Xiong and Zhu *Journal of Cloud Computing* (2022) 11:57
https://doi.org/10.1186/s13677-022-00332-3

Journal of Cloud Computing:
Advances, Systems and Applications

RESEARCH Open Access

Real-time trajectory privacy protection based on improved differential privacy method and deep learning model

Jing Xiong[1,2*] and Hong Zhu[1]

Abstract

Accurate and real-time trajectory data publishing plays an important role in providing users with the latest traffic and road condition information to help in rationally planning travel time and routes. However, the improper publishing of location information and reverse analysis and reasoning can easily leak users' personal information, which may threaten users' privacy and lives. Owing to the inclusion of differential privacy model noise, privacy protection introduces inaccuracies in data publishing and validity. To improve the accuracy and usability of published data, we propose a data publishing method based on deep learning and differential privacy models for securing spatiotemporal trajectory data publishing. The method divides the trajectory data into two-dimensional grid regions, counts the density of trajectories at grids, performs a top-down recursive division of regions, and formulates rules for privacy budget allocation from multiple perspectives as recurrence depth increases. Furthermore, the method integrates spatiotemporal sequence data according to temporal order. Subsequently, it extracts temporal and spatial features of the data by the temporal graph convolutional network model for budget matrix prediction, adds Laplace noise to the regions, and evaluates the effect of differential privacy protection with the original data to protect trajectory data privacy. Experiments demonstrate that under the premise of satisfying $\boldsymbol{\varepsilon}$-difference privacy, the query error and Jensen–Shannon divergence are smaller, the Kendall coefficient is more consistent, and the upper and lower limit values are more stable. Hence, the top-down division method achieves better results than those of the two traditional region division methods of the uniform grid and adaptive grid. The proposed method can be used to allocate the privacy budget more reasonably and achieve privacy protection of trajectories, which can be applied to a large amount of spatiotemporal trajectory data.

Keywords: Temporal graph convolutional network (T-GCN), Differential privacy, Privacy budget, Laplace noise

Introduction

Societal advancement and economic development are rapidly facilitating smart city planning. The widespread adoption of intelligent transportation systems, the internet of vehicles, and location-based service systems have increased the amount of data containing location information [1]. Owing to the desirable benefits of these modern applications, several "Internet+" travel methods have been proposed. Meanwhile, high-quality trajectory data publishing can deliver high-precision location-based services, provide users with accurate road condition information, and help users to schedule time and plan routes. However, inappropriate publishing of location information and inference of reverse analysis can easily breach users' privacy (for example, by revealing the specific location and movement trajectory), which may endanger lives and property.

The differential privacy model achieves differential privacy protection by adding random noise to published location data [2], and the model has been widely used in

*Correspondence: xiongjing3226@163.com

[2] Technology Promotion Division, CEPREI, Guangzhou, China
Full list of author information is available at the end of the article

the field of data publishing privacy protection. In fact, to protect users' privacy, adding differential privacy model noise introduces a deliberate discrepancy between published location data and actual statistical values [3]. The trajectory data publishing of regional locations aims for accuracy and efficiency without compromising users' privacy. Hence, it is essential to address the privacy protection problem of trajectory data publishing.

The meshing of 2D spatial regions and prediction of trajectory traffic are crucial in trajectory data distribution. Several models are currently used for trajectory traffic prediction, including the autoregressive integrated moving average model [4], support vector regression machine learning model [5], hidden Markov model [6], and partial neural network model [7]. However, these models ignore the spatial dependence and only consider the temporal dependence of the trajectory traffic, which cannot accurately predict the trajectory traffic in the region. To address the spatial dependence exclusion, a convolutional neural network has been introduced [8, 9]. The drawback of this attempt is that the convolutional neural network is most suitable for image processing rather than for complex spatial structures.

Recently, the rapid advances in graph neural networks have yielded desirable results in the fields of transportation and meteorology [10–12]. The graph convolutional neural network has a natural advantage in tasks involving complex spatial structures and can effectively capture the spatial dependence of regions while fully extracting spatial features. The grid division methods for 2D spatial regions are uniform grid (UG) [13], adaptive grid (AG) [14], and PrivTree divisions [15]. These methods are effective in dividing regions with a high time complexity but are unsuitable for on-time and effective publishing of trajectory data. In different regions, under-division and over-division problems may occur, and it is difficult to equalize noise errors, resulting in unreasonable privacy budget allocation, which inhibits differential privacy protection. To address these problems, we propose a differential privacy budget allocation method based on a temporal graph convolutional network (T-GCN) [16] to predict future privacy budget demands; we also propose a top-down division of regions for an accurate and rational allocation of the privacy budget. The main contributions of this study are summarized as follows:

(1) We formulate the grid division conditions for regions, use a recursive algorithm to divide similar regions by top-down search, and design the rules for privacy budget allocation from multiple perspectives as the recursive depth increases; this approach reasonably allocates privacy budget for each location.

(2) We use the T-GCN model to predict the privacy budget matrix for the future. Laplace noise is added in advance at the regional locations, and the trajectory noise data are published.

(3) We evaluate our approach using Yonsei pedestrian trajectory data and Citi Bike datasets. The results show that the T-GCN prediction is more stable and better than other deep learning models. In addition, compared with other region division methods, the method of a top-down division of regions for privacy budget allocation and the addition of Laplace noise is relatively more robust for each metric in the horizontal and vertical comparisons of privacy protection effects. Moreover, the method is effective for differential privacy protection.

Related work

The nonlinear and uncertain characteristics of trajectory data make it difficult to avoid the interference of random events, emphasizing the significance of studying trajectory data publishing. Dwork [2] proposed a differential privacy model that adds Laplace noise to the region location for differential privacy protection. The method has rigorous mathematical proof and has been widely used in the field of data distribution. The UG division method [13] uniformly divides the 2D space into numerous grids, counts the trajectories of regional grids, and adds noise to achieve differential privacy protection. This division approach is simple and efficient while easily achieving under-division in the region, weakening the availability of published data. Meanwhile, the AG division method [14] is based on a UG division for each region and adds noise to achieve differential privacy protection. This division method better reflects the influence of data distribution characteristics on the division structure but ignores the over-division problem caused by the complex spatial structure. Furthermore, the PrivTree division publishing method [15] introduces a controlled deviation for deciding whether to perform quadtree division, eliminating the restriction on the predefined quadtree division depth. Yan et al. [3] proposed the spatiotemporal–long short-term memory (ST-LSTM) model for predicting the structural hierarchy of region locations by adding Laplace noise to achieve differential privacy protection. However, the model does not precisely provide the differential privacy budget, and the granularity is insufficiently high when assigning the privacy budget.

The traditional hidden Markov model for spatiotemporal trajectory prediction [6] is based on spatiotemporal density clustering, which analyzes the correlation between time and space to predict different distributions of spatiotemporal series data. The traditional grid region division ignores the sparsity and denseness of the grid, the addition of noise increases the query error, and the traditional prediction method is only suitable for smooth data, which fails to capture the hidden nonlinear features in the spatiotemporal sequence [17] (the trajectory sections of spatiotemporal trajectory data have certain interactions and correlations, and methods based on deep learning can use such data more effectively to fully explore the hidden linear and nonlinear features). To publish the trajectory data effectively, we combine deep-learning and differential privacy protection models, design and implement a scheme for dividing the grid region of spatiotemporal sequence trajectory data, predict the privacy budget matrix, and add Laplace noise to each region location to achieve differential privacy protection. Overall, the original trajectory data are protected effectively.

Deep learning model

Overview

Since 2006, deep learning has become an emerging area in machine learning research. Deep learning has become a popular research area because of developments in computing power. With rapid advances in deep learning, deep neural networks demonstrate excellent performance in discovering intricate structures in high-dimensional data and outperform traditional methods [18]. In addition to beating records in image recognition [19, 20] and speech recognition [21], deep neural networks have beaten other machine-learning techniques at predicting the activity of potential drug molecules [22]. Deep neural networks can now effectively capture the dynamic features of data and achieve excellent prediction results. For example, convolutional neural networks can be used for image classification, image segmentation, and object detection. Typical convolutional neural networks include ImageNet [23], ResNet [24], U-Net [25], and R-CNN [26]. Recurrent neural networks can identify relationships within time series data and predict future occurrences; LSTM [27] and gated recurrent unit (GRU) [28] are popular examples of recurrent neural networks. In the field of meteorology, convolutional and recurrent neural networks have been widely used to predict rainfall, wind speed, and radar echoes [29–31], and examples of such networks are convolutional LSTM (ConvLSTM) [29] and ST-LSTM [30]. In the field of transportation, researchers [11, 16, 32, 33] used graph networks and recurrent neural networks to predict traffic flow, and examples are T-GCN [16] and ST-GCN [11]. These examples demonstrate that neural networks have been used to achieve desired progress in prediction and classification tasks in several fields.

T-GCN

The T-GCN model [16] comprises the graph convolutional network (GCN) [34] and GRU [28]. T-GCN essentially captures the spatial dependence of the topology using GCN and captures the temporal dependence of the trajectory data using GRU.

