Metrika
https://doi.org/10.1007/s00184-024-00985-x

Effective and robust clustering for spatiotemporally dependent data

Feiyun Wang[1] · Wangyong Chen[1] · Yao Hu[1] · Yewen Chen[2]

Received: 26 January 2024 / Accepted: 1 December 2024

Abstract

Spatio-temporal data exists in various fields such as transportation and environment, and the analysis of clustering plays a critical role in understanding such data. However, spatio-temporal clustering faces challenges, as it is frequently affected by the time length of data and the intricate dependence structure of spatio-temporal data. In this work, we propose a robust approach to achieve the fuzzy clustering of complex spatio-temporal data. This involves two key technologies: (1) the reconstruction of spatio-temporal data using B-splines, and (2) the incorporation of a weighted exponential function to characterize spatial and temporal dependencies. The data reconstruction reduces the impact of the time length on clustering while improving computational efficiency. Meanwhile, the integration of a spatio-temporal scaling factor within the weighted function addresses the scale difference between spatial and temporal coordinates. For the implementation, the clustering process is performed using the Partitioning Around Medoids (PAM) algorithm, and the optimization of the number of clusters is achieved through the use of the fuzzy silhouette coefficient. Extensive simulation studies and two real-world applications are used to demonstrate the effectiveness of the proposed method.

Keywords Spatio-temporal data · Spatio-temporal fuzzy clustering · Spatio-temporal distance · Weighted exponential function · PAM algorithm

✉ Yao Hu
yhu1@gzu.edu.cn

✉ Yewen Chen
Yewen.Chen@uga.edu

Feiyun Wang
wangfeiyun98@163.com

Wangyong Chen
gs.wychen22@gzu.edu.cn

[1] School of Mathematics and Statistics, Guizhou University, Guiyang, China

[2] College of Public Health, University of Georgia, Athens, USA

Published online: 27 December 2024

1 Introduction

With the advancements in Geographic Information Systems (GIS), Remote Sensing (RS), and Global Positioning System (GPS) technologies, a vast amount of spatio-temporal data is being generated and collected. Consequently, spatio-temporal data analysis has received more and more attention in recent decades. In this context, spatio-temporal clustering becomes more and more important, owing to its capability to identify comparable subgroups within datasets. As a result, spatio-temporal clustering finds applications in diverse fields such as geography, meteorology, social sciences, and economics (Liu and George 2003; Lurka 2021; Yürüşen et al. 2021; Cammalleri et al. 2023).

In literature, the main methods for spatio-temporal clustering can be classified into three types: spatio-temporal scan methods (Kulldorff et al. 2005; Gaudart et al. 2006), density-based methods (Wang et al. 2006; Pei et al. 2010), and distance-based methods (Zaliapin et al. 2008). Spatio-temporal scanning methods usually consider global autocorrelation and density-based methods determine the clustering structure based on the closeness of the samples. In contrast, distance-based methods can determine the clustering structure based on the spatio-temporal dependent variations among random samples, and the spatio-temporal dependence can be assessed by using prior knowledge. Recent developments in spatio-temporal clustering are reviewed in Ansari et al. (2020). Among all these approaches, distance-based methods have garnered increasing attention and found wide applications in the era of big data, primarily because of their remarkable flexibility and high computational efficiency (Tork 2012; Deb and Karmakar 2023). This article focuses on distance-based fuzzy methods.

In literature, many researchers prefer to use the framework of fuzzy clustering (McBratney and Moore 1985; Keogh et al. 2001; Maharaj and D'Urso 2011). Compared to traditional clustering methods, fuzzy clustering is more attractive because it does not require any distribution assumptions and offers greater flexibility in applications. For spatio-temporal data, spatial information plays an important role. Generally, objects that are closer in space tend to be more similar than those that are farther apart. Therefore, when conducting spatio-temporal clustering, some researchers incorporate spatial information as penalty terms in the objective function (Gao and Yu 2016; D'Urso et al. 2022). Some scholars (Mattera 2022) characterize spatio-temporal dependence by assigning weights to time and space distances, employing the Euclidean distance metric. This approach has been successfully employed in analyzing the spread of COVID-19 in Italy. Recently, a similar spatio-temporal clustering method, utilizing a weighting scheme for both temporal and spatial distances, was introduced and employed to analyze spatio-temporal COVID-19 data in the United States (Deb and Karmakar 2023).

In most of the aforementioned studies, researchers have commonly employed Euclidean distance to measure the similarities between samples. However, it is worth noting that Euclidean distance is susceptible to the presence of outliers or heavy-tailed data. Additionally, the time length of spatio-temporal data, closely related to the dimensionality of data, can result in the computational burden for a large number of observations.

To tackle these challenges, this work develops a robust clustering procedure of spatio-temporal dependent data under a fuzzy clustering framework. Firstly, a B-splines-based technique is employed to reconstruct the spatio-temporal data as the sum of the product of a series of fixed temporal bases and unknown coefficients to be estimated. This data reconstruction technique reduces the dimension of spatio-temporal data while preserving essential information from the original dataset, thus simplifying calculations. Secondly, a weighted exponential function is utilized to model data dependence across both time and space. Specifically, two different exponential functions are employed to represent the correlation between any two coefficients derived from B-splines and the correlation between any two spatial points, respectively. For the clustering process, the Partitioning Around Medoids (PAM) algorithm is utilized. PAM offers cluster centers that correspond to real spatial units, thus improving interpretability. Furthermore, the determination of the optimal number of clusters is accomplished using the fuzzy silhouette coefficient. Through numerical simulations and two real-world case studies involving urban traffic data in Guiyang, China, and urban air quality data in China, the proposed method's extensive applicability is validated.

The framework of this article is as follows: Sect. 2 provides details for the proposed method and presents an algorithm for implementation. Section 3 investigates the performance of the proposed method using a simulation study. Section 4 includes an analysis of two real data sets to illustrate the broad applicability of the proposed method. We conclude with a discussion in Sect. 5.

2 Robust fuzzy spatio-temporal clustering model

To carry out spatio-temporal clustering, this study adopts a fuzzy framework to construct a clustering model. This choice is motivated by the limitations of traditional clustering methods in defining precise cluster boundaries, whereas fuzzy clustering allows for membership-based assignments. Fuzzy clustering offers greater maneuverability compared to traditional methods and is more flexible as it does not necessitate assumptions about the data source.

The clustering process employs the PAM algorithm (Mondal and Choudhury 2013), which computes clustering centers to represent specific spatial units. This approach enhances interpretability compared to K-means, which merely calculates the centroid of the data. By using PAM algorithm, the resulting cluster centers correspond to actual spatial locations, providing more meaningful insights.

When clustering time series data, the problem of high dimensionality is usually faced, so the data needs to be reconstructed to reduce the dimensionality and retain as much information as possible from the original time series. Many methods for reconstructing time series data have been proposed in the literature, including discrete Fourier transform (Faloutsos et al. 1994), discrete wavelet transform (Ann et al. 2010), piecewise aggregation approximation (Keogh et al. 2001), B-splines (Abraham et al. 2003), and other methods. Since B-splines have good properties, in this paper B-splines are used for data reconstruction of time series data.

2.1 Time series reconstruction: B-splines

For any given time series $\{y_i(t)\}$ $(i = 1, 2, \ldots, n)$, which represents the observations of the i-th spatial unit at T time points, we consider a model with a p-dimensional functional basis $\{B_{s,d}(\cdot)\}_{s=0}^{p}$ as follows:

$$y_i(t) = \sum_{s=0}^{p} \beta_i^s B_{s,d}(t) + \varepsilon_i, t = 1, 2, \ldots, T \tag{1}$$

The coefficient vector $\hat{\boldsymbol{\beta}}_i = (\beta_i^1, \beta_i^2, \ldots, \beta_i^p)^\top$ is estimated by simple least squares, i.e., $\hat{\boldsymbol{\beta}}_i$ is solved so that $\sum_{j=1}^{T}\left(y(t_j) - \sum_{s=0}^{p} \beta_i^s B_{s,d}(t_j)\right)^2$ is minimized, which leads to $\hat{\boldsymbol{\beta}}_i = \left(\boldsymbol{B}_i^\top \boldsymbol{B}_i\right)^{-1} \boldsymbol{B}_i^\top \boldsymbol{y}_i$. $\boldsymbol{B}_i$ is a matrix, which leads to the n coefficient vectors $\left(\hat{\boldsymbol{\beta}}_1, \hat{\boldsymbol{\beta}}_2, \ldots, \hat{\boldsymbol{\beta}}_n\right)$ with dimension p.

