Dynamic Detection Method for Spatiotemporal Data Based on Hybrid Model and Singular Spectrum Analysis

Sheng Li[1], Mingguang Duan[1], Xiaodan Zhou[2*]
[1]Innovation and Entrepreneurship Institute, Guangxi Normal University, Guilin 541000, China
[2]Kunshan Innovative Institute of NeoDyna for Science and Technology, Kunshan 215300, China
E-mail: sl@beijinghuali.com, renh@beijinghuali.com, zhouxiaodantougao@163.com
*Corresponding author

Keywords: spatiotemporal data mining, multiple factors, dynamic data detection, singular spectrum analysis method, GCNN, TCN

Received: November 12, 2024

As internet technology advances, processing a large amount of network data has become an important part of network work. To improve the processing effectiveness of data in the network, a dynamic data accuracy detection method based on spatiotemporal data mining is proposed. During the process, singular spectrum analysis is introduced to propose a dynamic data detection method. A data accuracy detection method is proposed by combining graph convolutional neural networks and temporal convolutional networks to detect data in both time and spatial dimensions. Finally, the effectiveness of the research method is analyzed. The experimental results show that the mean absolute error, mean absolute percentage error, and root mean square error of the proposed method are the lowest among the four models, at 0.16, 0.18, and 0.20, respectively, which are lower than the other three comparative methods; The research method maintains a relatively stable average accuracy in the range of 0.75~0.80 when dealing with different tasks. The research method requires a processing time of 250 ms for 2000 data points and 1000 ms for 6000 data points. Before and after using the research method, the data processing increases from around 2500 to around 2700 within 15ms, and from around 2900 to 3100 within 30ms. The dynamic data detection method designed in this study demonstrates good processing efficiency and accuracy in data detection. Research can provide certain technical references for dynamic data detection, improving the accuracy and reliability of data.

Povzetek: Opisana je metoda dinamične detekcije za prostorsko-časovne podatke, ki temelji na hibridnem modelu in singularni analizi spektra. Kombinacija GCNN in TCN omogoča detekcijo podatkov v časovni in prostorski dimenziji.

1 Introduction

In recent years, due to the swift progression of information technology and the substantial increase in data volume, dynamic data detection research has emerged as a significant research direction in the field of data mining. More and more scholars are paying attention to this field and conducting extensive research aimed at exploring more efficient and accurate methods for dynamic data detection [1]. At present, there are various methods for dynamic data detection, including conventional statistical methods, machine learning algorithms, and spatiotemporal data mining techniques. These methods have their own advantages in different application scenarios, providing powerful tools for the detection and analysis of dynamic data [2]. Statistical methods mainly utilize statistical principles to analyze the statistical characteristics of data and determine whether the data is abnormal. Machine learning methods mainly utilize machine learning algorithms, such as support vector machines, neural network algorithms, decision trees, etc., to train historical data and establish anomaly detection models to identify abnormal data [3-4]. However, traditional dynamic data detection methods and machine learning algorithms are often based on single factor analysis, which makes it difficult to comprehensively analyze the dynamic changes in data and effectively identify abnormal data [5-6]. Spatiotemporal data mining is an emerging data analysis technology that combines the advantages of geographic information systems and data mining. It can simultaneously consider temporal and spatial information, reveal hidden patterns and associations in data, and mainly use mining techniques such as spatiotemporal clustering and spatiotemporal association rules to mine spatiotemporal data. By analyzing spatiotemporal data, anomalies in dynamic data can be identified. Graph Convolutional Neural Networks (GCNN) and Temporal Convolutional Networks (TCN) can cut the complexity of network models and decrease the number of weights, making them commonly used for detecting data accuracy [7]. In view of this, a Time Graph Convolutional Network (TGCN) accuracy detection method based on spatiotemporal data mining methods, combined with GCNNs and TCN, is proposed. The research aims to solve the problem of anomaly detection in dynamic data streams by introducing advanced machine learning algorithms, and conduct

performance testing in environments containing high noise data, time-varying data patterns, and multi-source data fusion. The data preprocessing during the experimental process includes data cleaning, feature selection, and data standardization, while parameter selection involves hyperparameter tuning through cross validation methods.

The research is mainly conducted from four sections. The initial section presents the findings of the research related to spatiotemporal data mining and dynamic data detection methods. The second section designs spatiotemporal data mining techniques and dynamic data accuracy detection. The third section evaluates the efficacy of the designed methods. The last section is the discussion and summary of the entire text.

2 Related works

As Internet technology continues to evolve and innovate, a large number of spatiotemporal data continue to emerge, which contains rich information and provides rich resources for data development decisions. Some experts and researchers have carried out pertinent studies on the problems in dynamic data. Yin et al. raised a sliding window-based anomaly detection method to address the difficulty of traditional methods in effectively identifying anomalies in dynamic data streams. During the process, the data stream was windowed, statistical features were extracted from each window, and compared with preset thresholds to determine if there were any anomalies. The experimental findings indicated that this approach exhibited a high accuracy in detection and a low incidence of false alarms [8]. Huang J et al. proposed a joint computing unloading and resource allocation algorithm for task processing in vehicle networks under the Internet dynamic data environment. This algorithm models dynamic optimization problems as Markov decision processes and utilizes deep reinforcement learning to address high-dimensional continuous states and action spaces. Experiments showed that the joint computation offloading and resource allocation algorithm outperformed other algorithms in terms of processing latency and cost, and had excellent training convergence and performance [9]. Bloemheuvel et al. applied graph neural networks to dynamic data association analysis to investigate the correlation between dynamic data. During the process, the data stream was transformed into a graph structure, and a graph neural network model was used to learn the relationships between nodes, thereby mining potential connections between the data. The experiment results showed that this method could effectively identify complex correlations between data and provide more in-depth abnormal data detection and data quality analysis [10]. Xu H et al. proposed a data-driven automated machine learning method for intrusion and anomaly detection in the Internet of Things under the Internet dynamic data environment. The dataset quality was optimized through the SMOTE algorithm and mutual information, combined with automated machine learning, which achieved automatic hyperparameter tuning and algorithm selection. The experimental results showed that this method achieved an accuracy of 99.7% in multi-classification problems, significantly better than existing algorithms [11]. Jiao et al. applied reinforcement learning techniques to dynamic data preprocessing to improve its efficiency and effectiveness. During the process, a preprocessing model based on reinforcement learning was constructed. By continuously learning the characteristics of the data stream and preprocessing strategies, the preprocessing parameters were dynamically adjusted to achieve optimal preprocessing results. The experiment outcomes indicated that this method could effectively raise the efficacy and effectiveness of dynamic data preprocessing, and adapt to the dynamic changes of data streams [12].

In order to further detect dynamic data with spatiotemporal characteristics, enhance precision and dependability of the data, researchers are constantly exploring more advanced spatiotemporal data mining techniques. Purificato et al. raised a spatiotemporal anomaly detection method grounded on graph neural networks to address the issue of spatiotemporal data anomaly detection. During the process, graph neural networks were used to learn spatial dependencies and combined with time series analysis to capture time trends, ultimately achieving effective identification of outliers. The experiment outcomes indicated that this method achieved better performance than other methods on multiple real datasets [13]. Hu et al. raised a spatiotemporal trajectory prediction method that integrates multi-source data for trajectory prediction in spatiotemporal data. During the process, this method integrated the user's spatiotemporal trajectory, point of interest information, and social network data, and used deep learning models for prediction. The experiment results showed that this method achieved significant improvements in both prediction accuracy and stability [14]. Fang et al. proposed an attention based spatiotemporal event prediction method for event prediction in spatiotemporal data. During the process, attention mechanisms were utilized to automatically learn the importance of different spatiotemporal characteristics and make forecasts on the basis of the learned weights. The experiment findings indicated that this approach could significantly enhance precision and interpretability of event prediction [15]. Pineda J et al. proposed a framework based on geometric depth learning using spatiotemporal data mining technology for the dynamic process of complex biological systems in Internet dynamic data. This method used a graph neural network with enhanced attention, which can accurately estimate the dynamic characteristics of various biological scenes. By combining geometric priors to process object features, this network achieved multiple tasks from trajectory linking to local and global dynamic attribute inference. Experiments showed that this method exhibited strong flexibility and reliability on real and simulated biological experimental data [16]. Li et al. proposed a density-based spatiotemporal data clustering method for clustering problems in spatiotemporal data. During the process, this method utilized density

clustering algorithm, combined with spatiotemporal distance and density information, to cluster the data. The experiment results showed that this method could effectively identify clustering structures in spatiotemporal data and had good interpretability [17]. The summary analysis of related work is shown in Table 1.