Definition 1 *Given a network graph G, we use an unweighted graph* $G=(V,E)$ *to describe the topological structure of the network graph.* $V=\{v_1,v_2,\cdots,v_{N\times N}\}$ *is the node at the grid region locations* $\mathrm{V}=\{\mathrm{v}_1,\mathrm{v}_2,\cdots,\mathrm{v}_{\mathrm{N}\times\mathrm{N}}\}$, *where* $N\times N$ *denotes the number of grid region locations; E is a set of edges. The adjacency matrix M represents the connection between grid region locations,* $M\in R^{N^2\times N^2}$*; as the adjacency matrix represents the relationship between nodes adjacent to each other, the only elements are 0 and 1. The element is 0 if no link exists between nodes but 1 if a link exists.*

Definition 2 *Given a feature matrix X, we regard the trajectory information on the network as the attribute feature of the node in the network, expressed as*$X\in R^{N^2\times P}$, *where P represents the number of node attribute features (the length of the historical time series), and*$X_t\in R^{N^2\times i}$ *represents the trajectory density on each location at time i.*

Thus, predicting the spatiotemporal trajectory can be regarded as learning a mapping function F given a graph G and a feature matrix X. Then, the trajectory information is obtained for the next T moments, as shown in the following equation:

$$[X_{t+1},\cdots,X_{t+T}]=F(G;(X_{t-n},\cdots,X_{t-1},X_t)), \quad (1)$$

where n is the length of the historical time series, and T is the length of the predicted time series step.

The internal structure of the T-GCN model is fully illustrated in Fig. 1a: the historical trajectory data are inputted into the model while the data of the future moment are outputted by the GCN model [34] to obtain spatial features as well as the GRU model [35] to obtain temporal features. Figure 1b depicts the graph convolution of the grid location information by the GCN model. Assuming that red grid is the central location, the GCN model can obtain the topological relationship between the central location and its surrounding green grids, encode the topological structure of the grid network and the attributes on the grids, and then obtain spatial dependence. Figure 1c illustrates the structure of a GRU cell, which can extract the temporal feature of a grid region at time t by taking the hidden status at time $t-1$. In this case, the GRU model captures the current trajectory information while retaining the changing trend of historical traffic information; the model also

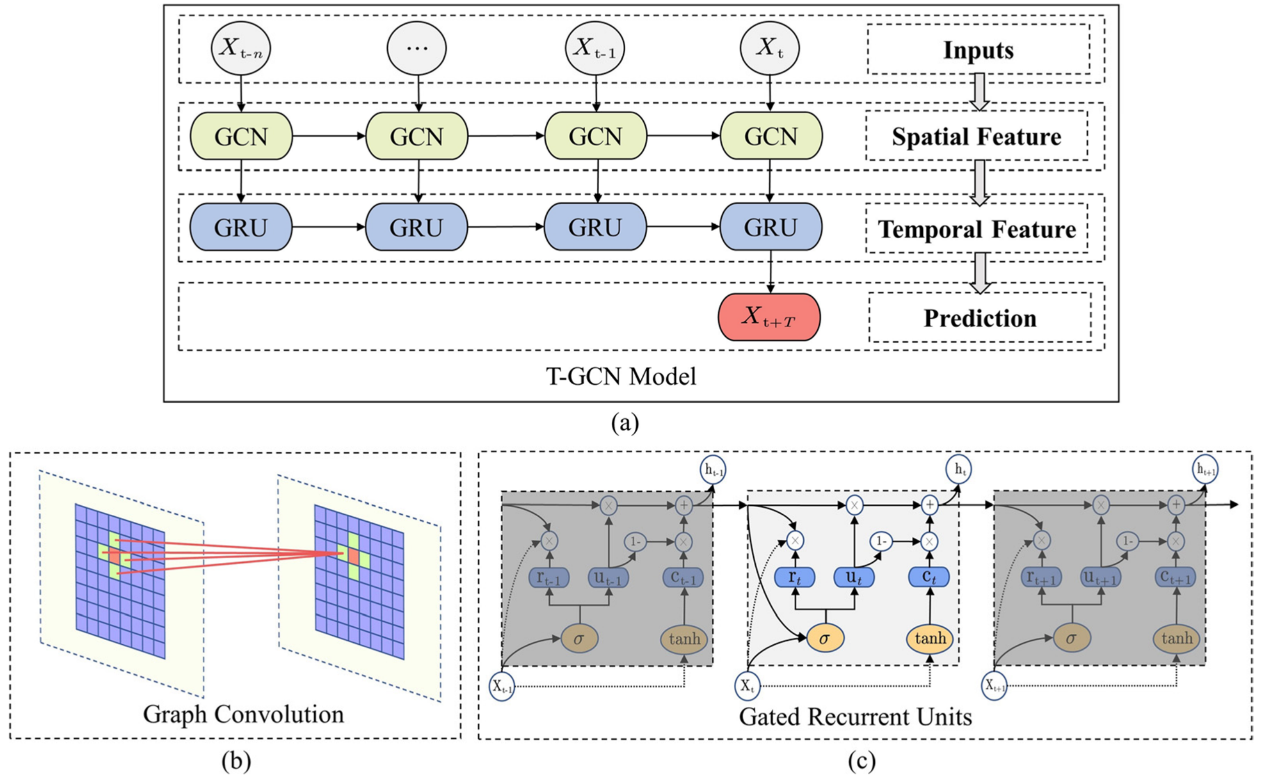

Fig. 1 **a** Internal structure of spatiotemporal prediction. **b** Spatial feature is obtained by obtaining the topological relationship between the red grid location and the surrounding green grids. **c** Temporal feature is obtained by obtaining the grid location status at time *t* and taking the hidden status at time *t*–1 and the current location information as inputs

captures temporal dependence. The model equations are expressed as follows, Eq. (2) represents the graph convolution process, and Eqs. (3), (4), (5), and (6) represent the GRU cell process.

$$F(M,X) = \sigma\left(\hat{M}Relu\left(\hat{M}XW_0\right)W_1\right), \tag{2}$$

$$u_t = \sigma(W_u[F(M,X_t),h_{t-1}] + b_u), \tag{3}$$

$$r_t = \sigma(W_r[F(M,X_t),h_{t-1}] + b_r), \tag{4}$$

$$c_t = \tanh(W_c[F(M,X_t),(r_t * h_{t-1})] + b_c), \tag{5}$$

$$h_t = u_t * h_{t-1} + (1-u_t) * c_t, \tag{6}$$

where $F(\cdot)$ denotes the graph convolution process, X is a feature matrix, M is an adjacency matrix, and $\hat{M} = M + I_{N\times N}$ is a Laplace matrix. $\hat{M} = \tilde{D}^{-\frac{1}{2}}\tilde{M}\tilde{D}^{-\frac{1}{2}}$ denotes the Laplace matrix normalization. $\tilde{D}$ is a degree matrix, where $\tilde{D} = \sum_{j=1}^{N\times N}\tilde{M}_{ij}, i = 1,\cdots,N\times N$. W_0 and W_1 denote the weight matrix. h_{t-1} is the output at moment $t-1$, μ_t is an update gate, and r_t is a reset gate at moment *t*. $\sigma(\cdot)$, $Relu(\cdot)$, and $tanh(\cdot)$ represent the activation function.

Summarily, the T-GCN model can capture the complex topology of grid regions through GCNs, and GRU cells can subsequently capture the dynamic changes in trajectory information. While the T-GCN model uses a GCN model with two layers to learn the spatial features of the grid region locations, it uses the GRU model to learn the temporal features of the trajectory data. Ultimately, the T-GCN model can sufficiently learn the spatial and temporal dependence to achieve trajectory prediction.

Differential privacy

The differential privacy model is effective for protecting privacy. The model primarily adds noise to the original data to prevent differential attacks, making it impossible for an attacker to identify specific samples in the dataset. If adding trajectory data to a location changes the information of that location, the distribution of the data would only change slightly according to the dynamic change in the trajectory data; thus, the technique is suitable for differential privacy protection.

Definition 3 *ε-differential privacy* [2]. *For two adjacent datasets $T_1$and $T_2$with a Hamming distance of 1 (only one different record exists for both), for any output set S of algorithm A, if the probability Pr satisfies*

$$\Pr[A(T_1) \in S] \leq e^{\varepsilon}\Pr[A(T_2) \in S], \tag{7}$$

the algorithm A satisfies ε-differential privacy, where ε is the privacy budget, the smaller its value, the more noise is added by algorithm A, the better the privacy protection.

Definition 4 *Sensitivity* [36, 37]. *Given a query function $f(\cdot)$ with sensitivity to the L_1-paradigm maximum distance between datasets $T_1$and T_2, the sensitivity is defined as:*

$$\Delta f = \max\left\|f(T_1) - f(T_2)\right\|_1. \tag{8}$$

Definition 5 *Laplace mechanism* [37]. *The Laplace mechanism adds independent noise to the result of the query function $f(\cdot)$ to achieve differential privacy protection. $f(T)$ represents the query result of the dataset T, and the addition of the noise ϕ can be denoted as $A(T) = f(T) + \phi$, where ϕ is an independent identically distributed random variable obeying the Laplace distribution with a probability density function of $\Pr[\varphi = x] = \frac{1}{2b}e^{-\frac{|x-\mu|}{b}}; b = \frac{\Delta f}{\varepsilon}$, and μ is the location parameter, where $\mu = 0$ in this study.*

Definition 6 *Serial combination property* [38]. *Given a set of randomized algorithms $\{A_1, \cdots, A_n\}$, each algorithm satisfies ε-differential privacy for the dataset T, and the algorithms can achieve $\sum_{i=1}^{n}\varepsilon_i$-differential privacy for the dataset T.*

Definition 7 *Parallel combination property* [38]. *Given a set of randomized algorithms $\{A_1, \cdots, A_n\}$, each algorithm satisfies ε-differential privacy for the sub-data $T = \{T_1, \cdots, T_n\}$ in dataset T separately, and the algorithms can achieve max$\{\varepsilon_i\}$-differential privacy for dataset T.*

Differential privacy data publishing based on UG and AG

Definition 8 *Count matrix. Initialize the $N \times N$ 2D grid region and construct a matrix C to represent the counting of the region. The matrix element $cnt_{rc}(r, c = 1, \cdots, N)$ indicates the cumulative number of trajectories. The matrix C is expressed as:*

$$C = \begin{bmatrix} cnt_{11} & cnt_{12} & \cdots & cnt_{1N} \\ cnt_{21} & cnt_{22} & \cdots & cnt_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ cnt_{N1} & cnt_{N2} & \cdots & cnt_{NN} \end{bmatrix}. \tag{9}$$

Model design by UG

UG [13] is a uniform division of the region, the method is given a differential privacy budget, and Laplace noise is uniformly added after releasing the data to achieve differential privacy protection. The addition of noise through UG division is the most rudimentary method, which essentially ignores future changes in trajectory data and only needs to publish noisy data before real data. The process is illustrated in Fig. 2 and outlined as follows:

(1) The trajectory data T_d, T_{d+1} on day d and day $d+1$ are counted.