In this paper, cubic B-splines are employed due to their favorable theoretical properties and ability to achieve dimension reduction by selecting an appropriate number of internal nodes (de Boor 2001). The dimensionality can be reduced to $p = S+4$, where S represents the number of internal nodes. For the selection of internal nodes, there are two approaches. One approach is to choose $S = T^{\frac{1}{5}}$ (where T represents the length of the time series), as suggested in Theorem 1 of Huang and Shen (2004) (for more details, please refer to their work). Another approach is to utilize cross-validation, where the data is divided into training and validation sets. Multiple models are trained and predicted using different numbers of internal nodes, and the mean squared error of the predicted results is calculated. The number of internal nodes that minimize the mean squared error is selected as the final choice. When the data exhibits periodic patterns, Fourier bases or wavelet bases can also be considered.

2.2 Spatio-temporal distance

Defining the similarity between two spatial units is a crucial step in cluster analysis. In this paper, we employ a weighted combination of spatial and temporal distances to characterize the distances between two spatial units. For measuring spatial distance, we use the latitude and longitude data of the spatial units. For temporal distance, we utilize the coefficient vectors obtained after B-splines reconstruction.

When measuring spatial and temporal distances, the choice of distance function is of great importance as it can impact the final clustering results. The Euclidean distance is widely used; however, it is sensitive to outliers. To address this drawback, an exponential distance measure is proposed, which can mitigate the impact of outliers and exhibit a certain level of robustness (Zhang and Chen 2004). For any two coefficient vectors $\hat{\boldsymbol{\beta}}_i$ and $\hat{\boldsymbol{\beta}}_j$, the exponential distance between them is defined as follows:

$$d_{\exp}\left(\hat{\boldsymbol{\beta}}_i, \hat{\boldsymbol{\beta}}_j\right) = \left[1 - \exp\left(-\theta \left\|\hat{\boldsymbol{\beta}}_i - \hat{\boldsymbol{\beta}}_j\right\|^2\right)\right]^{1/2}, \tag{2}$$

the parameter θ plays a significant role in the exponential distance measure. As it increases, the exponential distance approaches its maximum value of 1. The following methods were used to determine θ value (Wu and Yang 2002):

$$\theta = \left[\frac{\sum_{i=1}^{n} d^2\left(\hat{\boldsymbol{\beta}}_i, \hat{\boldsymbol{\beta}}_q\right)}{n} \right]^{-1}, \tag{3}$$

where $\hat{\boldsymbol{\beta}}_q$ is the q-th coefficient vector, and $q = \arg\min_{1\leq i\leq n} \sum_{j=1}^{n} d^2(\hat{\boldsymbol{\beta}}_i, \hat{\boldsymbol{\beta}}_j)$, $d^2(\hat{\boldsymbol{\beta}}_i, \hat{\boldsymbol{\beta}}_j)$ represents the Euclidean distance square i. e. That is $d^2(\hat{\boldsymbol{\beta}}_i, \hat{\boldsymbol{\beta}}_j) = \left\| \hat{\boldsymbol{\beta}}_i - \hat{\boldsymbol{\beta}}_j \right\|^2$.

In order to better capture the heterogeneity within the data sample set and enhance the clustering performance, this paper introduces the concept of weighted exponential distance for describing the temporal and spatial distances. The weight assigned to the distance measure is determined based on the coefficient of variation.

The coefficient of variation weighting method (Ren and Jian 2021) is proposed on the basis of inverse variance. The coefficient of variation for a given dataset is calculated by dividing its standard deviation by the absolute value of the mean. That is, for n data points in the data set $x_1, x_2, \ldots, x_n$, denote $\overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$, $S_x = \sqrt{\left(\frac{1}{n-1}\sum_{i=1}^{n}\left(x_i - x^-\right)^2\right)}$,then $v_x = \frac{S_x}{|\overline{x}|}$ is the coefficient of variation of $x_1, x_2, \ldots, x_n$.

In this paper, for the p-dimensional coefficient vector $\hat{\boldsymbol{\beta}}_i$, there are p coefficients of variation $v_1, v_2, \ldots, v_p$, and the weight corresponding to the i-th coefficient of variation is as follows:

$$\omega_i = \frac{v_i}{\sum_{i=1}^{p} v_i}, i = 1, 2, \ldots, p. \tag{4}$$

The larger the value of v_i, the greater the change in different objects and the stronger the ability to distinguish objects, so it should be paid attention to. Thus, the weighted exponential distance formula is obtained as:

$$d_{WExp}(\hat{\boldsymbol{\beta}}_i, \hat{\boldsymbol{\beta}}_j) = \left[1 - \exp\left(-\theta\left(\omega_1(\beta_{i1} - \beta_{j1})^2 + \ldots + \omega_p(\beta_{ip} - \beta_{jp})^2\right)\right)\right]^{1/2}. \tag{5}$$

Denoting the latitude and longitude coordinates of the i-th and j-th spatial units by $\mathbf{p}_i = (\mathrm{X}_{iLon}, \mathrm{Y}_{iLat})$ and $\mathbf{p}_j = \left(\mathrm{X}_{jLon}, \mathrm{Y}_{jLat}\right)$, the spatial distance between the two spatial unit is obtained defined as:

$$\begin{aligned} d_{WExp}(\mathbf{p}_i, \mathbf{p}_j) &= \left[1 - \exp\left(-\theta_s \left\| w_{ws} \odot \left(\mathbf{p}_i - \mathbf{p}_j\right)\right\|^2\right)\right]^{1/2} \\ &= \left[1 - \exp\left(-\theta_s\left(\omega_1\left(\mathrm{X}_{iLon} - \mathrm{X}_{jLon}\right)^2 + \omega_2\left(\mathrm{Y}_{iLat} - \mathrm{Y}_{jLat}\right)^2\right)\right)\right]^{1/2}, \end{aligned} \tag{6}$$

where ω_1 and ω_2 denote the weights of longitude and latitude, respectively. $\odot$ represents the Hadamard product. Let $\hat{\boldsymbol{\beta}}_i$ and $\hat{\boldsymbol{\beta}}_j$ denote the p-dimensional coefficient vectors of the i-th and j-th spatial units, respectively, obtained by B-splines transformation. The time distance between the two spatial units can be calculated as follows:

$$\begin{aligned} d_{WExp}(\hat{\boldsymbol{\beta}}_i, \hat{\boldsymbol{\beta}}_j) &= \left[1 - \exp\left(-\theta_t \left\| w_{wt} \odot \left(\hat{\boldsymbol{\beta}}_i - \hat{\boldsymbol{\beta}}_j\right)\right\|^2\right)\right]^{1/2} \\ &= \left[1 - \exp\left(-\theta_t \left(\omega_1 \left(\beta_{i1} - \beta_{j1}\right)^2 + \ldots + \omega_p \left(\beta_{ip} - \beta_{jp}\right)^2\right)\right)\right]^{1/2}. \end{aligned} \tag{7}$$

To balance the scale difference between spatial and temporal coordinates, this paper introduces a spatio-temporal scaling factor ”s” (Lurka 2021) into the construction of spatio-temporal distance. This approach enhances the representation of data information and yields the squared spatio-temporal distance between two spatial units, defined as follows:

$$\begin{aligned} d^2(s,t) &= w_s^2 \left[1 - \exp\left(-\theta_s \left\| w_{ws} \odot (\mathbf{p}_i - \mathbf{p}_j)\right\|^2\right)\right] \\ &+ w_t^2 \left[s^2 \left(1 - \exp\left(-\theta_t \left\| w_{wt} \odot \left(\hat{\boldsymbol{\beta}}_i - \hat{\boldsymbol{\beta}}_j\right)\right\|^2\right)\right)\right], \end{aligned} \tag{8}$$

where $w_s + w_t = 1$, w_s and w_t denote a weight in space and time, respectively, and $s^2 = \frac{Var(\mathrm{X}) + Var(\mathrm{Y})}{Var(T_\beta)}$, $Var(\mathrm{X})$, $Var(\mathrm{Y})$ and $Var\left(T_\beta\right)$ are the longitude and latitude of all spatial units of the input and the variance of all coefficient vectors, respectively.