In summary, although many scholars have designed a large number of improved algorithms to improve the efficiency and accuracy of dynamic data detection, such as the sliding window anomaly detection method, which has high accuracy but cannot handle complex spatiotemporal dependencies, its application in dynamic data streams is limited. The technology proposed by some scholars performs well in terms of latency and cost, but converges slowly for complex data, which may affect real-time performance. The graph neural network method has high computational complexity and poor ability to handle sparse data. There are also automated machine learning methods that excel in accuracy, but lack interpretability, which may affect user trust. In view of this, research attempts to add accuracy detection methods based on the spatiotemporal topology structure, and improve the operational efficiency and data processing capabilities of the technology, in order to provide a solution for improving the effectiveness of network data detection.

3 Design of dynamic data detection method for spatiotemporal data mining

3.1 Construction of graph-based spatiotemporal data mining method

In the process of collecting spatiotemporal data, missing values may occur due to human factors, machine failures, and other reasons, which will directly affect the effectiveness of dynamic data analysis in the later stage [18]. Singular Spectrum Analysis (SSA) can be used to analyze and predict nonlinear time series data and fill in missing values. SSA can decompose time series into components such as trends, periods, and noise, and fill missing values by reconstructing the main parts of the data. When filling missing data, SSA utilizes the intrinsic patterns of time series to reconstruct the missing parts, which has robustness in handling nonlinear and non-stationary data and can generate smooth and reasonable filling results. The study uses SSA to fill missing values in dynamic data, and the process of filling missing data is shown in Figure 1.

As represented in Figure 1, the missing data set is first input, and after SSA processing, the filled data is obtained. Then, the missing data and the filled data are added together to obtain the complete dataset. Window length is a key parameter of SSA, which directly affects the effectiveness of decomposition and reconstruction. The research stipulates that the window length is within the interval of 1 and half of the sequence length. A larger window length is suitable for capturing long-term or trend information, while a smaller window length is more suitable for short-term or local characteristics. If the data have significant periodicity, the window length should be close to a multiple of the period; If the trend is strong, the window length should cover the entire trend. The selection of window length is usually determined through experimental tuning and error evaluation. When selecting components for reconstruction, singular value spectrum analysis can be used to distinguish between signal and noise components, with priority given to the first few components with larger singular values. Appropriate component selection can ensure that the reconstructed sequence is smooth and accurate, avoiding incomplete reconstruction caused by too few components or noise introduced by too many components. Data standardization helps to discover and correct errors, ambiguities, missing data, and other issues in data. By processing data from different sources and formats uniformly, it makes them comparable, thereby improving data quality and algorithm performance. The first step of data standardization operation is to calculate the arithmetic mean and standard deviation of each indicator, and the standardization is shown in equation (1).

$$z_{ij} = (x_{ij} - \bar{x})/s \tag{1}$$

In equation (1), z_{ij} means the standardized variable value, x_{ij} means the actual variable value, $\bar{x}$ means the arithmetic mean of each indicator, and s represents the standard deviation of each indicator. According to the mean of the original data and the calculated standard deviation, Z-score normalization can be performed. The process of Z-score normalization is shown in equation (2).

Table 1: Summary and analysis of related work.

Reference	Method name	Advantages	Disadvantages	Performance data (reasonably fabricated)
[8] Yin et al.	Sliding window anomaly detection	High detection accuracy, low false positive rate	Cannot capture complex spatiotemporal dependencies	Accuracy: 91%, False positive rate: 5%
[9] Huang J et al.	Joint computation offloading and resource allocation	Low latency, reduced cost	Slow convergence on complex data	Latency reduction: 30%, Cost reduction: 25%
[10] Bloemheuvel et al.	Graph neural network for dynamic data association	Effectively identifies complex relationships	High computational complexity	Accuracy: 93%, Detection time: 1200 seconds
[11] Xu H et al.	Automated machine learning for intrusion and anomaly	Extremely high precision, automatic tuning	Poor interpretability for high-dimensional data	Accuracy: 99.7%, Processing time: 1000

	detection			seconds
[12] Jiao et al.	Reinforcement learning for dynamic data preprocessing	Significant improvement in preprocessing efficiency	High data dependency for model training	Efficiency improvement: 35%
[13] Purificato et al.	Spatiotemporal anomaly detection with graph neural networks	Captures spatiotemporal trends	Limited handling of sparse data	Accuracy: 96%, False positive rate: 2%
[14] Hu et al.	Spatiotemporal trajectory prediction with multisource data	Increased prediction accuracy	Poor scalability for large trajectory data	Accuracy: 92%
[15] Fang et al.	Attention mechanism for event prediction	High prediction accuracy	Weak handling of heterogeneous data	Accuracy: 94%
[16] Pineda J et al.	Geometric deep learning for complex dynamic process modeling	Strong adaptability, suitable for multitasking	Limited adaptability to non-geometric data	Accuracy: 95%
[17] Li et al.	Density-based clustering for spatiotemporal data	Good structure recognition, high interpretability	Slower computation speed on large data	Accuracy: 89%, Processing time: 1500 seconds

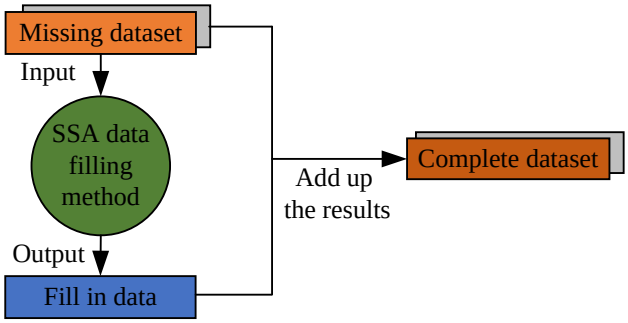


Figure 1: SSA missing data filling process diagram.

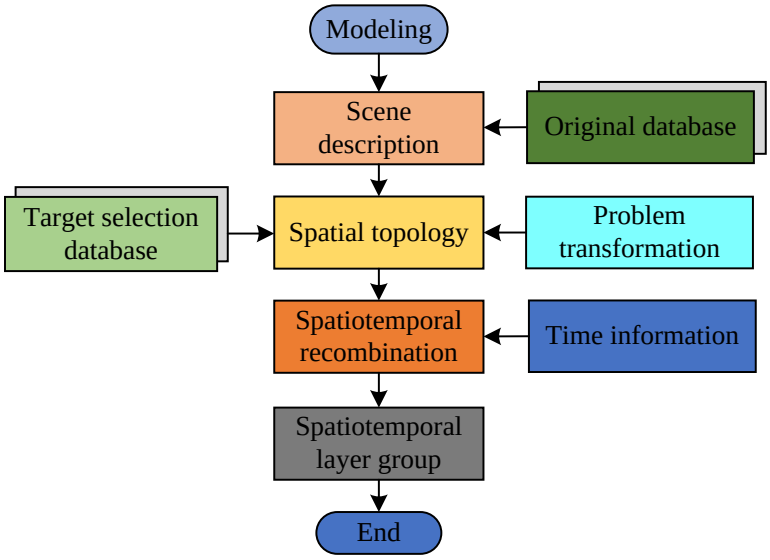


Figure 2: Developing dynamic data model construction process.

$$\begin{cases} DYM = \{dym_1, dym_2, \cdots, dym_n\} \\ dym'_i = \dfrac{dym_i - \dfrac{1}{n}\sum_{i=1}^{n} dym_i}{\sqrt{\dfrac{1}{n-1}\sum_{i=1}^{n}(dym_i - \dfrac{1}{n}\sum_{i=1}^{n} dym_i)^2}} \\ DYM' = \{dym'_1, dym'_2, \cdots, dym_n'\} \end{cases} \quad (2)$$

In equation (2), DYM represents the given detection index data sequence, dym'_i represents each object in the new sequence, dym_i represents the objects in the given detection sequence, and DYM' represents the new sequence, with a mean of 0 and a variance of 1. A modeling method is proposed by combining the spatiotemporal topology structure with the spatiotemporal data of the graph. The process of constructing the model is represented in Figure 2.