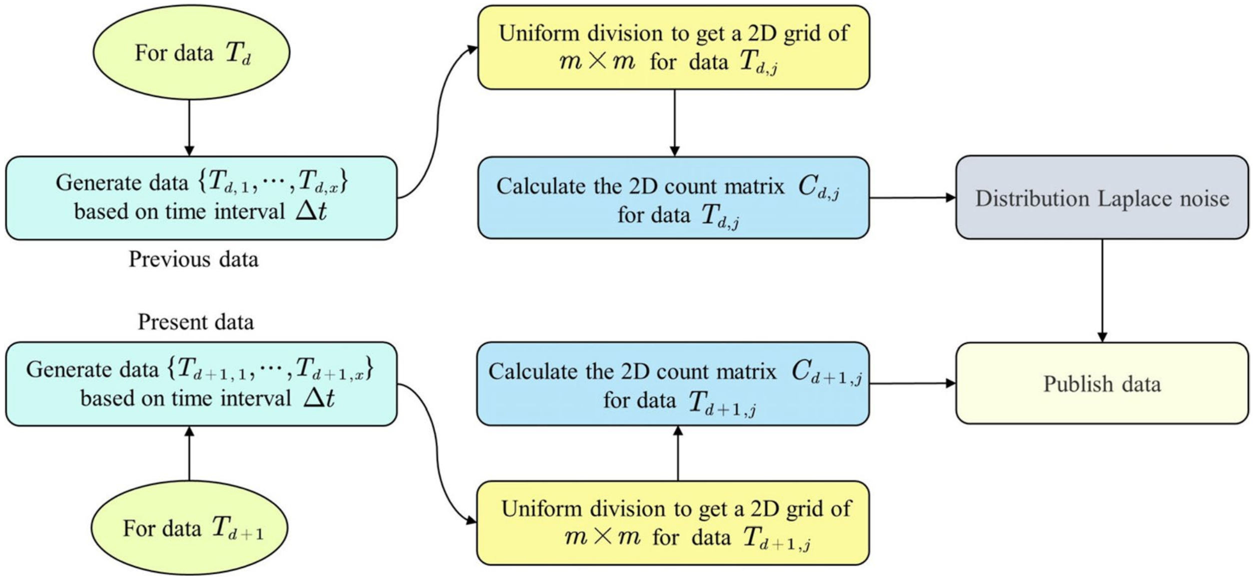
For data T_d
Generate data $\{T_{d,1}, \cdots, T_{d,x}\}$ based on time interval Δt
Previous data
Uniform division to get a 2D grid of $m \times m$ for data $T_{d,j}$
Calculate the 2D count matrix $C_{d,j}$ for data $T_{d,j}$
Distribution Laplace noise
Present data
Generate data $\{T_{d+1,1}, \cdots, T_{d+1,x}\}$ based on time interval Δt
Calculate the 2D count matrix $C_{d+1,j}$ for data $T_{d+1,j}$
Publish data
For data T_{d+1}
Uniform division to get a 2D grid of $m \times m$ for data $T_{d+1,j}$

Fig. 2 Differential privacy data publishing process in the UG division

(2) Data T_d, T_{d+1} are generated into a fine-grained collection $\{T_{d,\,j},\ T_{d+1,\,j} | j=1,\ \cdots,\ x\}$ according to the time interval Δt.
(3) The fine-grained data $T_{d,\,j}$, $T_{d+1,\,j}$ are divided uniformly into a 2D grid of $m \times m$, where m is the granularity of division.
(4) The count of the 2D grid region is determined to get the count matrix collection $\{C_{d,\,j},\ C_{d+1,\,j} | j=1,\ \cdots,\ x\}$.
(5) Laplace noise is distributed for each region location, i.e., $cnt_{rc}^{*} = Lap\left(\frac{S}{\varepsilon}\right)$, where ε is the privacy budget, and S is $max\{cnt_{rc}\}$.
(6) The count matrix $C_{d+1,\,j}$ of the present data is calculated, and Laplace noise is added.
(7) The data are published.

Model design by AG

AG [14] is the adaptive division of the region, and the trajectory data publishing idea of this method contains two layers. The first layer divides the 2D region into a matrix of an $m_1 \times m_1$ size and adds Laplace noise according to the differential privacy budget $\alpha\varepsilon(0 < \alpha < 1)$ to get the noise v at each location. The second layer divides the grid according to the noise v by adaptively selecting the size of the region and then adaptively selecting a new region of $m_2 \times m_2$ by adding noise again to get noise u at each location; reference [14] gives the method of adaptively selecting the region. Here, given a cell with a noisy count of N', to minimize the errors, this cell should be partitioned into $m_2 \times m_2$ cells, where m_2 is computed as follows:

$$m_2 = \left[\sqrt{\frac{N'(1-\alpha)\varepsilon}{c/2}}\right], \tag{10}$$

where $[\cdot]$ represents rounding down, $(1-\alpha)\varepsilon$ is the remaining privacy budget for obtaining noisy counts for grids, and setting $c = 10$.

The original noise v and the noise u of the redivided region are then weighted to obtain a more precise noise, making the deviation of the location noise error smaller. The weighted average formula is defined as:

$$v' = \frac{\alpha^2 m_2^2}{(1-\alpha)^2+\alpha^2 m_2^2} + \frac{(1-\alpha)^2}{(1-\alpha)^2+\alpha^2 m_2^2}\sum u_{i,j}. \tag{11}$$

Finally, the noise is uniformly distributed to all grids as the final noise at each region location. The uniform distribution noise formula is defined as:

$$u'_{i,j} = u_{i,j} + \left(v' - \sum u_{i,j}\right). \tag{12}$$

The difference between both methods is that the UG method considers all regions equally, whereas the AG method divides regions adaptively and adds Laplace noise. As the method requires the trajectory data of future moments to add noise, the data traversal is prohibited when publishing data, and the trajectory data published in the previous moment are used as the reference data of future moments to obtain publishing noise data. The process is illustrated in Fig. 3 and explained as follows:

(1) The trajectory data T_d, T_{d+1} on day d and day $d+1$ are counted.
(2) Data T_d, T_{d+1} are generated into a fine-grained collection $\{T_{d,\,j},\ T_{d+1,\,j} | j=1,\ \cdots,\ x\}$ according to the time interval Δt.

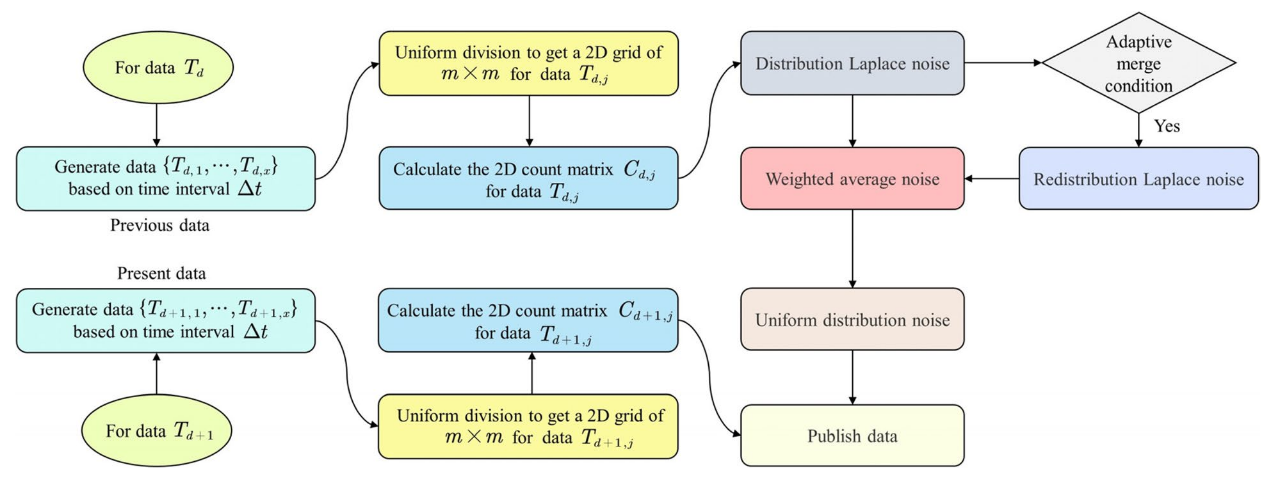
For data T_d
Generate data $\{T_{d,1},\cdots,T_{d,x}\}$ based on time interval Δt
Previous data
Uniform division to get a 2D grid of $m \times m$ for data $T_{d,j}$
Calculate the 2D count matrix $C_{d,j}$ for data $T_{d,j}$
Distribution Laplace noise
Adaptive merge condition
Yes
Weighted average noise
Redistribution Laplace noise
Present data
Generate data $\{T_{d+1,1},\cdots,T_{d+1,x}\}$ based on time interval Δt
For data T_{d+1}
Calculate the 2D count matrix $C_{d+1,j}$ for data $T_{d+1,j}$
Uniform division to get a 2D grid of $m \times m$ for data $T_{d+1,j}$
Uniform distribution noise
Publish data

Fig. 3 Differential privacy data publishing process in the AG division

(3) The fine-grained data $T_{d,\,j}$, $T_{d+1,\,j}$ are divided uniformly into a 2D grid of $m_1 \times m_1$; m_1 is the granularity of division.