2.3 Spatio-temporal clustering model based on B-splines with weighted exponential distance

2.3.1 Spatio-temporal clustering model

Let $D_{space} = \left(\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_n\right)$ be the latitude and longitude dataset for n spatial units. Additionally, let $D_{Time} = \left(\hat{\boldsymbol{\beta}}_1, \hat{\boldsymbol{\beta}}_2, \ldots, \hat{\boldsymbol{\beta}}_n\right)$ be the dataset obtained by transforming the time series data into coefficient vectors after B-splines reconstruction. Assuming that there are C clusters, the membership degree of the i-th spatial unit to the c-th cluster is denoted as u_{ic}. To cluster n spatial units, a spatio-temporal fuzzy clustering model based on B-splines with weighted exponential distance (STFCMd-BSWE) is proposed.

$$\begin{cases} \min \sum\limits_{\mathrm{i}=1}^{n} \sum\limits_{c=1}^{C} u_{ic}^m \left[w_s^2 \left[1 - \exp\left(-\theta_s \left\| w_{ws} \odot \left(\mathbf{p}_i - \tilde{\mathbf{p}_c}\right)\right\|^2\right)\right] \right. \\ \qquad \left. + w_t^2 \left[s^2 \left(1 - \exp\left(-\theta_t \left\| w_{wt} \odot \left(\hat{\boldsymbol{\beta}}_i - \tilde{\hat{\boldsymbol{\beta}}}_c\right)\right\|^2\right)\right)\right]\right] \\ s.t. \sum\limits_{c=1}^{C} u_{ic} = 1, u_{ic} \geq 0, u_{ic} \in [0,1] \\ s.t.\ w_s + w_t = 1, w_s, w_t \geq 0 \\ \quad s^2 = \frac{Var(\mathrm{X}) + Var(\mathrm{Y})}{Var(T_\beta)} \end{cases} \tag{9}$$

where $m > 1$ is a constant representing the fuzzy parameter. $\mathbf{p}_i$ and $\widetilde{\mathbf{p}_c}$ denote the two-dimensional vectors of latitude and longitude corresponding to the i-th spatial unit and the c-th cluster-centered spatial unit, respectively. $\hat{\boldsymbol{\beta}}_i$ and $\widetilde{\hat{\boldsymbol{\beta}}_c}$ denote the coefficient vectors of the time series corresponding to the i-th spatial unit and the c-th cluster-centered spatial unit, respectively, which are obtained by B-splines reconstruction. The optimization problem can be solved using the Lagrange multiplier method, the solution procedure is referred to by D'urso and Massari (2019). The optimal parameters can be obtained as follows:

$$u_{ic} = \frac{\left[w_s^2\left[1 - \exp\left(-\theta_s \left\| w_{ws} \odot \left(\mathbf{p}_i - \widetilde{\mathbf{p}_c}\right)\right\|^2\right)\right] + w_t^2\left[s^2\left(1 - \exp\left(-\theta_t \left\| w_{wt} \odot \left(\hat{\boldsymbol{\beta}}_i - \widetilde{\hat{\boldsymbol{\beta}}_c}\right)\right\|^2\right)\right)\right]\right]^{-\frac{1}{m-1}}}{\sum_{c'=1}^{C}\left[w_s^2\left[1 - \exp\left(-\theta_s \left\| w_{ws} \odot \left(\mathbf{p}_i - \widetilde{\mathbf{p}_{c'}}\right)\right\|^2\right)\right] + w_t^2\left[s^2\left(1 - \exp\left(-\theta_t \left\| w_{wt} \odot \left(\hat{\boldsymbol{\beta}}_i - \widetilde{\hat{\boldsymbol{\beta}}_{c'}}\right)\right\|^2\right)\right)\right]\right]^{-\frac{1}{m-1}}} \tag{10}$$

$$w_s = \frac{\sum_{\mathrm{i}=1}^{n}\sum_{c=1}^{C} u_{ic}^m \left[s^2\left(1 - \exp\left(-\theta_t \left\| w_{wt} \odot \left(\hat{\boldsymbol{\beta}}_i - \widetilde{\hat{\boldsymbol{\beta}}_c}\right)\right\|^2\right)\right)\right]}{\sum_{\mathrm{i}=1}^{n}\sum_{c=1}^{C} u_{ic}^m \left[\left[1 - \exp\left(-\theta_s \left\| w_{ws} \odot \left(\mathbf{p}_i - \widetilde{\mathbf{p}_c}\right)\right\|^2\right)\right] + \left[s^2\left(1 - \exp\left(-\theta_t \left\| w_{wt} \odot \left(\hat{\boldsymbol{\beta}}_i - \widetilde{\hat{\boldsymbol{\beta}}_c}\right)\right\|^2\right)\right)\right]\right]} \tag{11}$$

$$w_t = \frac{\sum_{\mathrm{i}=1}^{n}\sum_{c=1}^{C} u_{ic}^m \left[1 - \exp\left(-\theta_s \left\| w_{ws} \odot \left(\mathbf{p}_i - \widetilde{\mathbf{p}_c}\right)\right\|^2\right)\right]}{\sum_{\mathrm{i}=1}^{n}\sum_{c=1}^{C} u_{ic}^m \left[\left[1 - \exp\left(-\theta_s \left\| w_{ws} \odot \left(\mathbf{p}_i - \widetilde{\mathbf{p}_c}\right)\right\|^2\right)\right] + \left[s^2\left(1 - \exp\left(-\theta_t \left\| w_{wt} \odot \left(\hat{\boldsymbol{\beta}}_i - \widetilde{\hat{\boldsymbol{\beta}}_c}\right)\right\|^2\right)\right)\right]\right]} \tag{12}$$

Usually, m does not take on excessively large values. This is because as $m \to \infty$, the membership degree $u_{ic} \to \frac{1}{C}$. When $w_s = 0$, the spatio-temporal clustering model can be considered as a pure time series clustering. Similarly, when $w_t = 0$, the spatio-temporal clustering model can be seen as a pure spatial clustering in latitude and longitude. The computational steps of the proposed spatio-temporal clustering model are outlined in Algorithm 1.

Algorithm 1 STFCMd-BSWE algorithm

```
Fixed $C$, and maximum number of iterations max.iter;
Set the initial iteration number iter=0 and initial objective function value;
Generate $C$ initial iteration centroids: $\hat{\beta}_1, \hat{\beta}_2, \ldots, \hat{\beta}_C$:
    Calculate the spatio-temporal distance $d_{ij}$ between every pair of all objects;
    Calculate $v_j = \sum_{i=1}^{n} \frac{d_{ij}}{\sum_{l=1}^{n} d_{il}}, j = 1, 2, \ldots, n$;
    Sort $v_j$ in ascending order. Select $C$ objects having the first $C$ smallest values as initial medoids.
Randomly generate $w_s$ from a uniform distribution U(0,1), $w_t = 1 - w_s$;
Repeat the following steps:
    Store the current iteration centroids $(\hat{\beta}_1, \hat{\beta}_2, \ldots, \hat{\beta}_C)_{OLD} = (\hat{\beta}_1, \hat{\beta}_2, \ldots, \hat{\beta}_C)$;
    Assign each remaining object to the cluster represented by the nearest centroid;
    Calculate the membership degree matrix $U$, weight allocated to space $w_s$ and weight allocated to
        time $w_t$ according to equation (10), equation (11) and equation (12);
    Calculate the objective function value according to equation (9);
    The iterative process uses the PAM algorithm;
    iter ←iter + 1;
Until, $(\hat{\beta}_1, \hat{\beta}_2, \ldots, \hat{\beta}_C)_{OLD} = (\hat{\beta}_1, \hat{\beta}_2, \ldots, \hat{\beta}_C)_{OLD}$ or iter =max.iter.
```

2.3.2 Convergence analysis of algorithm

Let us now briefly analyze the convergence of Algorithm 1.

The cost function is bounded: since the sample units to be clustered are finite, there exists a maximum possible total cost (when all objects choose the farthest center) and a minimum possible total cost (i.e., a division of the global optimal solution). This means that the cost function is a bounded function.

The cost function decreases or remains constant with each iteration: centroids are replaced only if they lower the overall cost, ensuring that the total cost never increases during an iteration

Finite state space: Considering that the number of sample units is finite, the number of center objects (state space of the algorithm) that can be selected is also finite. Each iteration of the algorithm can be viewed as a move in the state space.

Each iteration maintains or reduces at least the total cost: if no cost-reducing center substitution can be found in a given iteration, the total cost remains unchanged, the algorithm stops, and the algorithm converges.

Based on the above, it is known that the Algorithm 1 will converge to a locally optimal partition after a finite number of iterations.

2.4 Selection of optimal clusters

In order to select the best partition C, we use the Fuzzy Silhouette (FS) coefficient as an evaluation criterion (Campello and Hruschka 2006), and select the number of clusters corresponding to the maximum value of FS as the optimal number of clusters.

The FS coefficient is a validity measure that considers both inter-cluster and intra-cluster distances. It is computed as a weighted average of individual Silhouette widths, denoted as λ_i. The formula for the FS index is as follows:

$$FS = \frac{\sum_{i=1}^{n}\left(u_{pi}-u_{qi}\right)^{\alpha}\cdot\lambda_i}{\sum_{i=1}^{n}\left(u_{pi}-u_{qi}\right)^{\alpha}} \quad \text{with} \quad \lambda_i = \frac{l_{pi}-a_{pi}}{\max\{l_{pi},a_{pi}\}}, \tag{13}$$

where a_{pi} is the average distance between the i-th spatial unit and other units in the same cluster p $(p = 1, 2, \ldots, C)$, and l_{pi} is the minimum average distance from the i-th spatial unit to all units belonging to cluster q $(q \neq p)$. $\left(u_{pi}-u_{qi}\right)^{\alpha}$ is the weight of each λ_i, while u_{pi} and u_{ai} are the first and second largest elements corresponding to the i-th column of the membership degree matrix. Here, $\alpha \geq 0$ is the optional weighting coefficient.