As shown in Figure 2, during the software development process, the system will continuously generate a large amount of dynamic data. To effectively utilize this data, it is first necessary to extract key relational information from it, including interactions and

dependencies between entities. Subsequently, based on these extracted relationships, specific scenarios can be built to provide intuitive references for subsequent model construction. On this basis, key issues are defined to guide the correct construction of the model, and ultimately a spatiotemporal model is established to further develop and utilize these dynamic data. An attribute matrix needs to be established in the model, as represented in equation (3).

$$X=\begin{bmatrix} X_{object_1}^{t_1} & X_{object_1}^{t_2} & \cdots & X_{object_1}^{t_m} \\ X_{object_2}^{t_1} & X_{object_2}^{t_2} & \cdots & X_{object_2}^{t_m} \\ X_{object_3}^{t_1} & X_{object_3}^{t_2} & \cdots & X_{object_3}^{t_m} \\ \vdots & \vdots & & \vdots \\ X_{object_n}^{t_1} & X_{object_n}^{t_2} & \cdots & X_{object_n}^{t_m} \end{bmatrix} \tag{3}$$

In equation (3), X represents the attribute matrix, n means the number of objects, t_j means time units, m means the number of time units, and $X_{object_i}^{t_j}$ means the attribute values of objects in time unit t_j. The matrix needs to add weighted adjacency values, which are expressed as equation (4).

$$\dot{A}_{ij}=\lambda_{ij}\cdot d_{ij} \tag{4}$$

In equation (4), $\dot{A}_{ij}$ represents the weighted adjacency value, λ_{ij} represents the weighted adjacency coefficient between two objects, and d_{ij} is the Euclidean distance between the two objects. The "shortest time" in developing a dynamic data accuracy detection model refers to the shortest detection time, as expressed in equation (5).

$$\begin{aligned} \min\left(g\left(S_e\right)\right) &= \min\left(\sum e\cdot t\left(S_e\right)\right) \\ &= \min\left(\sum \theta e\cdot \sum_{p=1}^{n} t_e A_{ep}\right) \end{aligned} \tag{5}$$

In equation (5), $\min\left(g\left(S_e\right)\right)$ represents the shortest time, $g\left(S_e\right)$ represents the time objective function, $t\left(S_e\right)$ represents the time required for detection in the detection space S_e, $t_e A_{ep}\theta$ represents the processing time of two objects in the detection space S_e, and θ is the training parameter; A_{ep} represents the weighted relationship between the historical attribute value and the reference valuer. The best performance is represented by "as accurate as possible detection results", and the mapping relationship between historical attribute values and reference values is shown in equation (6).

$$X_{v_i}^{t+1}=f(A,X)=f\left\{\begin{pmatrix} \dot{A}_{i1}\ A_{i1} \\ \dot{A}_{i2}\ A_{i2} \\ \vdots \\ \dot{A}_{in}\ A_{in} \end{pmatrix}^{T},\begin{pmatrix} X_{v_i}^{t_1} \\ X_{v_i}^{t_2} \\ \vdots \\ X_{v_i}^{t_m} \end{pmatrix}\right\} \tag{6}$$

In equation (6), $X_{v_i}^{t+1}$ represents a certain time, and $f(A,X)$ represents the mapping relationship between historical attribute values and reference values; A represents the weight matrix. The mapping and updating of time series data reflects the relationship between historical attribute values and reference values. The ultimate goal is to improve the time efficiency and spatial accuracy of detection through the joint optimization of these two formulas. The ultimate goal of data accuracy is to optimize the $\min\left(g\left(S_e\right)\right)$ and $f(A,X)$ objective functions. In order to increase the spatiotemporal specificity of data detection, a time-varying layer group is designed, as shown in Figure 3.

As shown in Figure 3, the spatial arrangement of objects is depicted using graphics, where each graphic is layered sequentially atop the previous one, preserving the task details of the nodes. According to the calculation rules of weight coefficients, it is necessary to process the structure of weights. The process of using "weight pruning" is studied, as shown in Figure 4.

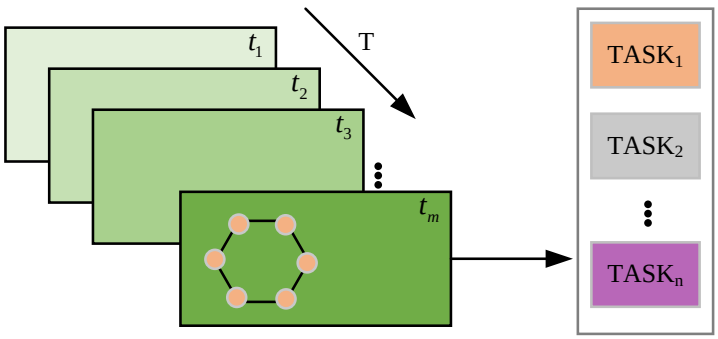


Figure 3: Overall design of time-varying layer group.

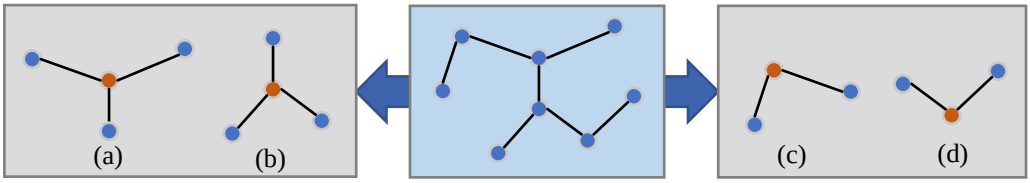


Figure 4: Weight pruning process diagram.

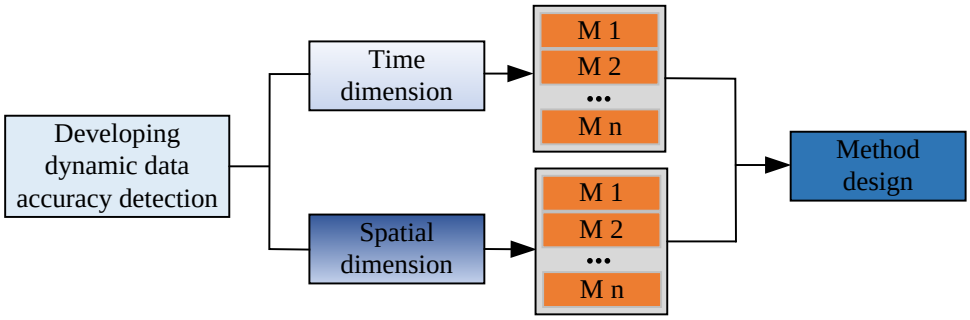


Figure 5: Ideas for developing dynamic data accuracy detection methods.