(4) The count of the 2D grid region is calculated to get the count matrix collection $\{C_{d,\,j}, C_{d+1,\,j} | j=1, \cdots, x\}$.

(5) Laplace noise is distributed for each region location, i.e., $cnt_{rc}^{v} = Lap\left(\frac{S}{\varepsilon}\right)$, where ε is the privacy budget, and is $max\{cnt_{rc}\}$.

(6) If the adaptive division condition is satisfied, the Laplace noise is redistributed for the new range of $m_2 \times m_2$, i.e., $cnt_{rc}^{u} = Lap\left(\frac{S}{\varepsilon}\right)$.

(7) cnt_{rc}^{v} and cnt_{rc}^{u} are weighed to distribute the noise uniformly to all grid locations.

(8) The count matrix $C_{d+1,\,j}$ of the present data is calculated, and Laplace noise is added.

(9) The data are published.

Differential privacy data publishing based on deep learning and top-down approach

Definition 9 *Density matrix. Initialize the $N \times N$ 2D grid region and construct a matrix D to represent the density of the region. The matrix element $\rho_{rc}(r, c=1, \cdots, N)$ indicates the density of the trajectories, where $\rho_{rc} = \frac{cnt_{rc}}{area_{rc}}$, $area_{rc}$ represents the area of the grid. The matrix D is expressed as:*

$$D = \begin{bmatrix} \rho_{11} & \rho_{12} & \cdots & \rho_{1N} \\ \rho_{21} & \rho_{22} & \cdots & \rho_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ \rho_{N1} & \rho_{N2} & \cdots & \rho_{NN} \end{bmatrix}. \tag{13}$$

Definition 10 *Regional division condition. In a 2D region, design a rule such that if the density of trajectories in the region is similar, then the region is divided into a sub-region, and privacy protection is provided for each sub-region. The rule is defined as:*

$$\frac{\sum_{r=1}^{N}\sum_{c=1}^{N}|\rho_{rc}-\overline{\rho}|}{\sum_{r=1}^{N}\sum_{c=1}^{N}\rho_{rc}} \le \varepsilon, \tag{14}$$

where $\overline{\rho}$ is the average density of the grid region, and ξ is the division factor; the smaller the factor, the stricter the condition for division.

Definition 11 *Privacy budget matrix. Initialize the $N \times N$ 2D grid region and construct a matrix E to represent the result of the privacy budget allocation. The matrix element $e_{rc}(r, c=1, \cdots, N)$ indicates the size of the privacy budget allocation for the grid. e_{rc} has an initial value of 0. The matrix E is expressed as:*

$$E = \begin{bmatrix} e_{11} & e_{12} & \cdots & e_{1N} \\ e_{21} & e_{22} & \cdots & e_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ e_{N1} & e_{N2} & \cdots & e_{NN} \end{bmatrix}. \tag{15}$$

Privacy budget allocation strategies

In this study, we use the top-down division of regions to allocate the privacy budget; the range is divided into two parts for the sub-regions to be searched each time, and the depth of recursion is $h(N=2^h)$. If the searched region satisfies the division condition, all grid locations in the region would be considered as a single location, and the allocated privacy budget would be adjusted. On the contrary, if the division condition is not satisfied, a new privacy budget would be reallocated to the region.

According to the definition of differential privacy, the smaller the value of ε is, the better the privacy protection of the region, and for dense regions, a smaller value of ε is expected to be allocated. On the contrary, for sparse regions, a larger value of ε is expected to be allocated. Therefore, according to the total differential privacy budget ε, the recursion depth is $i \in \{1, \cdots, h\}$, and ε is divided into an increasing sequence, which is defined as:

$$\varepsilon_i = \frac{a_i}{\sum_{i=1}^{h} a_i}\varepsilon, 1 \le i \le h, \tag{16}$$

where $a_i = \frac{i(i+1)}{2}$ is a sequence of progressively increasing tolerances.

Divisible regional privacy budget allocation

If the region of the top-down recursive search satisfies the division condition, allocates the privacy budget to each grid location according to the recursion depth of the region. When the recursion depth is 1, that is $e_{rc}^{<i>} = \varepsilon_i, i = 1$, the privacy budget is allocated to ε_1 if the initial grid region is divisible. When the recursion depth is not 1, the region with a smaller search range follows these rules.

Rule 1: From divisible region to divisible region. When the region of recursive depth $i-1$ satisfies the division condition, the region of depth i also satisfies the condition. Thus, these regions are more similar. In this case, the objective is to gradually increase the minimum privacy budget policy to allocate the privacy budget to this region and achieve differential privacy. The privacy budget is adjusted as follows:

$$e_{rc}^{<i>} = e_{rc}^{<i-1>} + \varepsilon_i, 1 < i \le h. \tag{17}$$

Rule 2: From indivisible region to divisible region. When the region of recursive depth $i-1$ does not satisfy the division condition but the region of depth i does, it indicates that as the search range decreases, certain regions have better similarity, requiring the allocation of a larger privacy budget. In the recursive search regions, we use the *MemoryState* region to recall the strategy of whether the region searched at depth $i-1$ satisfies the division condition. If the region does not satisfy the division condition, *MemoryState* is recorded as *True*; otherwise, it is recorded as *False*. Hence, the differential privacy budget summation of depth $j=1, \cdots, i-1$ can be found, to which the privacy budget of depth i is added. Such strategy can rapidly adjust the privacy budget allocation in the transition of region properties to achieve differential privacy. The adjusted privacy budget is as follows:

$$e_{rc}^{<i>} = \sum_{j=1}^{i-1} \varepsilon_j + \varepsilon_i, 1 < i \leq h. \tag{18}$$

Indivisible regional privacy budget allocation

If the region of the top-down recursive search does not satisfy the division condition, allocates the privacy budget to each location according to the recursion depth of the region. When the recursion depth is 1, that is $e_{rc}^{<i>} = \varepsilon_{h+1-i}, i = 1$, the privacy budget is allocated to ε_h if the initial grid region is indivisible. When the recursion depth is not 1, the region with a smaller search range follows these rules.

Rule 3: From divisible region to indivisible region. When the region of recursive depth $i-1$ satisfies the division condition but the region with depth i does not, it indicates that as the search range decreases, certain regions have worse similarity, requiring a reallocation of a smaller privacy budget to achieve differential privacy. The privacy budget reallocation is as follows:

$$e_{rc}^{<i>} = \varepsilon_{h+1-i}, 1 < i \leq h. \tag{19}$$

Rule 4: From indivisible region to divisible region. When the region of recursive depth $i-1$ does not satisfy the division condition and the region of depth i satisfies the condition, it indicates that the similarity of both regions is worse. In this case, the objective is to gradually reduce the maximum privacy budget to reallocate the privacy budget to this region for differential privacy. The privacy budget reallocation is as follows:

$$e_{rc}^{<i>} = \varepsilon_{h+1-i}, 1 < i \leq h. \tag{20}$$

Regional trackless privacy budget allocation

Rule 5: Trackless region. If a region of time interval Δt does not produce trajectory data, the region is a completely sparse region, and the privacy budget of the whole region is allocated as the total differential privacy budget ε.

Differential privacy availability

Data are published primarily to provide users with the service of querying the number of regional statistics. Predicting the privacy budget matrix for future moments enables the publishing of noise data in the query system in advance, avoiding the delay of traditional methods. When a user submits a query, the query range is noted as $Q_{m, n}$, and three query cases exist.

(1) When the query range $Q_{m, n}$ is a complete region in the top-down division process. According to Definition 6, the region may also have sub-regions with a higher similarity, and the privacy budget is adjusted by searching the sub-regions with a smaller range, which is similar to a gradual accumulation of privacy budget. The privacy budget of the sub-regions location is $\varepsilon = \sum_{i=1}^{h} \varepsilon_i$; thus, the region is protected by the differential privacy with the strength of $\varepsilon_{Q_{m,n}} = \sum_{i=1}^{h} \varepsilon_i$.

(2) When the query range $Q_{m, n}$ is multiple sub-regions of the same depth in top-down division. This means that the range is the intersection of multiple regions. According to Definition 7, each set of division algorithms satisfies ε_i-differential privacy protection for the corresponding region location at depth i. In particular, the privacy budget of the divisible region is $\varepsilon = \sum_{j=1}^{i-1} \varepsilon_j + \varepsilon_i$, and the privacy budget of the indivisible region is $\varepsilon = \varepsilon_{h+1-i}$. The region is protected by differential privacy with the strength of $\varepsilon_{Q_{m,n}} = \sum_{j=1}^{i} \varepsilon_j + \varepsilon_{h+1-i} > \max\{\varepsilon_i\}$.

