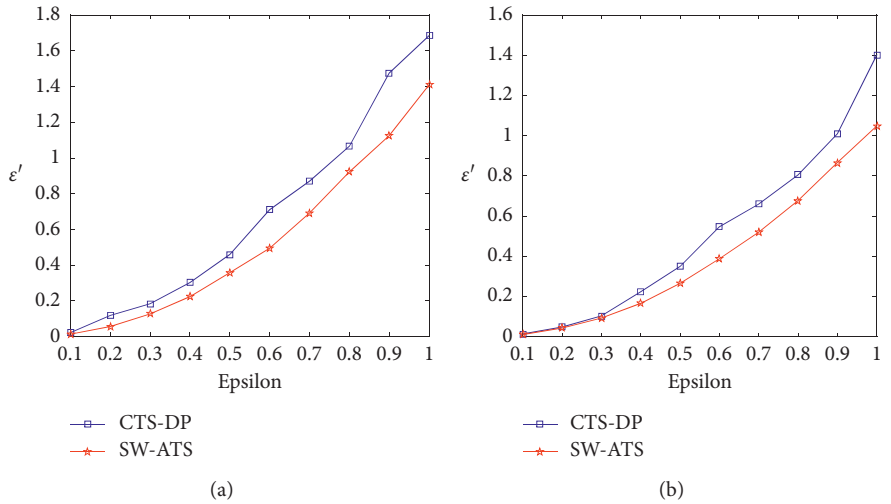

Figure 5: Comparison of privacy protection strength. (a) Diabetes dataset. (b) Steps dataset.

correlation between multiple attributes and publish multi-dimensional correlation time-series data.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors disclosed receipt of the following financial support for the research, authorship, and/or publication of this article. This work was supported by the National Natural Science Foundation of China (Grant no. 41971407), Major Technical Innovation Project of Hubei (Grant no. 2018AAA046), and Applied Basic Research Project of Wuhan (Grant no. 2017060201010162).

References

[1] L. Sweeney, "*K*-anonymity: a model for protecting privacy," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 5, 2002.

[2] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam, "L -diversity," *ACM Transactions on Knowledge Discovery from Data*, vol. 1, no. 1, p. 3, 2007.

[3] N. Li, T. Li, and S. Venkatasubramanian, "closeness: privacy beyond k-anonymity and l-diversity," in *Proceedings of the 2007 ICDE 2007. IEEE 23rd International Conference on IEEE*, Istanbul, Turkey, April 2007.

[4] C. Rupa, G. Srivastava, G. Reddy, and S. Bhattacharya, "Security and privacy of UAV data using blockchain technology," *Journal of Information Security and Applications*, vol. 8, p. 55, 2020.

[5] R. Kumar, "SP2F: a secured privacy-preserving framework for smart agricultural Unmanned Aerial Vehicles," *Computer Networks*, vol. 187, 2021.

[6] C. Dwork, "Differential privacy: a survey of results," in *Proceeding of the International Conference on Theory and Applications of Models of Computation*, Changsha, China, October 2008.

[7] F. Ma, S. Liu, X. Xiong et al., "Privacy protection based on local differential privacy for numerical sensitive data of wearable devices," *Journal of Computer Applications*, vol. 39, no. 7, pp. 1985–1990, 2019.

[8] X. J. Zhang and X. F. Meng, "Differential privacy in data publication and analysis," *Chinese Journal of Computers*, vol. 37, no. 4, pp. 927–949, 2014.

[9] C. Dwork, M. Naor, T. Pitassi et al., "Differential privacy under continual observation," in *Proceedings of the 42nd ACM Symposium on Theory of Computing*, Cambridge, MA, USA, May 2010.

[10] T.-H. Hubert Chan, "Private and continual release of statistics," *ACM Transactions on Information and System Security (TISSEC)*, vol. 14, p. 3, 2011.

[11] D. Kifer and A. Machanavajjhala, "No free lunch in data privacy," in *Proceedings of the ACM SIGMOD International Conference on Management of Data*, Athens, Greece, June 2011.