2.5 Evaluation of clustering results

To evaluate the effectiveness of the proposed method, certain clustering evaluation indices are required. In this study, two types of indices are considered: purity (Moayedi et al. 2019) and Fowlkes-Mallows index (FMI) (Campello 2007). Purity and FMI both provide measures of how closely the clustering results align with the true results, with higher values indicating better clustering performance. Both purity and FMI range between 0 and 1.

Purity is an external clustering validation method that compares the clustering results with externally provided true clustering labels. In other words, it measures the extent to which the clustering labels reflect the true underlying structure of the data.

FMI, on the other hand, is a measure of clustering algorithm performance that requires knowledge of the true cluster labels. It assesses similarity based on the intersection and union of the true labels and the predicted cluster assignments, as well as the ratio of point pairs within and between clusters.

Let $R = \{R_1, \ldots, R_r\}$ denote the clusters in which the data are truly divided, and $E = \{E_1, \ldots, E_e\}$ denote the clusters in which the data are estimated. We can then express the following relationship:

$$\text{Purity}\,(E) = \frac{1}{n}\sum_{i=1}^{e}\max_{j}\left|E_i \cap R_j\right|, \tag{14}$$

$$\text{FMI}\,(E) = \frac{TP}{\sqrt{(TP+FP)\,(TP+FN)}}, \tag{15}$$

where TP denotes the number of pairs of sample points belonging to the same cluster in R and also belonging to the same cluster in E; FP denotes the number of pairs of sample points belonging to the same cluster in R and different clusters in E; and table FN shows the number of pairs of sample points belonging to different clusters in R and the same cluster in E.

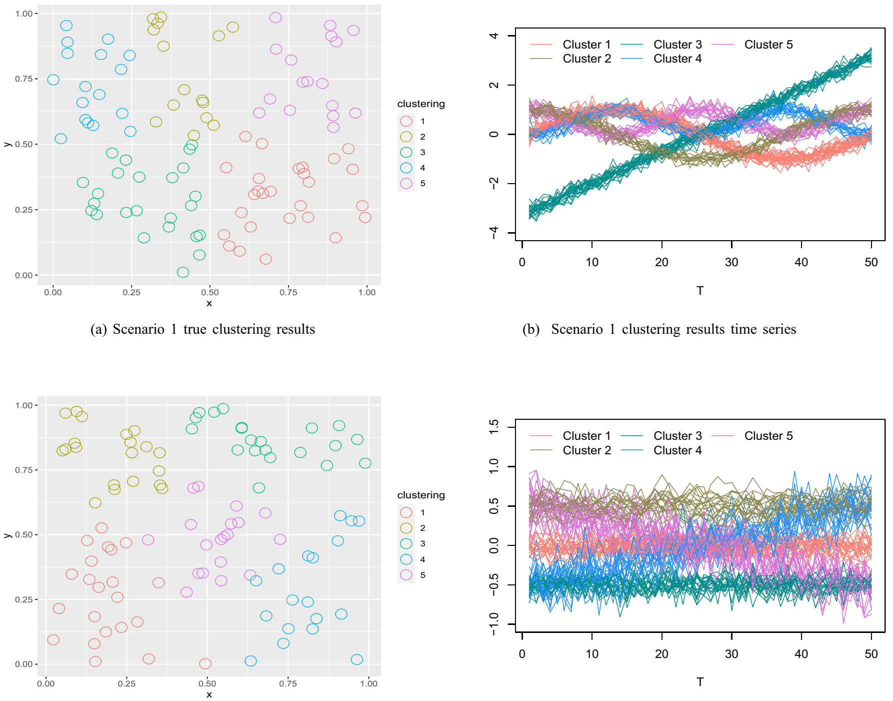
3 Simulation analysis

For the generation of spatial units, a two-dimensional xy-plane is utilized to replace the latitude and longitude data of the actual real spatial units. The n units are generated uniformly and randomly from a rectangular region of the xy-plane. The n units are divided into C clusters based on the (x, y) coordinates using the K-means algorithm. In order to assign values to the divided clusters, we simulate two scenarios.

In scenario 1 (Sce1), the units within the same cluster are set to have similar time series of length T, with distinct differences between clusters. On the other hand, in scenario 2 (Sce2), the distinctions between clusters are not obvious.

For the number of spatial units $n \in (50, 100)$, the number of clusters $C \in (3, 5)$, and the time length $T \in (50, 100, 500)$. For scenario 1, when $C = 3$, the time series data of these clusters are generated from the three functions $f(x) = \sin(x)$, $g(x) = \cos(x)$, and $h(x) = -\pi + x$, respectively, and each time series data error obeys $N(0, 0.2^2)$ when the interval $[0, 2\pi]$ is taken at equal intervals of size T. When $C = 5$, the time series data of these clusters are generated from the five functions $f(x) = \sin(x)$, $g(x) = \cos(x)$, $h(x) = -\pi + x$, $m(x) = \sin^2(x)$, and $n(x) = \cos^2(x)$, respectively, and other settings are the same as in $C = 3$ case. For scenario 2, the time series data of the same cluster comes from the same time series process of length T. When $C = 3$, three different stationary time series processes γ_1, $\gamma_2 gamma_3$ are set to generate the time series data. γ_1 is set to AR(1): $x_t = 0.5 + 0.9x_{t-1} + \varepsilon_t$, γ_2 is set to MA(1): $x_t = -0.5 + \varepsilon_t - 0.3\varepsilon_{t-1}$, γ_3 is set to ARMA(1,1): $x_t - 0.3x_{t-1} = \varepsilon_t - 0.2\varepsilon_{t-1}$ and ε_t obey $N(0, 0.1^2)$. When $C = 5$, the method for generating time series data with three clusters is the same as described earlier, and two more clusters are set to generate the time series data as the $f(x) = \sin(x)$ and $g(x) = \cos(x)$ functions, respectively. Where for $f(x) = \sin(x)$, the capacity of size T is taken at equal intervals in interval $[-\frac{\pi}{6}, \frac{\pi}{6}]$. For $g(x) = \cos(x)$, the capacity size of T is taken at equal intervals in the interval $[\frac{\pi}{3}, \frac{2\pi}{3}]$. Each time series error obeys $N(0, 0.2^2)$. The value of the fuzzy parameter m is also uncertain, in the previous literature, some scholars take $m = 1.5$ (D'Urso et al. 2022), and some scholars take $m = 2$ (Mattera 2022). In this paper's simulation analysis, we examine two cases: $m = 1.5$ and $m = 2$, to investigate the impact of the fuzzy parameter on the clustering results. When reconstructing the data using B-splines for time series, we explore a range of 3-10 nodes. The data is then divided into training and test sets, and the number of nodes corresponding to the minimum mean square error obtained in the test set is selected as the optimal choice. To determine the optimal number of clusters, we employ the fuzzy silhouette coefficient discussed in Section 2.4. Algorithm 1 is utilized for spatio-temporal clustering, and the evaluation indexes after clustering are calculated to assess the effectiveness of the model.

To assess the effectiveness of the proposed method in this paper, we will compare its accuracy in identifying real clusters with other methods. The first method being considered is the weighted fuzzy clustering model proposed by Raffaele Mattera (Mattera 2022). This method defines spatial distance using the Euclidean distance of local spatial autocorrelation coefficients and temporal distance using the Euclidean distance of lagged L-order autocorrelation coefficients for the time series. In this comparison, we set $L = 10$ and $L = 50$, but when the number of spatial units $n = 50$, only $L = 10$ is considered. We refer to this method as MWFC, and the fuzzy parameter for this method is set to $m = 2$. The second method is the spatio-temporal clustering algorithm ST-DBSCAN (Birant and Kut 2007) which is widely used in literature, in



(a) Scenario 1 true clustering results

(b) Scenario 1 clustering results time series

(c) Scenario 2 true clustering results

(d) Scenario 2 clustering results time series

Fig. 1 Example plots of simulated data for the two scenarios ((**a**) and (**b**) denote the spatial units after real clustering and the corresponding time series of the clustering results in scenario 1, respectively, and (**c**) and (**d**) denote the corresponding data in scenario 2, where $n = 100$, $T = 50$, and $C = 5$ are taken)

this algorithm, it is necessary to determine three important parameters, the maximum value of spatial distance, $Eps1$, the maximum value of temporal distance, $Eps2$, and the minimum point, MinPts, which are usually determined by determining the value of $MinPts$ before determining the values of $Eps1$ and $Eps2$ values. First, $MinPts$ can be approximated as the sample size of the data set, which is $MinPts \approx \ln(n)$. Then the average distance between each point and its nearest neighbor is calculated, the corresponding temporal $MinPts$ distances are plotted as well as the spatial $MinPts$ distances, and the values at the “elbows" of the plots are chosen as the corresponding values for $Eps1$ and $Eps2$. The simulation of each method was repeated 50 times for each scenario, and the average value of the 50 simulations was taken as the final experimental result. The spatial and temporal data of the two scenarios from one of the simulations are shown in Fig. 1. The same color represents the same cluster.