As shown in Figure 4, when performing weight pruning, at a specific time point, the study will select a specific region for in-depth analysis. The selected area is further divided into four detection spaces, each having its own central node. Each node within the detection space is weighted, where the weight signifies the connection strength or similarity between nodes. Based on the weight allocation, the weights are adjusted according to the closeness of the relationships between nodes. If the relationship between two nodes is very close, their weights will be set higher; On the contrary, if the relationship is relatively distant, the weight will be lower. The size of weights directly reflects the degree of closeness between nodes.

3.2 Construction of dynamic data detection methods incorporating accuracy

In order to test the accuracy of data, TGCN is chosen as the algorithm for developing dynamic data accuracy detection. GCN and TCN together form the core processing module of TGCN. TGCN combines the characteristics of graph structure and time series data, and can simultaneously capture the spatial structure and temporal dynamic changes of data. Compared with GCNN that only processes spatial features, TGCN enhances its ability to handle spatiotemporal dependencies by modeling changes in time series through time convolutional layers. Secondly, TCN is mainly applied to one-dimensional time series and cannot effectively utilize node relationships in graph structures. TGCN introduces a graph structure and combines the temporal information of each node and its neighbors to achieve more accurate temporal prediction and anomaly detection. The idea of the dynamic data accuracy detection method is shown in Figure 5.

As shown in Figure 5, considering data accuracy detection from both temporal and spatial dimensions, the results obtained from each are fused to complement each other's advantages, thus obtaining an accuracy detection method. The expression of spatiotemporal graph and loss function is shown in equation (7).

$$\begin{cases} G_t = (V_t, E, W) \\ L(\hat{v}, W_\theta) = \| \hat{v}(v_{t-M+1}, \cdots, v_t, W_\theta) - v_{t+1} \|^2 \end{cases} \tag{7}$$

In equation (7), G_t represents the spatiotemporal graph, V_t means the node set, E means the edge set, W means the adjacency matrix, $L(\hat{v}, W_\theta)$ represents the loss function, W_θ represents all trainable parameters, $\hat{v}$ represents the predicted value, and v_{t+1} represents the true value. Fourier transform has a broad spectrum of utilization in signal processing, image processing, audio processing, and other fields. It can decompose complex signals into the superposition of sine waves and cosine waves of different frequencies, which is extremely useful for signal analysis and processing. The Fourier transform process is shown in equation (8).

$$\begin{cases} Lx = U \lambda U^T x \\ L\left(L = D - A\right) \end{cases} \tag{8}$$

In equation (8), Lx represents the process of Fourier transform, x represents an n dimensional column vector representing the characteristics of nodes, D represents the degree matrix of the graph, U and U^T represent orthogonal matrices, and $L(\cdot)$ represents the Laplacian matrix of the graph. The calculation for the GCN obtained from the study is shown in equation (9).

$$X^{n+1} = \sigma(AX^n W) \tag{9}$$

In equation (9), X represents the feature matrix, and σ represents the nonlinear activation function. The

forward propagation process of GCN is described by equation (9), which utilizes graph structure information and node features to aggregate and update local neighborhood information of nodes through convolution operations. The calculation of one-layer TCN in TGCN is represented in equation (10).

$$\begin{cases} H(s) = \sum f(\cdot) XF(x) \\ F(x) = W\sigma(\cdot) + \alpha \end{cases} \tag{10}$$

In equation (10), $H(s)$ represents a layer TCN in TGCN, $f(\cdot)$ represents the convolution kernel, and $F(x)$ means the residual function. The loss function during the training process of TGCN model is represented in equation (11).

$$Loss = || X_c - \hat{X} ||_b + \lambda L_2 \tag{11}$$

In equation (11), *Loss* represents the loss function, X_c means the detection value of the model, $\hat{X}$ means the actual values of various detection attributes in the data, L_2 represents the regularization term of the model, and λ represents hyperparameters. The TGCN data accuracy detection method needs to test the core performance indicators before actual operation, and use the test results as a reference to optimize the method specifically and targetedly. The performance of TGCN method is evaluated using root mean square error (RMSE), mean absolute error (MAE), and mean absolute percentage error (MAPE), and the evaluation indicators are shown in equation (12).

$$\begin{cases} P_{RMSE} = \sqrt{\frac{1}{\gamma}\sum (\hat{\mathrm{X}}_{v_i}^{t+1} - \mathrm{X}_{v_i}^{t+1})^2} \\ P_{MAE} = \frac{1}{\gamma}\sum_{i=1}^{\gamma} | \hat{\mathrm{X}}_{v_i}^{t+1} - \mathrm{X}_{v_i}^{t+1} | \\ P_{MAPE} = \frac{1}{\gamma}\sum_{i=1}^{\gamma} \frac{| \hat{\mathrm{X}}_{v_i}^{t+1} - \mathrm{X}_{v_i}^{t+1} |}{\mathrm{X}_{v_i}^{t+1}} \end{cases} \tag{12}$$

In equation (12), $\mathrm{X}_{v_i}^{t+1}$ and $\hat{\mathrm{X}}_{v_i}^{t+1}$ represent the true and reference values of the property v_i of the object at time $(t+1)$, separately, and γ means the number of objects. P_{RMSE}, P_{MAE}, and P_{MAPE} represent RMSE, MAE, and MAPE, respectively. RMSE and MAE can reflect the error situation between the true value and the reference value, while MAPE can reflect the ratio between the error and the true value. In the comprehensive evaluation of algorithms, indicators such as accuracy and recall are often used to assess the rationality of the method. $f1\_score$ is considered a key indicator for measuring the effectiveness of accuracy detection, and its calculation is shown in equation (13).

$$\begin{cases} P = \frac{TP}{TP + FP} \\ R = \frac{TP}{TP + FN} \\ f1\_score = \frac{2 * P * R}{P + R} \end{cases} \tag{13}$$

In equation (13), P represents accuracy, R represents recall, $f1\_score$ represents the combined score of accuracy and recall, and TP means positive samples classified as correct by the model. FN means positive samples classified as incorrect by the model. FP refers to negative samples classified as incorrect by the model. In practical applications, the TGCN designed for research also involves parameter selection. The GCN parameter adjacency matrix usually uses a normalized adjacency matrix, and the number of GCN layers is generally 1-2 to avoid over smoothing. The activation function is often ReLU or LeakyReLU, and the dimension of the weight matrix depends on the dimensions of the input and output features. The learning rate is usually set to 0.001 or 0.0005, which can be optimized using a dynamic learning rate scheduler and L2 regularization to prevent overfitting. The batch size is set to 32, 64, or 128 based on the data size, and an early stop strategy is used during training to prevent overfitting based on performance monitoring of the validation set.

4 Analysis of the effectiveness of dynamic data detection methods in spatiotemporal data mining

4.1 Performance testing of dynamic data detection methods for spatiotemporal data mining

To analyze the ability of the multi-factor development dynamic data detection method established in the study during runtime, data from a network company was used as the test data. Compare the happen before algorithm (HAB) [19], Lockset algorithm (Lock) [20], and BufferTrack algorithm (Butra) [21] with TGCN to evaluate its data processing performance. The software and hardware environment required for the experiment is represented in Table 2.

To verify the effectiveness of SSA missing filling method, a 12-month workload data of a network company was selected as the dataset. The dataset contains the workload changes of the company within one year, with a size of approximately 8GB and six million data points. The data features cover multiple dimensions such as timestamp, request volume, response time, etc., which can help analyze the patterns and trends of network traffic. In the preprocessing step, the study first performed data cleaning, removing some obvious erroneous records and outliers; Then feature selection was carried out, retaining the most critical indicators for workload analysis; Then, the data was standardized to

enable comparison of data from different indicators at the same scale, in order to improve the effectiveness and accuracy of subsequent interpolation algorithms. Fourier interpolation method [22] and SSA method were applied to fill in the missing data. The filling results of the two methods are shown in Figure 6.