[12] T. Zhu, P. Xiong, G. Li et al., "Correlated differential privacy: hiding information in non-IID dataset," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 2, pp. 229–242, 2015.

[13] E. Shen and T. Yu, "Mining frequent graph patterns with differential privacy," in *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 545–553, Chicago, IL, USA, August 2013.

[14] X. Xiong, S. Liu, D. Li, Z. Cai, and X. Niu, "Real-time and private spatio-temporal data aggregation with local differential privacy," *Journal of Information Security and Applications*, vol. 55, Article ID 102633, 2020.

[15] V. Rastogi and S. Nath, "Differentially private aggregation of distributed time-series with transformation and encryption," in *Proceedings of the ACM SIGMOD International Conference on Management of Data*, Indianapolis, IN, USA, June 2010.

[16] H. Wang and Z. Xu, "CTS-DP: publishing correlated time-series data via differential privacy," *Knowledge-Based Systems*, vol. 122, no. 15, pp. 167–179, 2017.

[17] C. Rui and C. M. Benjamin, "Correlated network data publication via differential privacy," *Vldb Journal*, vol. 32, 2014.
[18] L. Cao, Y. Ou, and P. S. Yu, "Coupled behavior analysis with applications," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 8, pp. 1378–1392, 2012.
[19] S. Yin, "Coupled behavior analysis for capturing coupling relationships in group-based market manipulations," *Knowledge Discovery and Data Mining*, vol. 10, 2012.
[20] J. Zhang, Y. Xiang, Y. Wang et al., "Network traffic classification using correlation information," *IEEE Transactions on Parallel & Distributed Systems*, vol. 24, no. 1, pp. 104–117, 2012.
[21] Z.-H. Zhou, Y.-Y. Sun, and Y.-F. Li, "Multi-instance learning by treating instances as non-I.I.D samples," in *Proceedings of the 26th Annual International Conferenc Machine Learning (ICML)*, pp. 1249–1256, Long Beach, California, USA, August 2009.
[22] X. Xiao, "Differentially private data release: improving utility with wavelets and bayesian networks," *Web Technologies and Applications*, vol. 8709, pp. 25–35, 2014.
[23] X. Xiao, G. Wang, and J. Gehrke, "Differential privacy via wavelet transforms," *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, no. 8, 2009.
[24] W. Jiang, C. Xie, and Z. Zhang, "Wishart mechanism for differentially private principal components analysis," *Computer Ence*, vol. 9285, pp. 458–473, 2015.
[25] S. Zhou, K. Ligett, and L. Wasserman, "Differential privacy with compression," in *Proceedings of the IEEE International Symposium on Information Theory*, pp. 2718–2722, Seoul, Korea, July 2009.
[26] C. Dwork, "Differential privacy," in *Proceedings of the 33rd International Conference on Automata, Languages and Programming*, Venice, Italy, July 2006.
[27] D. Kifer and A. Machanavajjhala, "Pufferfish: a framework for mathematical privacy definitions," *ACM Transactions on Database Systems (TODS)*, vol. 22, 2014.
[28] C. Dwork, "Calibrating noise to sensitivity in private data analysis," *Lecture Notes in Computer Science*, vol. 3876, no. 8, pp. 265–284, 2012.
[29] M. S. Frank, "Privacy integrated queries: an extensible platform for privacy-preserving data analysis," *Communications of the ACM*, vol. 53, no. 9, pp. 89–97, 2010.
[30] F. Cai, S. Liang, and M. D. Rijke, "Time-sensitive personalized query auto-completion," in *Proceedings of the 2014 ACM International Conference on Information and Knowledge Management*, pp. 1599–1608, Shanghai, China, November 2014.
[31] A. Blum, K. Ligett, and A. Roth, "A learning theory approach to noninteractive database privacy," *Journal of the ACM*, vol. 60, p. 2, 2008.
[32] X. W. X. Zhengquan and S. Wanghao, "Privacy level evaluation of differential privacy for time-series based on filtering theory," *Journal on Communications*, vol. 38, no. 5, pp. 172–181, 2017.