(3) When the query range $Q_{m, n}$ is a trackless region. According to Section 5.1.3, the privacy budget for this region is allocated as the total differential privacy budget ε_1. As the trackless region does not disclose the user's privacy, it is unnecessary to add noise. When the user queries multiple time intervals $\{\Delta t_1, \cdots, \Delta t_x\}$ of the same range $Q_{m, n}$, the availability of privacy protection at other times $\Delta(t+s)$ is reduced if noise is added in this region, where $s=1, \cdots, x-t$. Therefore, the region at that moment can be used without adding Laplace noise.

Model design by top-down

Combining privacy budget matrix prediction and data publishing methods, given a granularity size m, the top-down recursive division of the region and the deep learning model predicts the privacy budget matrix for future moments. According to the privacy budget allocation strategy for divisible and non-divisible regions, Laplace noise is added; to achieve differential privacy protection more effectively, no noise is added for trackless regions.

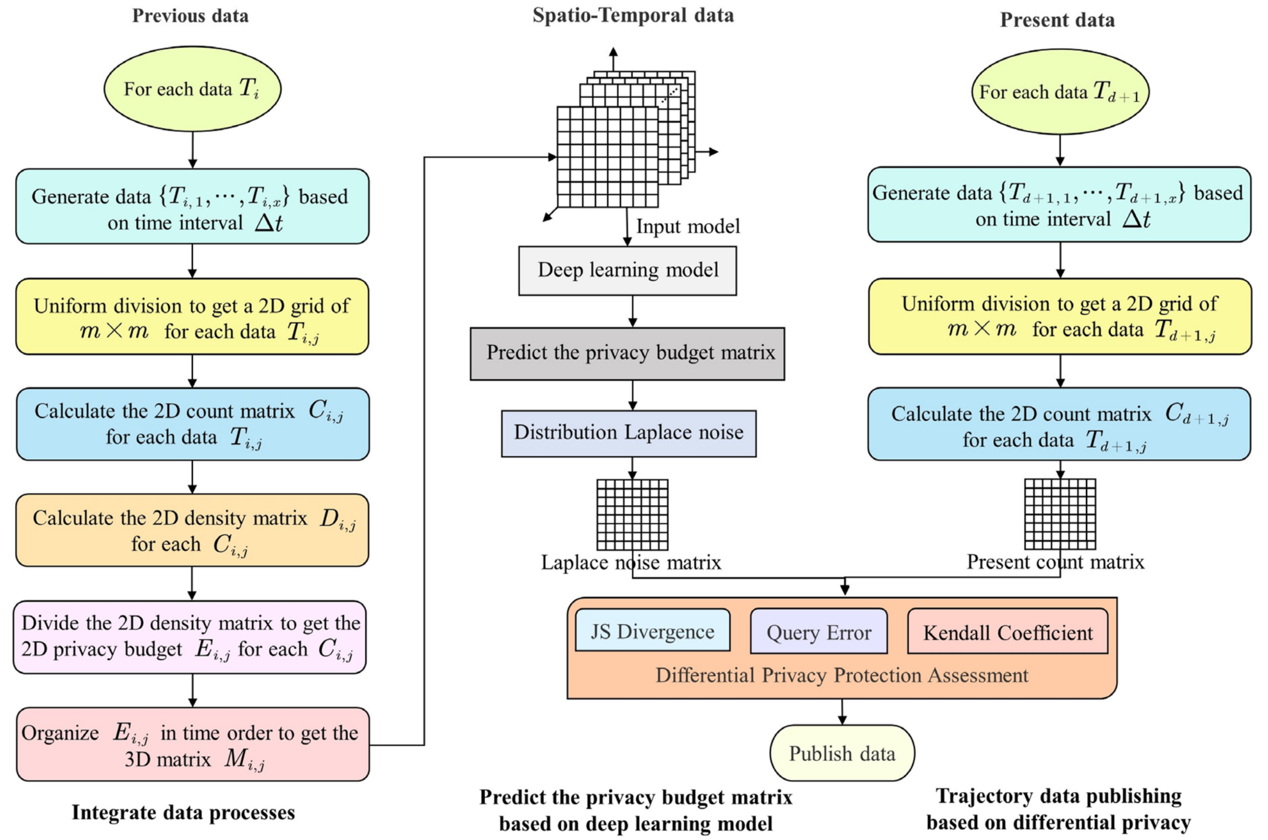
Previous data
For each data T_i
Generate data $\{T_{i,1},\cdots,T_{i,x}\}$ based on time interval Δt
Uniform division to get a 2D grid of $m\times m$ for each data $T_{i,j}$
Calculate the 2D count matrix $C_{i,j}$ for each data $T_{i,j}$
Calculate the 2D density matrix $D_{i,j}$ for each $C_{i,j}$
Divide the 2D density matrix to get the 2D privacy budget $E_{i,j}$ for each $C_{i,j}$
Organize $E_{i,j}$ in time order to get the 3D matrix $M_{i,j}$
Integrate data processes
Spatio-Temporal data
Input model
Deep learning model
Predict the privacy budget matrix
Distribution Laplace noise
Laplace noise matrix
Predict the privacy budget matrix based on deep learning model
Present data
For each data T_{d+1}
Generate data $\{T_{d+1,1},\cdots,T_{d+1,x}\}$ based on time interval Δt
Uniform division to get a 2D grid of $m\times m$ for each data $T_{d+1,j}$
Calculate the 2D count matrix $C_{d+1,j}$ for each data $T_{d+1,j}$
Present count matrix
JS Divergence
Query Error
Kendall Coefficient
Differential Privacy Protection Assessment
Publish data
Trajectory data publishing based on differential privacy

Fig. 4 Differential privacy data publishing process by top-down division and deep learning

Figure 4 illustrates the main process of the proposed privacy budget matrix prediction with the top-down division to achieve differential privacy for data publishing. The process is detailed as follows:

1) *Integration of data processes*

(1) The trajectory data are processed in days to obtain the collection $\{T_i \mid i=1, \cdots, d\}$.
(2) Data T_i are generated into a fine-grained collection $\{T_{i,j} \mid i=1, \cdots, d, j=1, \cdots, x\}$ according to the time interval Δt.
(3) The fine-grained data $T_{i,j}$ are divided uniformly into a 2D grid of $m \times m$; m is the granularity of division.
(4) The count of the 2D grid region is determined to obtain the count matrix collection $\{C_{i,j} \mid i=1, \cdots, d, j=1, \cdots, x\}$.
(5) The density of the 2D grid region is calculated to obtain the density matrix collection $\{D_{i,j} \mid i=1, \cdots, d, j=1, \cdots, x\}$.
(6) Regions are divided by the top-down division, and privacy budgets are allocated to get the privacy budget matrix collection $\{E_{i,j} \mid i=1, \cdots, d, j=1, \cdots, x\}$.
(7) The privacy budget matrix collection is organized into a three-dimensional matrix collection $\{M_{i,j} \mid i=1, \cdots, d, j=1, \cdots, x\}$ in chronological order.

2) *Prediction of the privacy budget matrix based on deep learning*

(1) The spatiotemporal data are inputted into deep learning models and trained iteratively.
(2) The privacy budget matrix $M_{d+1,j}$ for day $d+1$ is predicted.

3) *Trajectory data publishing based on differential privacy*

(1) Laplace noise is distributed for each region location, i.e., $cnt_{rc}^{*} = Lap\left(\frac{S}{\varepsilon_k}\right)$, where $k \in \{1, h\}$; ε_k is the privacy budget, and S is $max\{cnt_{rc}\}$.

(2) The count matrix $C_{d+1,j}$ of the present data is calculated, and Laplace noise is added.
(3) The effectiveness of differential privacy protection is evaluated, and the data are published.

Experiments

Data description

We evaluated the prediction performance of the deep learning model on two datasets: the Yonsei dataset and the Citi Bike dataset, as both datasets are related to trajectory density. Without loss of generality, we used the trajectory density as trajectory information in the experiments. Table 1 presents the details of both datasets.

In the experiments, the spatiotemporal trajectory data of the dimension [364, 24, 64 × 64, 1] were obtained by processing the original data, where 364 denotes the time span, 24 denotes the frequency of data publishing, 64 × 64 denotes the number of nodes, and 1 denotes the dimension of node features. The adjacency matrix was constructed according to whether an edge exists at each region location, and its dimension was [64 × 64, 64 × 64].

Evaluation metrics

Deep learning evaluation metrics

To evaluate the effectiveness of the model in privacy budget prediction, we used two practical criteria, namely regression and classification. The regression evaluation metrics were mean absolute error (MAE) and mean squared error (MSE). The classification evaluation metric was binary cross entropy (BCE). For the regression, let the true value of the privacy budget matrix be y^{true} and the predicted value be y^{pred}. For the classification, there are C categories, the true value of the privacy budget matrix is y_c^{true}, and the predicted value is y_c^{pred}. Let the sample size be N, then the formula for each index is expressed as:

$$\mathrm{MAE} = \frac{1}{N}\sum_{n=1}^{N}\left|y_n^{true} - y_n^{pred}\right|, \tag{21}$$

$$MSE = \frac{1}{N}\sum_{n=1}^{N}\left(y_n^{true} - y_n^{pred}\right)^2, \tag{22}$$

$$BCE = \frac{1}{N}\sum_{n=1}^{N}\sum_{c=1}^{C} y_{c,n}^{true}\log\frac{e_{c,n}^{pred}}{\sum_{k=1}^{C} y_{k,n}^{pred}}. \tag{23}$$

The smaller the values of MAE and MSE, the more accurate the prediction. Additionally, the smaller the value of BCE, the better the prediction effect.

Differential privacy evaluation metrics

To evaluate the effectiveness of differential privacy, we used three measures of published data, including query error (QE), Jensen–Shannon (JS) divergence, and Kendall coefficient.