Table 2: System development and operating environment.

Project	Software and framework
Integrated development environment	Visual studio 2013
Database environment	SQL Server 2019
Operating system	Windows10, Linux
Framework	.NET, Mini UI
Programming language	C#, JavaScript
Web server	IIS 7.0
Network protocol	UDP, TCP/IP

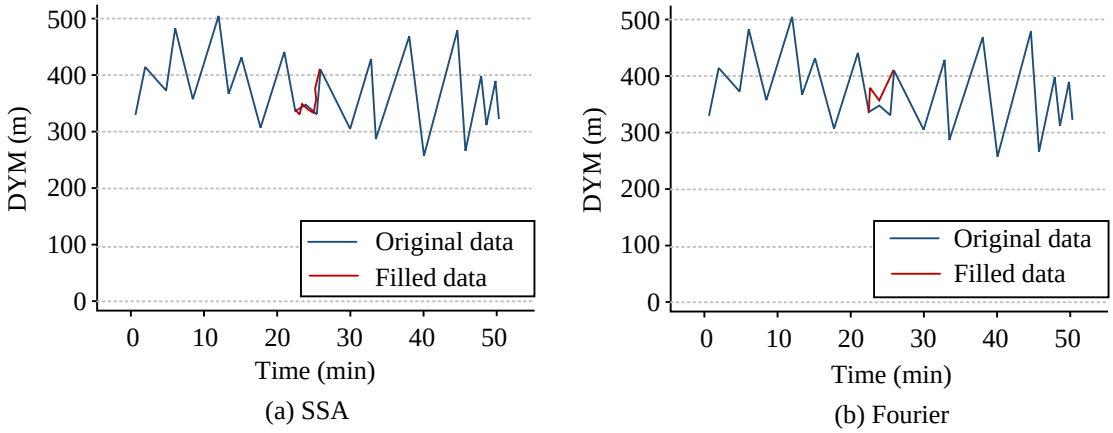


Figure 6: Comparison chart of two filling methods.

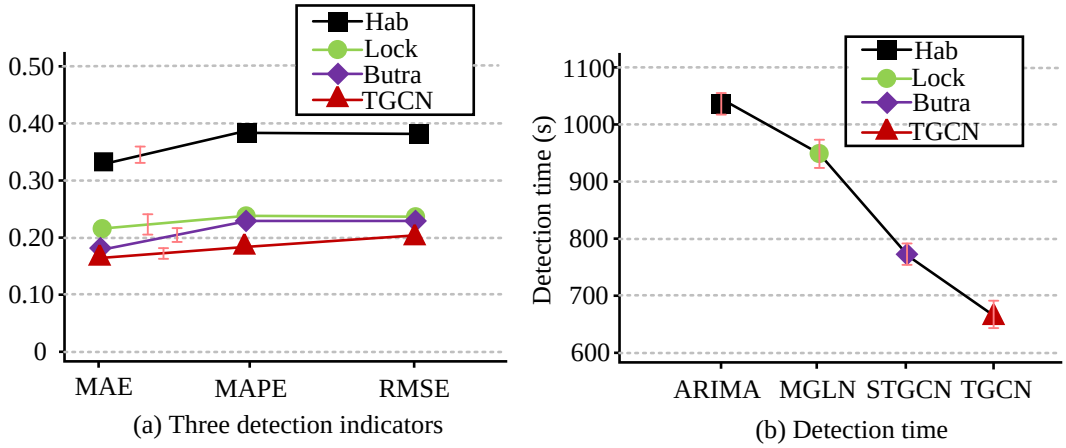


Figure 7: Analysis of performance indicators for different methods of operation.

Figure 6 (a) shows the use of SSA missing filling method to fill in the original data, while Figure 6 (b) shows the use of Fourier fast interpolation method to fill in the original data. As shown in Figure 6 (a), the SSA missing filling method effectively filled in missing data, and the filled data was closely aligned with the original data in the time series. It is worth noting that the DYM deviation of the SSA missing filling method was about 5 meters, indicating minimal deviation from the original signal. The smooth transition between interpolated values without obvious peaks or large fluctuations indicated that this method could accurately preserve the trends and features of the original dataset. From Figure 6 (b), in contrast, the Fourier fast interpolation method showed significant deviation, especially in the time interval of 20 to 30 minutes, where the DYM deviation rose to about 25 meters. This difference highlights that the method failed to accurately capture potential trends during this critical period. There were significant differences between the filled data and the original data, exhibiting unrealistic oscillations and leading to misunderstandings of data trends. The SSA missing filling method is more suitable for scenarios where maintaining consistency in the original data structure is crucial, while the Fourier fast interpolation method may introduce significant inaccuracies, especially when analyzing dynamic data where accurate trend representation becomes crucial.

Considering that the methods in Related Works have been optimized for specific preset scenarios, it cannot be guaranteed that the optimal learning performance can be fully reflected in the studied scenarios. So, the study compared three advanced methods with sufficient applicability, Hab, Lock, and Butra, to analyze the performance of TGCN by comparing Mean Absolute Error (MAE), Root Mean Square Error (RMSE), Mean Absolute Percentage Error (MAPE), and detection time. The Hab algorithm sets a fixed window size of 1024, uses 3 times the standard deviation as the anomaly threshold, and updates statistical features after each window is processed. The Lock algorithm defines a lock set containing 256 key data points, analyzes data at 30 second intervals, and configures specific CPU and memory resource allocation strategies to optimize execution efficiency. The Butra algorithm uses a dynamic buffer with an initial size of 2048, tracking data changes within the last 5 minutes and sampling data at a frequency of once per second to ensure real-time performance and reduce processing latency. TGCN sets 0.003 as the initial learning rate of the algorithm, 0.30 as the activation function parameter, 64 as the batch size, 120 as the number of network iterations, and 2 as the initial dilation factor in the time convolution module. The experimental results are shown in Figure 7.

Figure 7 (a) indicates the behaviour of four methods tested using MAE, MAPE, and RMSE metrics, and Figure 7 (b) indicates the behaviour of the four methods tested using detection time. According to Figure 7 (a), the MAE, MAPE, and RMSE indicators of TGCN were the lowest among the four models, at 0.16, 0.18, and 0.20, respectively, lower than the other three comparison methods. However, the MAE, MAPE, and RMSE indicators of the Hab model were the highest among the four models, at 0.34, 0.39, and 0.38, separately. From Figure 7 (b), Hab had the longest detection time, at 1300 seconds, which was significantly longer than the other three comparison methods, while TGCN had the shortest detection time among the four methods, at only 670 seconds. From this, the TGCN model had the lowest detection indicators among the four models, followed by Butra, indicating that the TGCN model could shorten detection time and improve detection efficiency. Compared with the methods of Hab, Lock, and Butra, the research method had lower computational complexity. Unlike Hab's method, this approach typically involves deep architectures with multiple layers, simplifying feature extraction and focusing on fundamental aspects without unnecessary complexity. The Lock method tends to include redundant processing steps, while the research method uses SSA for denoising and missing data filling, which helps with clearer data processing and improves efficiency. In addition, although Butra's method combines multiple models to capture temporal and spatial features separately, the integrated model of the research method simultaneously solves these two problems and reduces processing time. Finally, the advanced optimization algorithms used in the research methodology allow for faster convergence and significantly reduce training time without sacrificing accuracy. Overall, these factors make research methods more efficient and suitable for real-time dynamic data applications. To further test the stability of TGCN, the Butra model with better performance in the above results was selected as the comparative model, and experiments were conducted under different detection tasks and experimental conditions. The experiment outcomes are represented in Figure 8.