The simulation results for each parameter setting in both scenarios are presented in Tables 1, 2, and 3. The average number of clusters, Purity, and FMI of the clustering results are displayed for each method. The simulation results of the proposed method in this paper demonstrate that the average number of clusters closely matches the real situation for both Scenario 1 and Scenario 2. Moreover, the Purity and FMI values

Table 1 Average optimal number of clusters for different models simulated in both scenarios (Sce1 stands for scenario 1, and Sce2 stands for scenario 2. T represents the length of the time series, m represents fuzzy parameter, and L represents lagged $L-$order)

n	Scenario	Class	T	STFCMd-BSWE		MWFC		ST-DBSCAN
				$m = 1.5$	$m = 2$	$L = 10$	$L = 50$	
50	Sce1	$C = 3$	50	3	3	3.4	–	6.5
			100	3	3	4	2	4.6
			500	3	3	2.4	2.45	9
		$C = 5$	50	4.4	4.6	2	–	3
			100	3.8	5	4	2	8
			500	4.8	4.9	2	3	8.4
	Sce2	$C = 3$	50	2.2	3	2	–	7.32
			100	3	3	2	2	11.4
			500	3	3	2	2	6
		$C = 5$	50	4	4.6	5	–	10.88
			100	4.9	5	4.02	4.04	6.06
			500	5	5	2	2	6.08
100	Sce1	$C = 3$	50	3	3	3.15	–	6
			100	3	3	4	3.5	11
			500	2.95	2.98	4.33	2	7.18
		$C = 5$	50	3	4	2	–	11
			100	3.06	4	2	2	12
			500	2	4	2	2	14
	Sce2	$C = 3$	50	3	3	2	–	6.22
			100	3	3	2	2	14.54
			500	3	3	2	2	9.08
		$C = 5$	50	4.9	5	2	–	9.02
			100	4.96	5	4	2	7.46
			500	4	4.04	2	6	8.14

indicate that the proposed method performs optimally in most cases, highlighting its effectiveness compared to the other two methods.

Regarding the fuzzy parameters, the simulations show that the difference in clustering effect between $m = 1.5$ and $m = 2$ is not very obvious. This suggests that the fuzzy parameter does not have a major impact on the clustering effect. However, there are certain parameter settings where the clustering effect is notably better for $m = 2$ than for $m = 1.5$ (e.g., in Scenario 1 with $n = 100$, $T = 50$, and $C = 5$). Furthermore, it is observed that the clustering effect of the other two methods is better for Scenario 1 compared to Scenario 2. This indicates that these methods are not effective when the distinction between clusters is less obvious.

Table 2 Average Purity index of the simulation approach for different models in both scenarios

n	Scenario	Class	T	STFCMd-BSWE		MWFC		ST-DBSCAN
				$m = 1.5$	$m = 2$	$L = 10$	$L = 50$	
50	Sce1	$C = 3$	50	1.0000	1.0000	0.8000	–	0.9056
			100	1.0000	1.0000	0.9820	0.6400	0.9592
			500	1.0000	1.0000	0.7040	1.0000	0.7244
		$C = 5$	50	0.8680	0.9573	0.4600	–	0.3600
			100	0.776	1.0000	0.5660	0.4600	1.0000
			500	0.9696	0.9848	0.5200	0.4400	0.9096
	Sce2	$C = 3$	50	0.7280	0.9800	0.6615	–	0.6680
			100	0.9208	0.9800	0.6996	0.6404	0.8016
			500	0.9976	0.9700	0.7200	0.7100	0.6400
		$C = 5$	50	0.8004	0.9240	0.4188	–	0.7832
			100	0.9816	1.0000	0.4400	0.3424	0.5944
			500	1.0000	1.0000	0.2400	0.2400	0.5732
100	Sce1	$C = 3$	50	1.0000	1.0000	0.7800	–	0.8872
			100	1.0000	1.0000	0.9970	0.7636	0.7590
			500	0.988	0.9952	0.9033	0.7600	0.9812
		$C = 5$	50	0.6200	0.8300	0.2896	–	0.8736
			100	0.6772	0.8500	0.4000	0.4300	0.8268
			500	0.5000	0.8300	0.3995	0.2700	0.8288
	Sce2	$C = 3$	50	0.9567	0.9800	0.6896	–	0.6692
			100	0.9706	0.9900	0.7200	0.7175	0.5244
			500	1.0000	1.0000	0.7440	0.7600	0.5040
		$C = 5$	50	0.9816	0.9992	0.2700	–	0.6294
			100	0.9932	0.9996	0.3662	0.2884	0.5880
			500	0.8400	0.8462	0.2700	0.5300	0.5712

4 Real-life Data Sets

In this paper, we apply the proposed spatio-temporal clustering model BSWE-STFCMd to spatio-temporal data in two different domains: one is traffic data and the other is air quality data. In applying the STFCMd-BSWE model, the fuzzy parameter $m = 2$, the optimal number of clusters is determined by choosing the number of clusters corresponding to the maximum value of the fuzzy silhouette coefficient in Section 2.4.

Example 1: Traffic Data

The dataset used in this study consists of traffic flow data collected from 25 intersections located in Guanshanhu District, Guiyang City, Guizhou Province, China. The data was recorded on April 15, 2020, with a time interval of five minutes, resulting in a total of 288 data points per day. The spatial units in this dataset correspond to the 25 intersections, denoted by $n = 25$, while the length of the time series is $T = 288$.

Table 3 Average FMI of the simulation approach for different models in two scenarios

n	Scenario	Class	T	STFCMd-BSWE		MWFC		ST-DBSCAN
				$m = 1.5$	$m = 2$	$L = 10$	$L = 50$	
50	Sce1	$C = 3$	50	1.0000	1.0000	0.7698	–	0.8536
			100	1.0000	1.0000	0.8830	0.7614	0.9482
			500	1.0000	1.0000	0.8065	0.9031	0.5699
		$C = 5$	50	0.8861	0.9546	0.6419	–	0.4741
			100	0.8531	0.8708	0.8239	0.6576	0.8994
			500	1.0000	1.0000	0.6816	0.7791	0.7632
	Sce2	$C = 3$	50	0.7542	0.9651	0.6969	–	0.8684
			100	0.8608	0.9651	0.7603	0.6667	0.7649
			500	0.9961	0.9452	0.7997	0.7804	0.9023
		$C = 5$	50	0.8507	0.9392	0.5955	–	0.6703
			100	0.9872	1.0000	0.6928	0.6069	0.7371
			500	1.0000	1.0000	0.6366	0.6366	0.6976
100	Sce1	$C = 3$	50	1.0000	1.0000	0.7685	–	0.6469
			100	1.0000	1.0000	0.8799	0.6145	0.4923
			500	0.9910	0.9964	0.8214	0.8064	0.9225
		$C = 5$	50	0.7376	0.8896	0.6458	–	0.6924
			100	0.7330	0.9013	0.6279	0.6510	0.6095
			500	0.5810	0.8896	0.6261	0.6431	0.6145
	Sce2	$C = 3$	50	0.9331	0.9985	0.9844	–	0.9191
			100	0.9399	0.9789	0.7318	0.7282	0.4446
			500	1.0000	1.0000	0.7752	0.8064	0.6472
		$C = 5$	50	0.9872	0.8863	0.5888	–	0.7541
			100	0.9935	0.9992	0.6074	0.5830	0.7623
			500	0.8469	0.8528	0.6431	0.7190	0.7232

These 25 intersections collectively form a road network, and it is essential to partition this network to facilitate traffic management. By dividing the road network into appropriate sub-networks, the management department can gain insights into intersections with high and low traffic flows. This allows them to make informed decisions and implement tailored management strategies for different sub-networks. Such an approach enables more effective traffic flow management, optimization of road conditions, and improved efficiency of road access. Figure 2 displays the geographic locations of the 25 intersections, while Fig. 3 illustrates the corresponding traffic flow observed at these intersections. For narrative purposes, the 25 intersections are numbered as shown in Table 4.