(1) Query error [37]. Given a query function $f(\cdot)$, $f(T)$ denotes the correct result for a query region T, where $|T|$ denotes the size of the query region. Let $f\left(\tilde{T}\right)$ denotes the noisy query result, and then the query error is defined as follows:

$$QE(f) = \frac{\left|f(T) - f\left(\tilde{T}\right)\right|}{\max\{f(T), 0.01|T|\}}. \tag{24}$$

(2) JS divergence [39]. Given probability distribution functions P, Q for original publishing data and noise-added publishing data, respectively, $M = \frac{P+Q}{2}$. p_i, q_i are the probabilities of the distribution functions P, Q; then the JS divergence is defined as follows:

$$JS(P|Q) = \frac{1}{2}\sum_{i=1}^{n} p_i\log\frac{p_i}{p_i+q_i} + \frac{1}{2}\sum_{i=1}^{n} q_i\log\frac{q_i}{p_i+q_i} + \log 2. \tag{25}$$

(3) Kendall coefficient [40]. Given the original data X and noise-added data Y, we consider X and Y to be independently and identically distributed. When the observed sample $(x_i - x_j)(y_i - y_j) > 0$, it means that the two

Table 1 Dataset description

Dataset	Acquisition	Frequency	Time span	Application	Dimension
Yonsei	Site sensor	1 h	364 day	Monitored track	[day, timestep, nodes, 1]
Citi Bike	Loop detector	1 h	364 day	Shared bicycle	[day, timestep, nodes, 1]

(1) The Yonsei dataset comprises pedestrian trajectory data generated by nine graduate students over a two-month period in 2011 using a cell phone location service application called SmartDC. The data were collected in Yonsei University in Seoul, South Korea. The data size is [33203, 4], and the data features contain person_id, time, latitude, and longitude

(2) The Citi Bike dataset comprises bike-sharing trajectory data across more than 600 stations in Toronto, Canada, in 2016. We used data features including time, latitude, and longitude

samples are overlapped and vice versa; thus, the Kendall coefficient is defined as follows:

$$KD = \frac{\sum_{1\le i<j\le l} \operatorname{sgn}(x_i - x_j)\operatorname{sgn}(y_i - y_j)}{C_l^2}, \tag{26}$$

where $C_l^2 = \frac{l(l-1)}{2}$, $\operatorname{sgn}(x) = \begin{cases} 1, x > 0 \\ 0, x = 0 \\ -1, x < 0 \end{cases}$.

Model training design

There are two kinds of spatiotemporal trajectory data: the divisible region with privacy budget $\varepsilon = \sum_{i=1}^{h} \varepsilon_i$ and the indivisible region with a privacy budget $\varepsilon = min\{\varepsilon_i\}$. Influenced by geographic location, environment and time period, trajectory data are always unevenly distributed, with dense locations and sparse locations in regions. The trajectory data used in this study contained more sparse locations than dense locations; trajectory data was divided into a majority category as the divisible region and a minority category as the non-divisible region. Thus, the values of the majority category were mapped to 0 and the values of the minority category to 1. By using such a training strategy, the spatiotemporal trajectory data were predicted on a rolling basis. The loss functions for regression and classification were MSE and BCE, respectively; the optimizer was Adam, the activation function was Elu, and both early stoppages of training and learning rate adaptation were used to prevent model overfitting. In the experiments, 80% of the data was used as the training set, while the remaining 20% as the testing set. We predicted the privacy budget for the next 24 h.

Experimental results

Analysis of deep learning results

Table 2 presents the prediction results of Yonsei and Citi Bike datasets by different deep learning models, obtained by Eqs. (21), (22), and (23). The models include the convolutional neural network (CNN) [41], LSTM [30], ConvLSTM [11], ST-LSTM [34], and T-GCN [16]. The results of the different models were analyzed [42–44], and we obtained the following conclusions.

(1) Small values of the evaluation metrics were obtained in the test sets of both datasets. Thus, the model correctly learned the temporal and spatial features of the data for an excellent performance.
(2) Figure 5 shows that the method based on the spatiotemporal features (T-GCN) achieved better prediction precision than that of the other model. The T-GCN model is subsequently used to predict the privacy budget matrix for future moments which facilitates obtaining the Laplace noise at each location and publishing the noise data in advance.

Analysis of differential privacy results

To verify the privacy protection effect obtained from the test data, we used the T-GCN model to predict the privacy budget matrix, given the total privacy budget $\varepsilon = \{0.1, 0.3, 0.5, 0.7, 0.9\}$, AG division parameter $\alpha = 0.5$, and division factor $\xi = 0.5$. We query the original publishing data and noise-added publishing data to obtain the query error, JS divergence, and Kendall coefficient for test data. The query range was extracted using UG, AG, and top-down division methods. Table 3 presents the parameters for the test data and query ranges.

We evaluated the degree of protection of the dataset with different total differential privacy budgets by modifying ε using three region-division methods; the top-down method requires the prediction of data for future moments using the T-GCN model. The data for 1 day was continuously queried by time interval, and the query range was $\{4\times4, 8\times8, 16\times16, 32\times32, 64\times64\}$; the regional privacy protection evaluation metrics were averaged for each query time to obtain the average performance of each metric. Table 4 presents the results.

Figure 6 shows that the query error and JS divergence decreased as ε increased, whereas the Kendall coefficient increased as ε increased. The phenomenon occurred because as ε increases, the smaller the value of

Table 2 Prediction results of deep learning models on Yonsei and Citi Bike datasets

Model	Yonsei			Citi Bike		
	MAE	MSE	BCE	MAE	MSE	BCE
CNN	0.00508	0.00156	0.01831	0.00018	0.000034	0.00048
LSTM	0.00346	0.00158	0.01491	0.00016	0.000037	0.00016
ConvLSTM	0.00293	0.00147	0.00978	0.00014	0.000031	0.00012
ST-LSTM	0.00286	0.00138	0.00901	0.00012	0.000029	0.00011
T-GCN	0.00271	0.00122	0.00885	0.00011	0.000023	0.00009

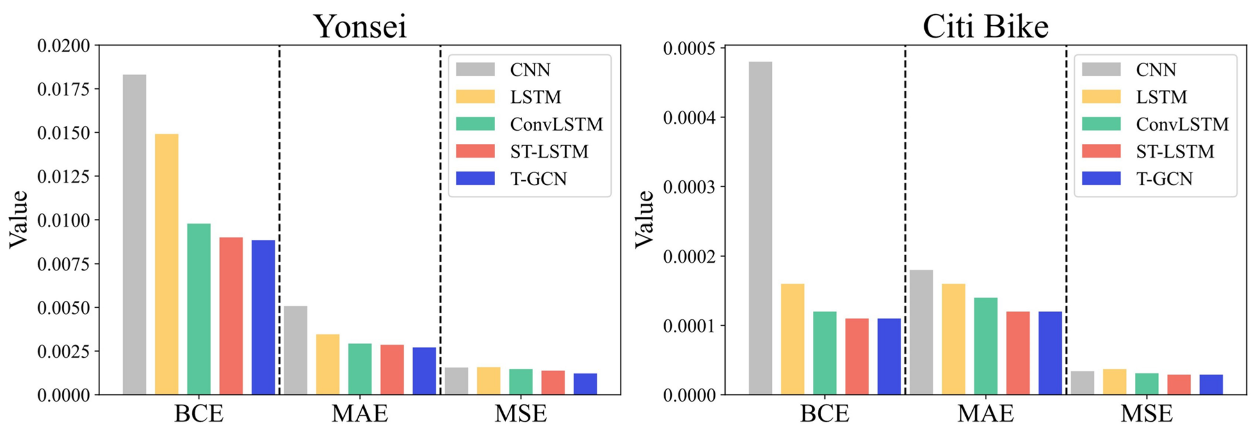

Fig. 5 BCE, MAE, and MSE of the T-GCN model

the Laplace distribution sampled to both sides is, reducing the added Laplace noise. The JS divergence of added noise shows that the top-down division outperformed the other methods. ×.

We further analyzed the degree of protection of original data at different query ranges. Given a total differential privacy budget $\varepsilon = 0.5$, the average performance of each metric was obtained by gradually increasing the query range and querying 1 day of trajectory data using different division methods. The results are presented in Table 5.

Table 3 Parameters of test data and query ranges

Dataset	Duration	Dimension	Query ranges
Yonsei	1 day	[1, 24, 64, 64, 1]	$\{4 \times 4, 8 \times 8, 16 \times 16, 32 \times 32, 64 \times 64\}$
Citi Bike	1 day	[1, 24, 64, 64, 1]	$\{4 \times 4, 8 \times 8, 16 \times 16, 32 \times 32, 64 \times 64\}$

As shown in Fig. 7, we varied the query range to verify the performance of privacy protection, and the results show the following:

(1) For the top-down division of regions, the query error, JS divergence, and Kendall coefficient are better than those of the traditional UG and AG methods, which is because the method proposed in this paper divides similar regions to allocate privacy budget, thus adding noise. The Laplace noise added in dense regions is larger than that in sparse regions. The reasonable addition of noise in each location achieves differential privacy protection and increases the usability of the dataset.