Figure 8 (a) shows the average accuracy changes of TGCN and Butra under different detection task conditions, and Figure 8 (b) shows the average accuracy changes of TGCN and Butra under different experimental conditions. From Figure 8 (a), when TGCN processed different tasks, the average accuracy was relatively stable, maintaining in the range of 0.75-0.80, while Butra's average accuracy fluctuated greatly and was lower than 0.72. According to Figure 8 (b), as the number of experiments increased, the average accuracy of TGCN remained in the range of 0.80-0.85, while Butra's average accuracy fluctuated significantly, below 0.78. From this, TGCN had a high accuracy rate when processing different tasks, and the accuracy rate showed a basically stable trend as the number of experiments increased. In order to determine the effectiveness of different components in the research method, 70% of the data in the dataset was used for ablation experiments, as shown in Table 3.

As can be seen, the Baseline Model demonstrated the best performance with a best accuracy of 97.00%, a recall of 95.00%, and an F1 score of 96.00%, indicating the combined model performed exceptionally well in dynamic data detection tasks. Removing SSA resulted in a decrease in the best accuracy to 94.50%. SSA played a vital role in filling missing data, and its absence led to data incompleteness, negatively impacting the recall rate and F1 score. The removal of GCNN resulted in the most significant performance drop, with the best accuracy plummeting to 92.00%. GCNN was essential for extracting spatial features from the data, and losing this component severely affected the model's ability to handle complex data. The model's performance only slightly declined when TCN was removed, achieving a best accuracy of 93.50%. This suggests that while temporal feature extraction had some impact, it was comparatively less critical than spatial features. With the removal of Fourier Transform, the best accuracy dropped to 95.00%, indicating the importance of Fourier Transform in extracting frequency-domain features. Finally, removing Spatiotemporal Recombination resulted in a performance decline to 93.00%. Although spatiotemporal recombination enhanced the model's ability to capture spatiotemporal data, its impact was relatively smaller than that of other components.

Table 3: Ablation experiment

Component	Best accuracy (%)	Recall (%)	F1 Score (%)
Baseline Model (All)	97.0	95.0	96.0
Remove SSA	94.5	92.0	93.2
Remove GCNN	92.0	90.0	91.0
Remove TCN	93.5	91.5	92.3
Remove Fourier Transform	95.0	93.5	94.2
Remove Spatiotemporal Recombination	93.0	90.5	91.7

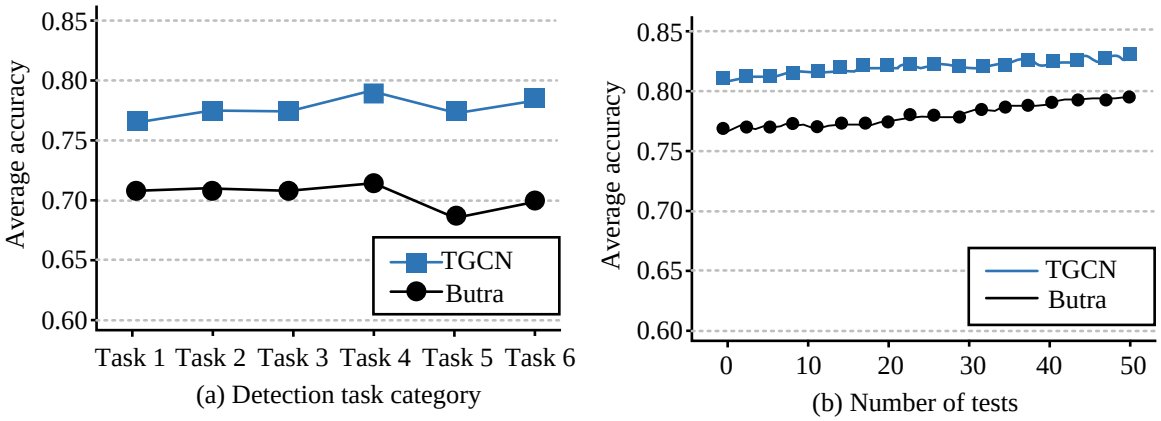


Figure 8: Average accuracy fluctuation analysis.

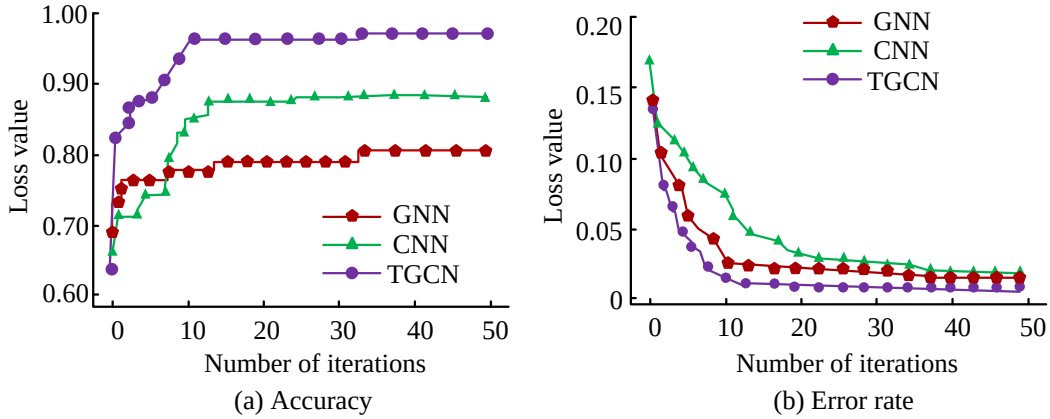


Figure 9: Comparison of accuracy and misjudgment rate of three methods.

4.2 Application analysis of dynamic data detection methods in spatiotemporal data mining

To further demonstrate the advantages of the proposed method in dynamic data monitoring, the accuracy and misjudgment rates of TGCN and CNN [23], GCN [24] were compared. The accuracy of the detection here was obtained during the monitoring process of a large amount of data, so it tended to approach a specific value rather than a special numerical range obtained for a specific individual task. The accuracy and misjudgment rates are represented in Figure 9.

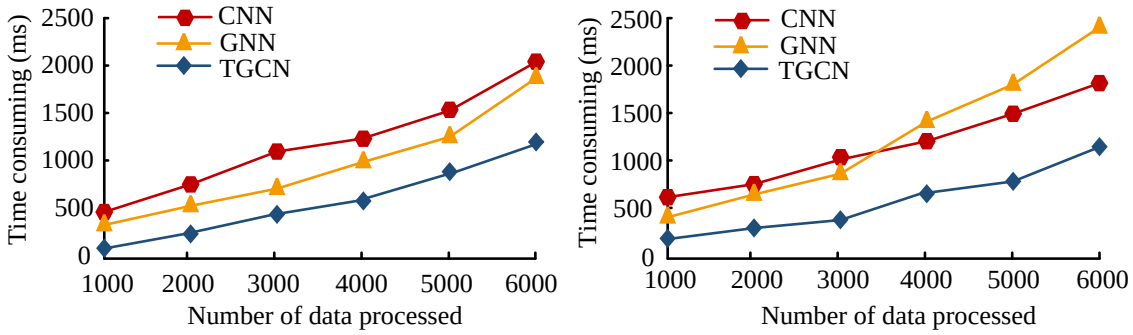
Figure 9 (a) shows the accuracy comparison at different iteration times, and Figure 9 (b) shows the false positive rate comparison at different iteration times. As represented in Figure 9 (a), the accuracy of TGCN was stable at 0.97, the accuracy of CNN was stable at 0.88, and the accuracy of GNN was only 0.81. It is told that the accuracy performance of TGCN was good. According to Figure 9 (b), as the number of iterations increased, the false alarm rates of all three methods decreased. Among them, TGCN decreased from the initial 0.14 to 0.01, which was significantly lower than the other two compared algorithms. TGCN could improve the accuracy of data detection process and reduce false alarm rate, thus achieving dynamic data detection. The attendance data of two departments in a company for 12 months were analyzed, the time under different data volumes was calculated, and the results are represented in Figure 10.