Taking the number of clusters $C = (2, 3, 4, 5, 6)$, the fuzzy silhouette coefficient corresponding to different numbers of clusters is displayed as shown in Fig. 4, when taking the number of clusters $C = 2$, the fuzzy contour coefficients attain the maximum value of $FS = 0.9122$, i.e., it is the most appropriate to divide the road network

Table 4 Numbering of 25 intersections

Intersection	Intersection of Beijing West Road and Jinyuan Street	Intersection of Donglin Temple Road and Changling North Road	Intersection of Donglin Temple Road and Jinyang North Road	Intersection of Yuntan South Road and Shilin East Road	Intersection of Linxi West Road and Yuntan North Road
Number	1	2	3	4	5
Intersection	Intersection of Shilin East Road and Baoxing Road	Intersection of Bihai South Road and Tulip Road	Intersection of Duyun Road and Hubin Road	Intersection of Jinyuan Street and Longquan Yuan Street	Intersection of Jinyang North Road and Jinzhu East Road
Number	6	7	8	9	10
Intersection	Intersection of Jinyang South Road and Shilin East Road	Intersection of Jinyang South Road and Jianhu Road	Intersection of Changling North Road and Donglin Temple Road	Intersection of Changling North Road and Guanshan East Road	Intersection of Changling North Road and Jinzhu East Road
Number	11	12	13	14	15
Intersection	Intersection of Changling South Road and Hefei Road	Intersection of Changling South Road and Guanshan East Road	Intersection of Changling South Road and Yangguan Avenue	Intersection of Qianling Mountain Road and Xingyi Road	Intersection of Qianling Mountain Road and Hubin Road
Number	16	17	18	19	20
Intersection	Intersection of Longquan Yuan Street and Lanzhou Street	Intersection of Longquan Garden Street and Fuzhou Street	Intersection of Guan Shanxi Road and Jinyang South Road	Intersection of Jinyang South Road and Xingzhu West Road	Intersection of Xinzhu West Road and Yuntan North Road
Number	21	22	23	24	25

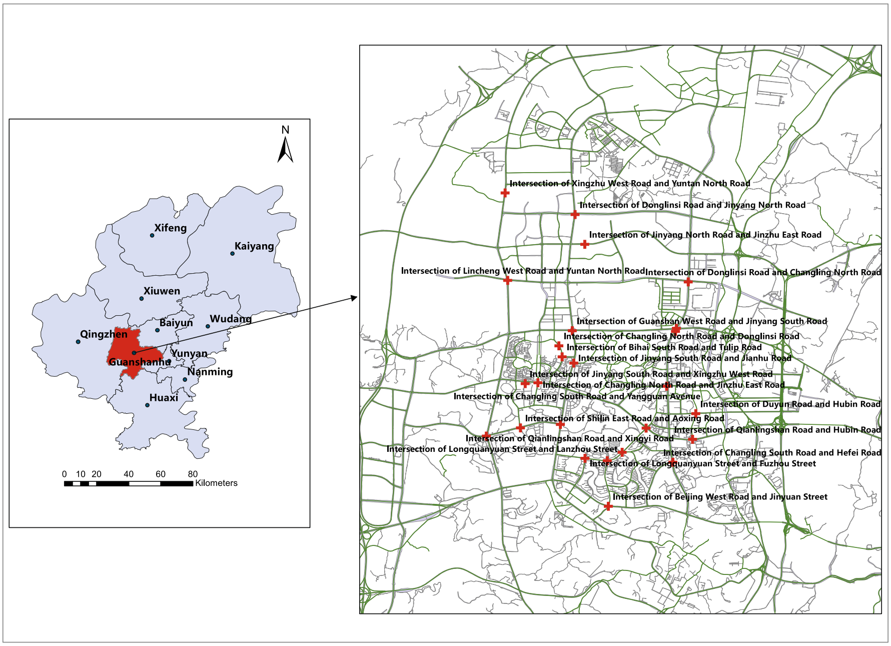

N
Xifeng
Kaiyang
Xiuwen
Baiyun
Wudang
Qingzhen
Guanshanhu
Yunyan
Nanming
Huaxi
0 10 20 40 60 80
Kilometers
Intersection of Xingzhu West Road and Yuntan North Road
Intersection of Donglinsi Road and Jinyang North Road
Intersection of Jinyang North Road and Jinzhu East Road
Intersection of Lincheng West Road and Yuntan North Road
Intersection of Donglinsi Road and Changling North Road
Intersection of Guanshan West Road and Jinyang South Road
Intersection of Changling North Road and Donglinsi Road
Intersection of Bihai South Road and Tulip Road
Intersection of Jinyang South Road and Jianhu Road
Intersection of Jinyang South Road and Xingzhu West Road
Intersection of Changling North Road and Jinzhu East Road
Intersection of Changling South Road and Yangguan Avenue
Intersection of Duyun Road and Hubin Road
Intersection of Shilin East Road and Aoxing Road
Intersection of Qianlingshan Road and Hubin Road
Intersection of Qianlingshan Road and Xingyi Road
Intersection of Longquanyuan Street and Lanzhou Street
Intersection of Changling South Road and Hefei Road
Intersection of Longquanyuan Street and Fuzhou Street
Intersection of Beijing West Road and Jinyuan Street

Fig. 2 Schematic of 25 intersection locations (The red area on the left represents the geographical distribution of the Guanshanhu District under study in Guizhou Province, while the right diagram shows the locations of 25 intersections in the Guanshanhu District)

formed by these 25 intersections into two sub-zones. When the optimal number of clusters is determined, the final weights assigned to space and time are obtained as $w_s = 0.71$ and $w_t = 0.29$, respectively. The final clustering results are also obtained for the membership degree of each intersection as well as the division results as shown in Table 5 and Fig. 5. It is known that the clustering centers are numbered 10 and 11, and the two clusters contain intersections numbered (2, 3, 5, 10, 14, 17, 23, 25) and $(1, 4, 6, 7, 8, 9, 11, 12, 13, 15, 16, 18, 19, 20, 21, 22, 24)$, respectively. The method divides 25 road networks formed by each intersection into two subareas, a high-traffic area, and a low-traffic area.

Example 2: Air Quality Data

The example dataset consists of the daily average Air Quality Index (AQI) for 36 cities in the northeastern region of China from January 1, 2020, to December 31, 2020. This dataset represents spatial units with $n = 36$ cities and a time series length of $T = 366$ days. By performing spatio-temporal clustering based on the AQI values of these 36 cities, we can understand the regional differences and develop appropriate strategies, which are of great significance for mitigating air pollution. The spatial representation of the 36 cities is illustrated in Fig. 6, where the spatial location of each city is indicated by its center latitude and longitude coordinates.

To determine the number of clusters, we calculate the fuzzy silhouette coefficients for different cluster numbers, as shown in Fig. 7. The highest coefficient value $FS = 0.9275$ is achieved when the number of clusters $C = 3$, indicates that the

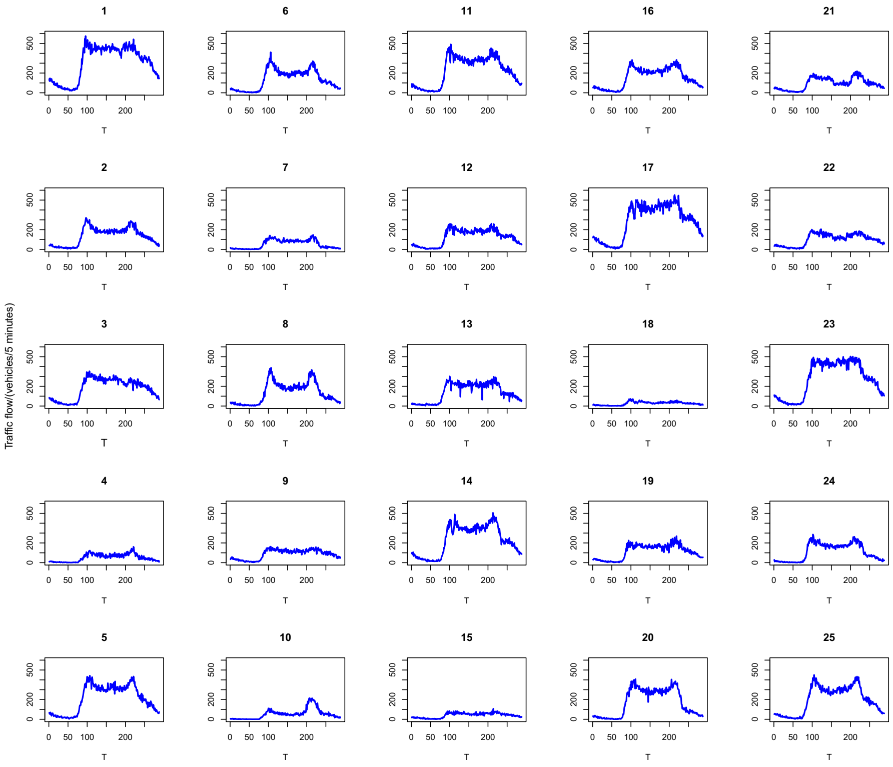


Fig. 3 Plots of traffic flow over time at 25 intersections

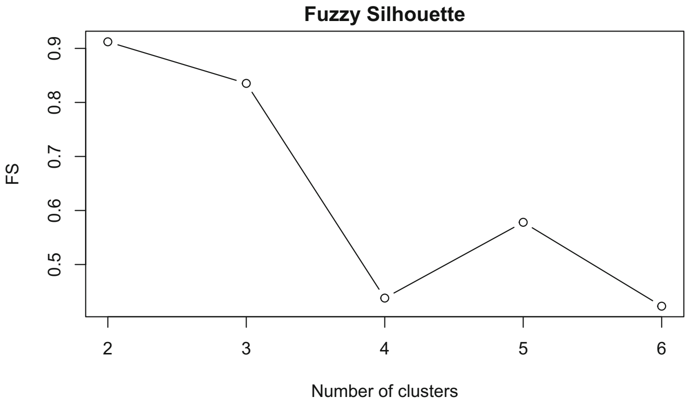


Fig. 4 Fuzzy silhouette coefficient values corresponding to different numbers of clusters C