(2) As the query range $\{4 \times 4, 8 \times 8, 16 \times 16\}$ increases, it is easier to query dense locations and the greater the noise added, increases QE by the noise factor when calculating the average query error using the original data; JS divergence also increases because of the difference between the original data and

Table 4 Privacy protection average performance of the total differential privacy budget

Dataset	ε	Query Error			JS Divergence			Kendall Coefficient		
		UG	AG	Top-Down	UG	AG	Top-Down	UG	AG	Top-Down
Yonsei	0.1	0.00094	0.00055	0.00046	4.07402	1.56135	0.53506	0.00156	0.0016	0.00168
	0.3	0.00092	0.00031	0.00028	4.00918	1.38628	0.31496	0.00156	0.0016	0.00168
	0.5	0.00096	0.00044	0.00026	3.99288	1.47073	0.44056	0.00168	0.00172	0.00181
	0.7	0.00126	0.0003	0.00021	2.76302	0.88986	0.12478	0.00186	0.00191	0.002
	0.9	0.00124	0.00055	0.0003	2.41934	0.89516	0.27262	0.00206	0.00211	0.0022
Citi Bike	0.1	0.00112	0.00101	0.00096	1.66402	1.50994	1.46392	0.00424	0.00432	0.00458
	0.3	0.00121	0.001	0.00092	3.52368	2.77579	2.5524	0.00424	0.00496	0.00704
	0.5	0.00115	0.00091	0.00082	3.54326	2.87243	2.67206	0.00424	0.0055	0.00827
	0.7	0.00108	0.0009	0.00084	3.78866	2.90351	2.63912	0.00433	0.00582	0.00872
	0.9	0.0011	0.00095	0.00081	3.10342	2.65329	2.51884	0.00439	0.00616	0.00902

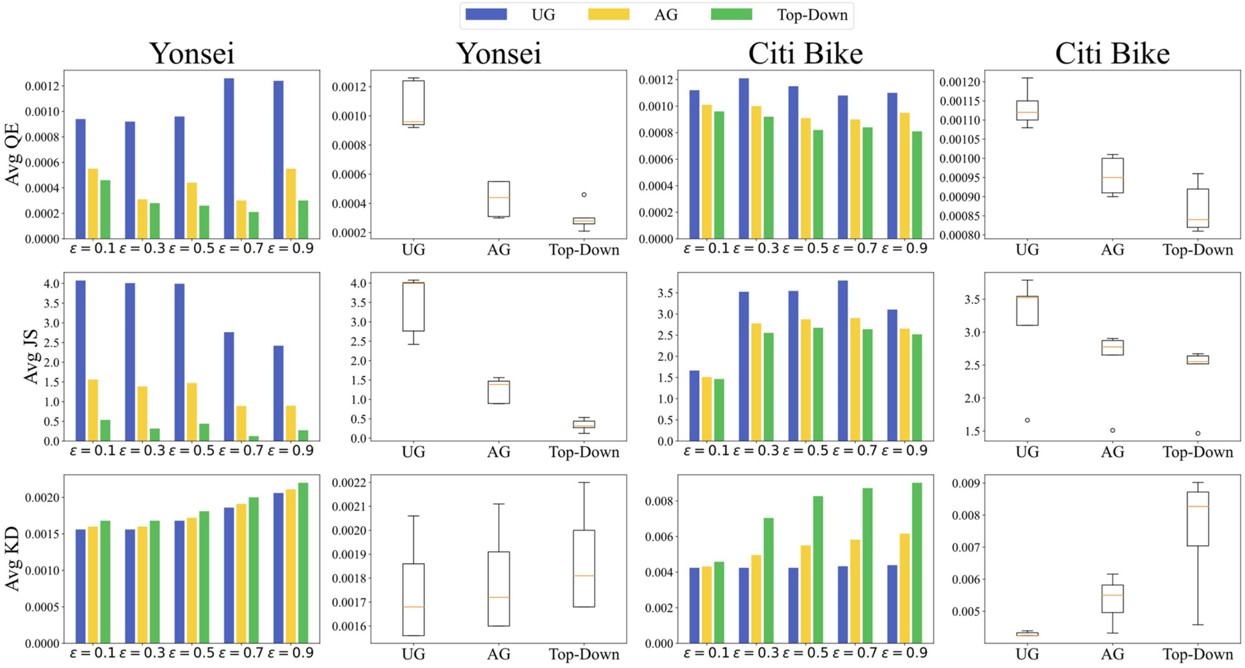

Fig. 6 Average performances of the total differential privacy budget

Table 5 Privacy protection average performance of different query ranges

Dataset	Query Range	Query Error			Jensen–Shannon Divergence			Kendall Coefficient		
		UG	AG	Top-Down	UG	AG	Top-Down	UG	AG	Top-Down
Yonsei	4*4	0.00016	0.00011	0.0001	0	0.0314	0.0787	0	0	0
	8*8	0.00074	0.00032	0.00019	4.3482	1.8328	0.3173	0	0	0
	16*16	0.00086	0.00063	0.00058	5.3425	2.6006	1.0003	0	0	0
	32*32	0.0011	0.00067	0.00034	4.9071	1.7646	0.8035	0.0014	0.0018	0.0021
	64*64	0.0008	0.00064	0.00042	4.3649	1.1301	0.003	0.0051	0.0057	0.0064
Citi Bike	4*4	0.00065	0.00059	0.0005	0.3271	0.7985	1.5057	0	0	0
	8*8	0.0008	0.00073	0.00056	4.6952	4.1017	3.2115	0	0	0
	16*16	0.0011	0.00096	0.0009	4.8393	4.5999	4.2408	0	0	0
	32*32	0.00101	0.00106	0.0011	4.6977	4.5786	4.4001	0	0	0
	64*64	0.00106	0.00109	0.0009	0.1748	0.1057	0.0022	0.0212	0.0353	0.0447

noise-added data. Here, the Kendall coefficient is nearly zero. This phenomenon occurs because the range-dense locations dominate, and the noise added to the regional locations affects the ranking with the size in the original data, which reduces the data ranking consistency.

(3) When the query range is the whole region, both query error and JS divergence are reduced, indicating that the region contains numerous traceless locations. The Laplace noise added to these locations is small or zero. As this range is several times larger than the other query ranges, interference occurs from small noisy data when calculating these average metrics, reducing their values. The Kendall coefficient gradually increases because the ranking of the original data and the data after adding Laplace noise are more consistent when the data are sorted by a small noise, increasing the indicator.

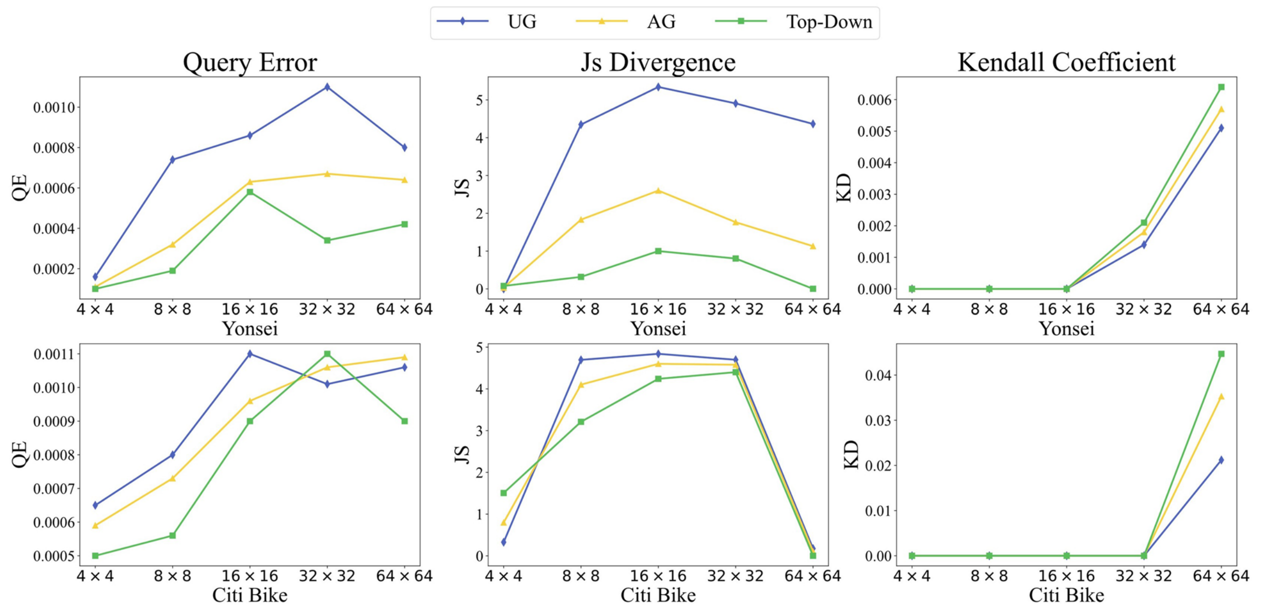

Fig. 7 Visualization average performance of different query ranges

Conclusion

In this paper, we propose a new privacy budget allocation method based on deep learning and a top-down division of regions to allocate privacy budgets for each location. We model the trajectory data by neural networks and differential privacy. First, we formulate the conditions for region division, use a recursive algorithm to divide similar regions by a top-down search, and design rules for location privacy budget allocation at different depths of recursion and division conditions. Subsequently, the spatial and temporal dependencies of grid regions are captured using the T-GCN model in comparison with other deep learning models. Finally, the T-GCN model with better prediction results is used to perform the spatiotemporal trajectory prediction task. Laplace noise is obtained according to the predicted privacy budget matrix, evaluated horizontally and vertically in two real datasets, and compared with UG and AG division methods. The top-down division is more effective and pragmatic in achieving differential privacy protection. In addition, the predicted privacy budget matrix can publish the noisy data in advance to prevent attackers from using the publishing time gap to obtain the real trajectory data. The method proposed in this paper not only predicts the privacy budgets that should be allocated at future moments, but also allocates privacy budgets more reasonably for each location while effectively achieving differential privacy protection.