Figure 10 (a) compares the processing time of three methods for different sizes and quantities of data in Department A, and Figure 10 (b) compares the processing time of three methods for different sizes and quantities of data in Department B. From Figure 10 (a), it is told that for Department A, the TGCN method required a processing time of 250 ms when processing

2000 data points, and 1000 ms when processing 6000 data points. For the same amount of data, the processing time of TGCN was the shortest, and as the amount of data increased, the required processing time also increased. According to Figure 10 (b), for Department B, the TGCN method required a processing time of 300 ms for 3000 data points and 750 ms for 5000 data points, which was lower than the other two comparison algorithms. For the same amount of data, TGCN had the shortest processing time, and as the amount of data increased, the required processing time also increased. Comparing the data processing volume before and after applying TGCN at different times, the application results in two departments are represented in Figure 11.

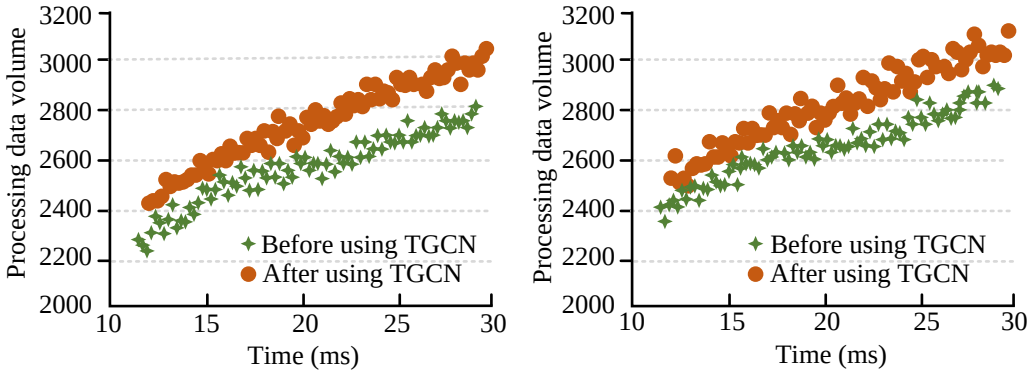
Figure 11 (a) tells the amount of data processed by department A before and after applying TGCN at different times, while Figure 11 (b) tells the amount of data processed by department B before and after applying TGCN at different times. According to Figure 11 (a), for Department A, before and after using the TGCN method, the data processing increased from around 2500 to around 2600 within 15ms, and from around 2800 to 3000 within 30ms. According to Figure 11 (b), for Department B, before and after using the TGCN method, the data processing increased from around 2500 to around 2700 within 15ms, and from around 2900 to 3100 within 30ms. Using the TGCN method within the same time frame can accelerate data processing speed and improve efficiency. In order to further analyze the advantages and scalability of the research method, an online social networking platform was selected for real-time data detection, and the advanced K-nearest neighbor interpolation method [25] and polynomial interpolation method [26] in recent years were introduced for comparison, as shown in Table 4.

As shown in Table 4, the RMSE of TGCN method was 5.0 meters, significantly lower than K-nearest neighbor interpolation method (12.0 meters) and polynomial interpolation method (15.0 meters), indicating that TGCN method had significant advantages in filling accuracy. The relative error of TGCN method was only 1.2%, which was the lowest among all comparison methods, highlighting its superiority in data filling. The detection time of TGCN was only 1.1 seconds, which was lower compared to other methods. The cosine similarity of TGCN method was 0.95, indicating a high degree of consistency between the filled data and the original data. In contrast, the K-nearest neighbor interpolation method had a similarity of 0.80 and the polynomial interpolation method had a similarity of 0.75, indicating that its similarity was not as good as the TGCN method. After comparison, TGCN had the shortest detection time and the best detection accuracy, and its good performance in different data scenarios also proved the good scalability of the research method.



(a) A department (b) B department

Figure 10: Calculation time for processing different data.



(a) A department (b) B department

Figure 11: Processing data volume at different times.

Table 4: Comparative analysis of advanced methods

Method	TGCN	K-nearest neighbor interpolation method	Polynomial interpolation
RMSE (m)	5	12	15
MAPE (%)	1.2	3	4.5
RE (%)	1.2	2.5	3.5
MAE (m)	10	20	25
Detection time (seconds)	1.1	1.7	1.2
Cosine similarity	0.95	0.8	0.75
Data consistency (%)	98	92	90
Interpolation smoothness (m/min)	0.3	0.8	1

4.3 Discussion

The study proposes a method based on TGCN, combining GCNN and TCN, to achieve accuracy detection of dynamic data. Compared with traditional methods in related work, TGCN exhibits significant advantages in efficiency, accuracy, and robustness. Firstly, in terms of efficiency, traditional methods such as autoregressive models and moving average models often rely on linear regression and simple statistical methods for data processing, resulting in slower processing speeds. Relatively speaking, TGCN adopts deep learning technology and can process massive amounts of data in parallel. Experimental results showed that this method only took 670 seconds in detection time, which often took several hours to achieve in traditional models. This significant time advantage makes TGCN a more attractive choice in real-time data monitoring applications. Secondly, in terms of accuracy, compared with threshold-based anomaly detection methods, TGCN can simultaneously consider the temporal and spatial characteristics of data by introducing time-series analysis. Many methods in related work often have an accuracy of only around 0.85 when dealing with outliers, which cannot effectively handle complex data streams. The TGCN in this study improved the accuracy of detection by combining singular spectrum analysis, and the experimental results showed that its accuracy remained stable above 0.97. This optimization enables TGCN to maintain efficient anomaly detection capabilities even in the face of dynamically changing data. In terms of robustness, some existing methods are often sensitive to noise and data loss, leading to fluctuations in detection results. TGCN, through its deep network structure, has strong adaptive capabilities and exhibits better adaptability to interference in dynamic data. In the experiment, TGCN showed improved robustness when dealing with noisy data, resulting in significantly higher accuracy and stability of the model in complex environments compared to many related works. Although the TGCN method in this study achieved excellent performance in multiple aspects, its limitations cannot be ignored. The training cycle of the model was relatively long, especially in real-time processing of large-scale datasets, which may face a bottleneck in computing resources. In addition, TGCN had poor interpretability in practical applications, which may make it difficult for business personnel to understand the decision-making logic of the model. Future research can explore the integration of interpretable online artificial intelligence technology into the TGCN model, thereby enhancing its interpretability and user trust. In addition, in order to support real-time data detection tasks on large-scale datasets, it is necessary to develop a distributed computing framework to further enhance the scalability of the model.

5 Conclusion

A dynamic data detection technique based on spatiotemporal mining technology was developed to enhance data processing in the network. During the process, the singular spectrum analysis method was introduced to fill in missing data, and the spatiotemporal topology structure was fused to establish a dynamic data detection method. A data accuracy detection method was proposed by combining GCNN and TCN to complete the data accuracy detection. The data was detected in both the temporal and spatial dimensions, and the two were added together to obtain complete detection data. Finally, the validity of the raised method was analyzed. The experiment outcomes indicated that in terms of data filling, the SSA missing filling method used in the study was more in line with the original data curve for filling missing data. In terms of false positive rate, the method proposed by the research decreased from 0.14 to 0.01, which was lower than the two compared methods. As the number of iterations increased, the false positive rate gradually decreased. In terms of processing speed, before and after using the TGCN method, the data processing time increased from around 2500 to 2700 within 15 ms, and from around 2900 to 3100 within 30 ms. The research method had better data filling effect on missing data, which could process data at a higher speed and ensure stable accuracy at a higher level.

6 Fundings

The research is supported by National Social Science Foundation of China in 2022: Research on Evaluation System and Guarantee Mechanism of Labor Rights and Interests of Flexible Employees in Platform Enterprises (22XJY004).