Table 5 Clustering results: membership degrees of the 25 intersections

Intersection number	Center intersection number	
	10	11
1	0.0961	0.9039
2	0.8674	0.1326
3	0.9693	0.0307
4	0.0357	0.9644
5	0.8888	0.1112
6	0.0125	0.9875
7	0.2730	0.7269
8	0.1135	0.8865
9	0.0385	0.9615
10	**1.0000**	0.0000
11	0.0000	**1.0000**
12	0.2126	0.7874
13	0.3731	0.6269
14	0.5704	0.4296
15	0.0948	0.9052
16	0.0756	0.9244
17	0.5351	0.4649
18	0.1554	0.8446
19	0.0429	0.9571
20	0.0896	0.9104
21	0.0365	0.9635
22	0.0291	0.9709
23	0.5248	0.4752
24	0.0944	0.9056
25	0.9303	0.0697

36 cities should be divided into three clusters. Once the optimal number of clusters is determined, the weights assigned to space and time are $w_s = 0.6685$ and $w_t = 0.3315$, respectively. This suggests that spatial proximity has a greater influence on the clustering results compared to temporal proximity. The final clustering results are presented in Fig. 8 and Table 6.

Table 6 shows the membership degrees of the 36 cities belonging to each cluster, and from the results, the whole Northeast region is divided into three clusters according to the average daily AQI values. The center cities of the three clusters are Jilin City in Jilin Province, Anshan City in Liaoning Province, and Hegang City in Heilongjiang Province, and the clustering is centered on one of the cities in each province, which further indicates that the spatio-temporal clustering in the model is effective and consistent with common sense. It can also be seen that Yingkou City in Liaoning Province belongs to the second cluster with a high membership degree of

Table 6 Clustering results: membership degrees of the 36 cities

City	Center city		
	Jilin	Anshan	Hegang
Shenyang	0.3981	0.5292	0.0728
Dalian	0.1642	0.7413	0.0945
Anshan	0.0000	**1.0000**	0.0000
Fushun	0.6677	0.2578	0.0745
Benxi	0.3576	0.5664	0.0760
Dandong	0.2655	0.6070	0.1276
Jinzhou	0.0529	0.9235	0.0236
Yingkou	0.0159	0.9777	0.0064
Fuxin	0.2302	0.6739	0.0959
Liaoyang	0.0517	0.9321	0.0162
Panjin	0.1795	0.7575	0.0630
Tieling	0.7039	0.2302	0.0659
Chaoyang	0.1045	0.8421	0.0535
Huludao	0.0676	0.8983	0.0341
Changchun	0.8543	0.0723	0.0734
Jilin	**1.0000**	0.0000	0.0000
Siping	0.7187	0.1845	0.0968
Liaoyuan	0.8852	0.0746	0.0402
Tonghua	0.2458	0.6107	0.1435
Baishan	0.3321	0.4686	0.1993
Songyuan	0.6990	0.1393	0.1617
Baicheng	0.3769	0.2544	0.3688
Yanbian Korean Autonomous Prefecture	0.3650	0.2157	0.4193
Harbin	0.6388	0.1716	0.1896
Qiqihar	0.3169	0.1611	0.5220
Jixi	0.1370	0.0748	0.7882
Hegang	0.0000	0.0000	**1.0000**
Shuangyashan	0.0892	0.0377	0.8730
Daqing	0.3352	0.2689	0.3959
Yichun	0.0856	0.0414	0.8729
Jiamusi	0.1139	0.0683	0.8178
Qitaihe	0.2849	0.0928	0.6223
Mudanjiang	0.5591	0.1059	0.3350
Heihe	0.1602	0.1295	0.7104
Suihua	0.5999	0.1644	0.2357
Daxinganling	0.2243	0.1864	0.5893

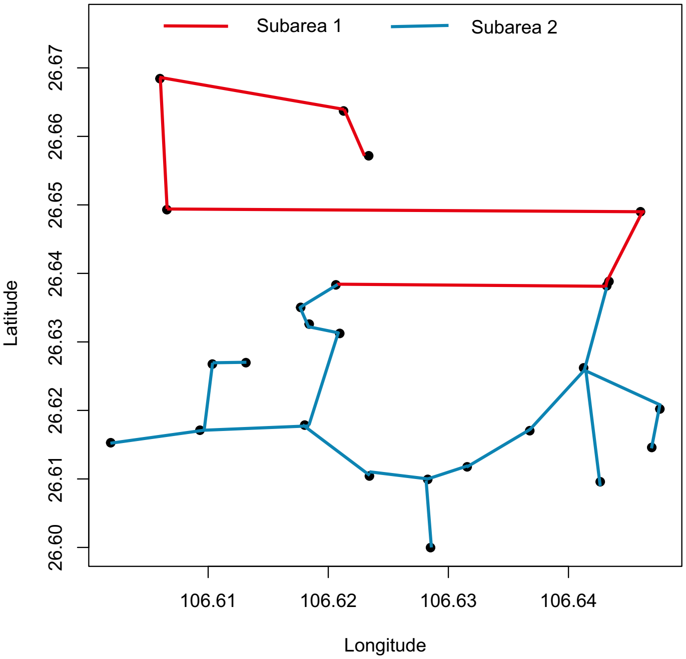


Fig. 5 Clustering results: two subareas

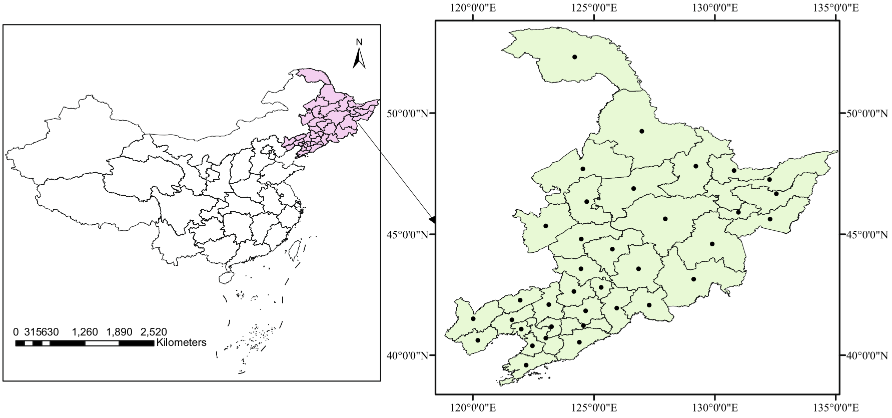


Fig. 6 Map of the distribution of 36 cities: black dots indicate the center of each city(The data set is provided by Geographic remote sensing ecological network platform (www.gisrs.cn)

0.9777, and Liaoyang City in Liaoning Province belongs to the second cluster with a membership degree of 0.9321, which indicates that it is reasonable to classify these two cities in the second cluster. Figure 8 shows the spatial locations of the 36 cities after clustering, and the cities with the same color belong to the same cluster, which indicates that these cities have similar spatial and temporal variations in air quality in a year. Obvious differences in air quality levels can also be found between cities in

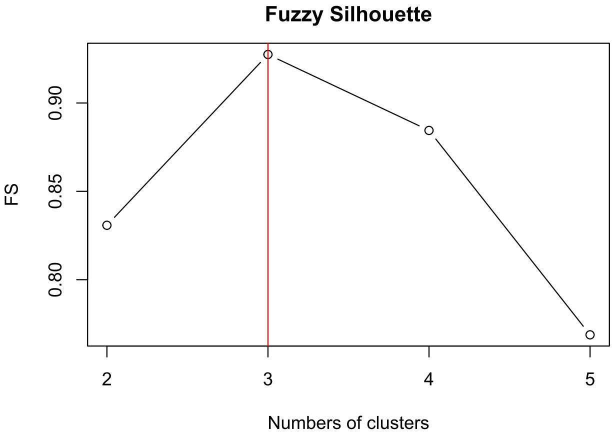


Fig. 7 Fuzzy silhouette coefficient values corresponding to different numbers of clusters *C*