Abbreviations

CNN: Convolutional neural network; GCN: Graph convolutional network; GRU: Gated recurrent unit; LSTM: Long short-term memory; ConvLSTM: Convolutional LSTM; ST-LSTM: Spatiotemporal LSTM; T-GCN: Temporal graph convolutional network; MAE: Mean absolute error; MSE: Mean squared error; BCE: Binary cross entropy; UG: Uniform grid; AG: Adaptive grid; QE: Query error; JS: Jensen–Shannon.

Authors' contributions

Jing Xiong: Propose real-time trajectory privacy protection based on improving differential privacy and deep learning model and formulate the grid division conditions of regions and design the rules of privacy budget allocation. We justify the privacy protection. The neural network code is coded with the deep learning framework Torch and TensorFlow, and after extensive experiments to verify deep learning results, a suitable set of parameters is selected as the final parameters of the model. Finally, the effectiveness of the proposed method is proved by differential privacy results analysis. Hong Zhu: Establish the experimental environment, which is Python-3.7, TensorFlow-2.8, and Torch-1.11. For the research investigation, we select the two commonly used trajectory datasets Yonsei and Citi Bike. Then, take some time to process the data into the desired format. The author(s) read and approved the final manuscript.

Funding

The research received no funds.

Availability of data and materials

(1) The Yonsei dataset generated during this study is included in this published article (Method and system for privacy preservation of inter-trajectory correlation based on PSO [Invention]), which is available in the repository (https://download.csdn.net/download/sycong/11442172).
(2) The Citi Bike datasets generated during the current study are available in the Citi Bike System Data repository (https://www.citibikenyc.com/system-data).

Declarations

Competing interests

The authors declare no conflict of interest.

Author details
[1]School of Computer of Science and Technology, Huazhong University of Science and Technology, Wuhan, China. [2]Technology Promotion Division, CEPREI, Guangzhou, China.

Received: 2 August 2022 Accepted: 13 September 2022
Published online: 29 September 2022

References

1. Zhu L, Yu FR, Wang YG et al (2019) Big data analytics in intelligent transportation systems: a survey. IEEE Trans Intell Transp Syst 20(1):383–398
2. Dwork C (2008) Differential privacy: a survey of results. In: International conference on theory and applications of models of computation. Springer, Berlin, Heidelberg, pp 1–19
3. Yan Y, Cong YM, Mahmood A (2022) A deep learning-based method for statistical publishing and privacy protection of location big data. J Commun 43(01):203–216
4. Ahmed MS, Cook AR (1979) Analysis of freeway traffic time series data by using box-jenkins techniques. Transp Res Board 722:1–9
5. Wu CH, Ho JM, Lee DT (2004) Travel-time prediction with support vector regression. IEEE Trans Intell Transp Syst 5(4):276–281
6. Liu JJ, Yu SP (2016) A hidden Markov model-based method for spatio-temporal sequence prediction. Microcomput Appl 35(01):74–76+80. https://doi.org/10.19358/j.issn.1674-7720.2016.01.023
7. Huang W, Song G, Hong H, Xie K (2014) Deep architecture for traffic flow prediction: deep belief networks with multitask learning. IEEE Trans Intell Transp Syst 15(5):2191–2201
8. Zhou FY, Jin LP, Dong J (2017) A review of convolutional neural network research. J Comput Sci 40(06):1229–1251
9. Zhang J, Zheng Y, Qi D (2016) Deep spatio-temporal residual networks for citywide crowd flows prediction
10. Diehl F, Brunner T, Le MT et al (2019) Graph neural networks for modeling traffic participant interaction. In: 2019 IEEE intelligent vehicles symposium (IV). IEEE, p 695–701. Available: http://arxiv.org/abs/1903.01254
11. Yu B, Yin H, Zhu Z (2017) Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting. arXiv preprint arXiv:1709.04875
12. Jin L, Yao C, Huang XY (2008) A nonlinear artificial intelligence ensemble prediction model for typhoon intensity. Mon Weather Rev 136(12):4541–4554
13. Wang J, Zhu R, Liu S et al (2018) Node location privacy protection based on differentially private grids in industrial wireless sensor networks. Sensors 18(2):410
14. Qardaji W, Yang WN, Li NH (2013) Differentially private grids for geospatial data. In: Proceedings of the IEEE 29th international conference on data engineering, Brisbane, Australia, pp 757–768
15. Zhang J, Xiao X, Xie X (2016) Privtree: a differentially private algorithm for hierarchical decompositions. In: Proceedings of the 2016 international conference on management of data, pp 155–170
16. Zhao L, Song Y, Zhang C et al (2019) T-gcn: a temporal graph convolutional network for traffic prediction. IEEE Trans Intell Transp Syst 21(9):3848–3858
17. Li W, Tao W, Zhou XY, Pan ZS (2020) A review of spatio-temporal sequence prediction methods. Comput Appl Res 37(10):2881–2888. https://doi.org/10.19734/j.issn.1001-3695.2019.05.0184
18. LeCun Y, Bengio Y, Hinton G (2015) Deep learning. Nature 521(7553):436–444
19. Krizhevsky A, Sutskever I, Hinton GE (2017) Imagenet classification with deep convolutional neural networks. Commun ACM 60(6):84–90
20. Tompson J, Jain A, LeCun Y, and Bregler C (2014) Joint training of a convolutional network and a graphical model for human pose estimation. In: NIPS, p 1799–1807
21. Hinton G, Deng L, Yu D et al (2012) Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups. IEEE Signal Process Mag 29(6):82–97
22. Ma J, Sheridan RP, Liaw A et al (2015) Deep neural nets as a method for quantitative structure–activity relationships. J Chem Inf Model 55(2):263–274
23. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems, p 1097–1105
24. He K, Zhang X, Ren S et al (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 770–778
25. Ronneberger O, Fischer P, Brox T (2015) U-net: convolutional networks for biomedical image segmentation. In: International conference on medical image computing and computer-assisted intervention. Springer, Cham, pp 234–241
26. Girshick R, Donahue J, Darrell T, Malik J (2014) Rich feature hierarchies for accurate object detection and semantic segmentation. CVPR, p 580–587, 2014. 1, 2
27. Begleiter R, El-Yaniv R, Yona G (2004) On prediction using variable order Markov models. J Artif Intell Res 22:385–421
28. Cho K, Van Merriënboer B, Gulcehre C, et al. Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078, 2014
29. Shi X, Chen Z, Wang H, Yeung D-Y, Wong W-k, Woo W-c (2015) Convolutional LSTM network: a machine learning approach for precipitation Nowcasting. In: NIPS, p 802–810
30. Wang Y, Long M, Wang J et al (2017) Predrnn: recurrent neural networks for predictive learning using spatiotemporal lstms. Adv Neural Inf Proces Syst 30:879–888
31. Shi X, Gao Z, Lausen L et al (2017) Deep learning for precipitation nowcasting: a benchmark and a new model. Adv Neural Inf Proces Syst 30:5617–5627
32. Wang X et al (2020) Traffic flow prediction via spatial-temporal graph neural network. In: Proceedings of the web conference 2020
33. Li Z et al (2019) A hybrid deep learning approach with GCN and LSTM for traffic flow prediction. In: 2019 IEEE intelligent transportation systems conference (ITSC), p 1929–1933
34. Bruna J, Zaremba W, Szlam A, Lecun Y (2013) Spectral networks and locally connected networks on graphs. CoRR, abs/1312.6203
35. Cho K, Merrienboer BV, Bahdanau D, Bengio Y (2014) On the properties of neural machine translation: encoder-decoder approaches. In: Proceedings of the Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation, p 103–111
36. Dwork C, Roth A (2014) The algorithmic foundations of differential privacy. Found Trends Theor Comput Sci 9(3–4):211–407
37. Cormode G, Procopiuc C, Srivastava D, Shen E, Yu T (2012) Differentially private spatial decompositions. In: Proceedings of the IEEE 28th International Conference on Data Engineering, Washington, DC, USA, pp 20–31
38. Yu Y, Si XS, Hu CH et al (2019) A review of recurrent neural networks: LSTM cells and network architectures. Neural Comput 31(7):1235–1270
39. Qiao S, Zeng Y, Zhou L, Liu Z, Ma J (2017) A secure authentication method of intelligent terminals based on jensen-shannon divergence," in International Conference on Networking and Network Applications (NaNA), p 158–163
40. Zhang Q (2020) Correlation test of function-based data based on Kendall's correlation coefficient. Northeast Normal University. https://doi.org/10.27011/d.cnki.gdbsu.2020.000389
41. Shi Y, Tian Y, Wang Y et al (2017) Sequential deep trajectory descriptor for action recognition with three-stream CNN. IEEE Trans Multimed 19(7):1510–1520
42. Iwendi C, Mohan S, Khan S, Ibeke E, Ahmadian A, Ciano T (2022) Covid-19 fake news sentiment analysis. Comput Electr Eng 101:107967. https://doi.org/10.1016/j.compeleceng.2022.107967 Epub 2022 Apr 22. PMID: 35474674; PMCID: PMC9023343
43. Kumar RL, Khan F, Din S, Band SS, Mosavi A, Ibeke E (2021) Recurrent neural network and reinforcement learning model for COVID-19 prediction. Front Public Health 9:744100. https://doi.org/10.3389/fpubh.2021.744100 PMID: 34671588; PMCID: PMC8521000
44. Iwendi C, Srivastava G, Khan S et al (2020) Cyberbullying detection solutions based on deep learning architectures. Multimed Syst. https://doi.org/10.1007/s00530-020-00701-5

Publisher's Note