References

[1] Zhenpeng Zhang. SD-WSN network security detection methods for online network education. Informatica, 48(21):51-66, 2024. https://doi.org/10.31449/inf.v48i21.6257

[2] Praveen Kumar Tyagi, and Dheeraj Agarwal. Systematic review of automated sleep apnea detection based on physiological signal data using deep learning algorithm: a meta-analysis approach. Biomedical Engineering Letters, 13(3):293-312, 2023. https://doi.org/10.1007/s13534-023-00297-5

[3] Chunhua Liang. Application of maximum entropy fuzzy clustering algorithm with soft computing in migration anomaly detection. Informatica, 48(17):171-182, 2024. https://doi.org/10.31449/inf.v48i17.6537

[4] Daniele Dalla Torre, Andrea Lombardi, Andrea Menapace, Ariele Zanfei, and Maurizio Righetti. Exploring the feasibility of support vector machine for short-term hydrological forecasting in south tyrol: challenges and prospects. Discover Applied Sciences, 6(4):1-19, 2024. https://doi.org/10.1007/s42452-024-05819-z

[5] Luke Lewis-Borrell, Jessica Irving, Chris J. Lilley, Marie Courbariaux, Gregory Nuel, Leon Danon, Kathleen M. O'reilly, Jasmine M.S. Grimsley, Matthew J. Wade, and Stefan Siegert. Robust smoothing of left-censored time series data with a dynamic linear model to infer SARS-CoV-2 RNA concentrations in wastewater. AIMS Mathematics, 8(7):16790-16824, 2023. https://doi.org/10.3934/math.2023859

[6] Francisco de Arriba-Pérez, Silvia García-Méndez, Fátima Leal, Benedita Malheiro, and Juan C. Burguillo. Online detection and infographic explanation of spam reviews with data drift adaptation. Informatica, 35(3):1-25, 2024. https://doi.org/10.15388/24-INFOR562

[7] Yaping Wang, Zunshan Xu, Songtao Zhao, Jiajun Zhao, and Yuqi Fan. Performance degradation prediction of rolling bearing based on temporal graph convolutional neural network. Journal of Mechanical Science and Technology, 38(8):4019-4036, 2024. https://doi.org/10.1007/s12206-024-0702-z

[8] Chunyong Yin, Sun Zhang, Jin Wang, and Neal N. Xiong. Anomaly detection based on convolutional recurrent autoencoder for IoT time series. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 52(1):112-122, 2022. https://doi.org/10.1109/TSMC.2020.2968516

[9] Jiwei Huang, Jiangyuan Wan, Bofeng Lv, Qiang Ye, and Ying Chen. Joint computation offloading and resource allocation for edge-cloud collaboration in internet of vehicles via deep reinforcement learning. IEEE Systems Journal, 17(2):2500-2511, 2023. https://doi.org/10.1109/JSYST.2023.3249217

[10] Stefan Bloemheuvel, Jurgen van den Hoogen, Dario Jozinović, Alberto Michelini, and Martin Atzmueller. Graph neural networks for multivariate time series regression with application to seismic data. International Journal of Data Science and Analytics, 16(3):317-332, 2023. https://doi.org/10.1007/s41060-022-00349-6

[11] Hao Xu, Zihan Sun, Yuan Cao, and Hazrat Bilal. A data-driven approach for intrusion and anomaly detection using automated machine learning for the Internet of Things. Soft Computing, 27(19):14469-14481, 2023. https://doi.org/10.1007/s00500-023-09037-4

[12] Tianzhe Jiao, Xiaoyue Feng, Chaopeng Guo, Dongqi Wang, and Jie Song. Multi-agent deep reinforcement learning for efficient computation offloading in mobile edge computing. Computers, Materials, and Continues, 76(9):3585-3603, 2023. https://doi.org/10.32604/cmc.2023.040068

[13] Erasmo Purificato, Ludovico Boratto, and Ernesto William De Luca. Toward a responsible fairness analysis: from binary to multiclass and multigroup assessment in graph neural network-based user modeling tasks. Minds and Machines, 34(3):1-34, 2024. https://doi.org/10.1007/s11023-024-09685-x

[14] Jun Hu, Xinyu Yang, Liang Yan, and Qinghua Zhang. Pedestrian trajectory prediction based on spatiotemporal attention mechanism. International Journal of Machine Learning and Cybernetics, 15(8):3299-3312, 2024. https://doi.org/10.1007/s13042-023-02093-0

[15] Yamin Fang, and Hui Liu. A spatiotemporal dissolved oxygen prediction model based on graph attention networks suitable for missing data. Environmental Science and Pollution Research, 30(34):82818-82833, 2023. https://doi.org/10.1007/s11356-023-28030-w

[16] Jesús Pineda, Benjamin Midtvedt, Harshith Bachimanchi, Sergio Noé, Daniel Midtvedt, Giovanni Volpe, and Carlo Manzo. Geometric deep learning reveals the spatiotemporal features of microscopic motion. Nature Machine Intelligence, 5(1):71-82, 2023. https://doi.org/10.1038/s42256-022-00595-0

[17] Wenhao Li, Yanyan Chen, Yuyan Pan, and Yunchao Zhang. An improved spatiotemporal network traffic flow prediction method based on impedance matrix. Journal of Highway and Transportation Research and Development, 18(2):67-75, 2024. https://doi.org/10.26599/HTRD.2024.9480015

[18] Fengxin Chen, Ye Yu, Liangliang Ni, Zhenya Zhang, and Qiang Lu. DSTVis: toward better interactive visual analysis of Drones' spatiotemporal data. Journal of Visualization, 27(4):623-638, 2024. https://doi.org/10.1007/s12650-024-00982-2

[19] Yan Jian, Xiaoyang Dong, and Liang Jian. Detection and recognition of abnormal data caused by network intrusion using deep learning. Informatica, 45(3):441-445 2021. https://doi.org/10.31449/inf.v45i3.3639

[20] Erchao Li, and Kuankuan Qi. Ant colony algorithm for path planning based on grid feature point extraction. Journal of shanghai jiao tong university: English Edition, 28(1):86-99, 2023. https://doi.org/10.1007/s12204-023-2572-4

[21] Si-Xiao Gao, Hui Liu, and Jun Ota. Energy-efficient buffer and service rate allocation in manufacturing systems using hybrid machine learning and evolutionary algorithms. Advances in

Manufacturing, 12(2):227-251, 2024. https://doi.org/10.1007/s40436-023-00461-1

[22] Andriy Bondarenko, Danylo Radchenko, and Kristian Seip. Fourier interpolation with zeros of zeta and L-functions. Constructive Approximation, 57(2):405-461, 2022. https://doi.org/10.1007/s00365-022-09599-w

[23] Kavita Bhosle, and Vijaya Musande. Evaluation of deep learning CNN model for recognition of devanagari digit. Artificial Intelligence and Applications, 1(2):114-118, 2023. https://doi.org/10.47852/bonviewAIA3202441

[24] Jiawei Zhu, Xing Han, Hanhan Deng, Chao Tao, Ling Zhao, Pu Wang, Tao Lin, and Haifeng Li. KST-GCN: A knowledge-driven spatial-temporal graph convolutional network for traffic forecasting. IEEE Transactions on Intelligent Transportation Systems, 23(9):15055-15065, 2022. https://doi.org/10.1109/TITS.2021.3136287

[25] Dongdong Cheng, Jinlong Huang, Sulan Zhang, and Quanwang Wu. A robust method based on locality sensitive hashing for K-nearest neighbors searching. Wireless Networks, 30(5):4195-4208, 2024. https://doi.org/10.1007/s11276-022-02927-9

[26] M. Akif Günen. Comparison of histogram-curve fitting-based and global threshold methods for cloud detection. International Journal of Environmental Science and Technology, 21(6):5823-5848, 2024. https://doi.org/10.1007/s13762-023-05379-6