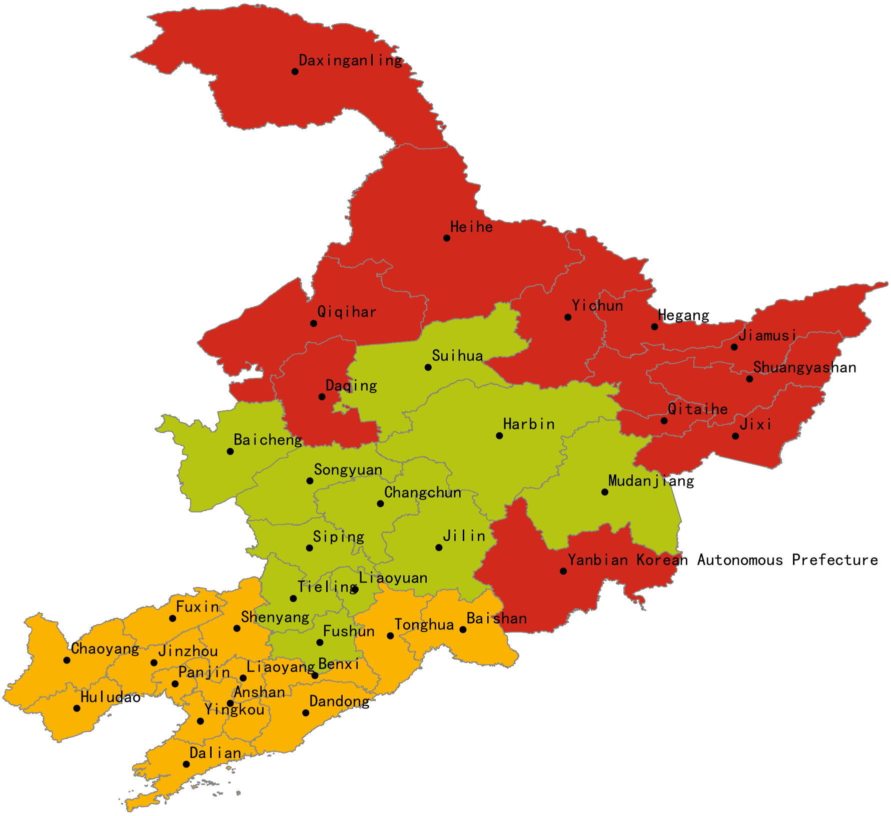


Fig. 8 Clustering results: divided into three clusters

Jilin Province, and the Yanbian Korean Autonomous Prefecture in Jilin Province is more different from other cities in the province. The air quality levels in Fushun City, Liaoning Province, are similar to those in Jilin Province.

5 Conclusion

This paper proposes a new robust clustering method for complex spatio-temporal data, addressing the need to cluster spatial units with similar time series. Given the fact that spatio-temporal clustering is usually affected by the dependence on time and space, a weighted exponential function is used to describe the correlation of spatio-temporal data. Additionally, a B-splines-based expansion is employed to reconstruct the spatio-temporal data. Further, the Partitioning Around Medoids (PAM) algorithm is introduced to accelerate the implementation of the proposed clustering method. In summary, the proposed method can effectively characterize the correlation of spatio-temporal data and is also easy to understand and implement.

Compared with several competing spatio-temporal clustering methods, the simulation studies show that the proposed method generally exhibits better performance in terms of purity and FMI. The method's extensive applicability is also demonstrated through two real-world applications involving traffic and air quality data. Furthermore, the proposed method can be extended to cater to a diverse range of requirements, including the clustering of spatio-temporal data from mobile monitors.

Our work is subject to limitations that need further study. Specifically, the depiction of spatio-temporal data correlation relies on a specific kernel form, such as the weighted exponential function employed in this study. The choice of the kernel function holds significance for the clustering approach and is frequently influenced by our comprehension of the data.

Author Contributions Feiyun Wang: Visualization, methodology, algorithm, coding, writing-original draft, writing-review and editing. Wangyong Chen: Methodology, algorithm, reviewing, and editing. Yao Hu: Idea, reviewing, and editing. Yewen Chen: Methodology, reviewing and editing, polishing the manuscript.

Funding This work was supported by the National Natural Science Foundation of China (Grant Nos. 12161016 and 11661018) and the Guizhou Provincial Basic Research Program (Natural Science) (No. QianKeHeJiChu-ZK[2024]YiBan082).

Availability of data The datasets exemplified in this paper are not public due to data non-disclosure reasons but are available from the corresponding author upon reasonable request.

Declarations

Conflict of interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Ethical approval The manuscript was not submitted to more than one journal for simultaneous consideration. This article does not contain any studies with human participants or animals performed by any of the authors.

References

Abraham C, Cornillon PA, Matzner-Løber E et al (2003) Unsupervised curve clustering using b-splines. Scand J Stat 30(3):581–595

Ann Maharaj E, D'Urso P, Galagedera DU (2010) Wavelet-based fuzzy clustering of time series. J Classif 27:231–275

Ansari MY, Ahmad A, Khan SS et al (2020) Spatiotemporal clustering: a review. Artif Intell Rev 53:2381–2423

Birant D, Kut A (2007) St-dbscan: an algorithm for clustering spatial-temporal data. Data Knowl Eng 60(1):208–221

Cammalleri C, Acosta Navarro JC, Bavera D et al (2023) An event-oriented database of meteorological droughts in europe based on spatio-temporal clustering. Sci Rep 13(1):3145

Campello RJ (2007) A fuzzy extension of the rand index and other related indexes for clustering and classification assessment. Pattern Recogn Lett 28(7):833–841

Campello RJ, Hruschka ER (2006) A fuzzy extension of the silhouette width criterion for cluster analysis. Fuzzy Sets Syst 157(21):2858–2875

de Boor C (2001) A Practical Guide to Spline. Springer

Deb S, Karmakar S (2023) A novel spatio-temporal clustering algorithm with applications on covid-19 data from the united states. Comput Statist Data Anal 18:107810

Durso P, Massari R (2019) Fuzzy clustering of mixed data. Inf Sci 505:513–534

Durso P et al (2022) Spatial robust fuzzy clustering of covid 19 time series based on b-splines. Spatial Statistics 49:100518

Faloutsos C, Ranganathan M, Manolopoulos Y (1994) Fast subsequence matching in time-series databases. ACM SIGMOD Rec 23(2):419–429

Gao X, Yu F (2016) Fuzzy c-means with spatiotemporal constraints. In: 2016 International Symposium on Computer, Consumer and Control (IS3C), IEEE, pp 337–340

Gaudart J, Poudiougou B, Dicko A et al (2006) Space-time clustering of childhood malaria at the household level: a dynamic cohort in a mali village. BMC Public Health 6:1–13

Huang JZ, Shen H (2004) Functional coefficient regression models for non-linear time series: a polynomial spline approach. Scand J Stat 31(4):515–534

Keogh E, Chakrabarti K, Pazzani M et al (2001) Dimensionality reduction for fast similarity search in large time series databases. Knowl Inf Syst 3:263–286

Kulldorff M, Heffernan R, Hartman J et al (2005) A space-time permutation scan statistic for disease outbreak detection. PLoS Med 2(3):e59

Liu Z, George R (2003) Fuzzy cluster analysis of spatio-temporal data. In: International Symposium on Computer and Information Sciences, Springer, pp 984–991

Lurka A (2021) Spatio-temporal hierarchical cluster analysis of mining-induced seismicity in coal mines using ward's minimum variance method. J Appl Geophys 184:104249

Maharaj EA, D'Urso P (2011) Fuzzy clustering of time series in the frequency domain. Inf Sci 181(7):1187–1211

Mattera R (2022) A weighted approach for spatio-temporal clustering of covid-19 spread in italy. Spatial and Spatio-temporal Epidemiology 41:100500

McBratney AB, Moore AW (1985) Application of fuzzy sets to climatic classification. Agric For Meteorol 35(1–4):165–185

Moayedi A, Abbaspour RA, Chehreghan A (2019) An evaluation of the efficiency of similarity functions in density-based clustering of spatial trajectories. Ann GIS 25(4):313–327

Mondal B, Choudhury JP (2013) A comparative study on k means and pam algorithm using physical characters of different varieties of mango in india. Int J Computer Appl 78(5):21–24

Pei T, Zhou C, Zhu AX et al (2010) Windowed nearest neighbour method for mining spatio-temporal clusters in the presence of noise. Int J Geogr Inf Sci 24(6):925–948

Ren W, Jian H (2021) The comprehensive evaluation of "five aspects" based on coefficient-of-variation-modified g1 combination weighting. In: Sensor Networks and Signal Processing: Proceedings of the 2nd Sensor Networks and Signal Processing (SNSP 2019), 19-22 November 2019, Hualien, Taiwan, Springer

Tork HF (2012) Spatio-temporal clustering methods classification. Faculdade de Engenharia da Universidade do Porto Porto, Portugal, Doctoral symposium on informatics engineering

Wang M, Wang A, Li A (2006) Mining spatial-temporal clusters from geo-databases. In: International Conference on Advanced Data Mining and Applications, Springer, pp 263–270

Wu KL, Yang MS (2002) Alternative c-means clustering algorithms. Pattern Recogn 35(10):2267–2278

Yürüşen NY, Uzunoğlu B, Talayero AP et al (2021) Apriori and k-means algorithms of machine learning for spatio-temporal solar generation balancing. Renewable Energy 175:702–717

Zaliapin I, Gabrielov A, Keilis-Borok V et al (2008) Clustering analysis of seismicity and aftershock identification. Phys Rev Lett 101(1):018501

Zhang DQ, Chen SC (2004) A comment on alternative c-means clustering algorithms. Pattern Recognition 37(2):173–174

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.