Intelligent Analysis Method of Multidimensional Time Series Data Based on Deep Learning

Zhongwei Chen and Xiaofeng Li[(✉)]

College of Information Engineering, Guangxi University of Foreign Languages,
Nanning 530222, China
`top00112233@163.com`

Abstract. In order to improve the intelligent analysis performance of multidimensional time series data, an intelligent analysis method of multidimensional time series data based on deep learning is proposed. The deep learning network model is used to establish the distribution function of correctly classified and wrongly classified data. According to the calculation process of the deep learning network, the outliers of multi-dimensional time series data are mined. The feature space of multi-dimensional time series data is determined using the k-nearest neighbor method. The multi-dimensional time series data is reduced in dimension by calculating the covariance matrix of multi-dimensional time series data, and the time series data is classified according to the preference level, extract the characteristics of multidimensional time series data, and combine with the design of multidimensional time series data analysis algorithm to achieve intelligent analysis of multidimensional time series data. The experimental results show that the method in this paper can reduce the mean absolute error and root mean square error to within 0.25 and 0.3 respectively in the intelligent analysis of multidimensional time series data, and improve the analysis performance of multidimensional time series data.

Keywords: Deep Learning · Multi Dimensional Time Series Data · Intelligent Analysis · Dimension Reduction Processing

1 Introduction

Time series data refers to time series data, which is a data column recorded by the same indicator in time sequence [1]. The analysis of time series data mainly predicts unknown time series data by analyzing the existing time series data and constructing a time series model. The difficulty of time series data analysis lies in how to grasp the characteristics of historical time series data, and then more effectively predict future time series data.

With the extensive application of IoT technology in smart cities, Industry 4.0, supply chains and home automation, it has generated massive data. According to a research report of Cisco, by 2021, the number of devices connected to the Internet of Things will reach 12 billion, which means that the data traffic generated each month will exceed 49 exabytes. Whether in the industrial field or in the life field, the use of sensors is

everywhere. A large amount of data and information generated by these sensors has gradually become the most common data form in the application of the Internet of Things [2]. Therefore, in the application of the Internet of Things, the analysis and processing of these data has become increasingly important. The whole IoT architecture consists of data perception layer, data transmission layer, data processing and storage layer, joint learning and analysis and IoT application layer. Through the analysis and processing of the data collected by these sensors, more valuable and meaningful information can be obtained, thus providing guarantee for more intelligent decision-making, deployment and supervision of these IoT applications.

With the rapid development of Internet of Things technology and artificial intelligence technology, people's life and work are becoming more and more intelligent, and more and more sensors appear in daily work and life. Due to the high requirements of real-time performance in the Internet of Things society, it is necessary to ensure The normal operation of the IoT system can reduce the loss caused by failure. For this reason, it is necessary to make more accurate predictions of the time series data generated in the IoT, and to detect the abnormal data that may be generated during the operation of the IoT system to reduce the risk of failure. The loss caused by abnormal data.

In the domestic research, Tao Tao et al. [3] proposed an anomaly detection method based on deep learning to identify abnormal refueling vehicles. First, the automatic encoder was used to extract the features of the relevant data collected by the refueling station, and then the Seq2Seq model embedded in two-way short-term and short-term memory was used to predict refueling behavior. Finally, the threshold of the abnormal point was defined by comparing the predicted value and the original value. Xia Ying et al. [4] conducted anomaly detection through data analysis, which helps to accurately identify abnormal behaviors, thus improving service quality and decision-making ability. However, due to the spatiotemporal dependence of multidimensional time series data and the randomness of abnormal events, existing methods still have certain limitations. Aiming at the above problems, this paper proposes a multi-dimensional time series data anomaly detection method MBCLE which integrates a new statistical method and bidirectional convolution LSTM. The method introduces stacked median filtering to deal with point anomalies in the input data and smooth data fluctuations; design a predictor combining bidirectional convolutional long short-term memory network and bidirectional long short-term memory network for data modeling and prediction; bidirectional circular index Weighted moving average smoothes prediction error; uses dynamic thresholding method to compute thresholds to detect contextual anomalies. However, the above two methods have the problem of large average absolute error in the intelligent analysis of multidimensional time series data.

In foreign research, Meng C et al. [5] proposed a multi-dimensional time series outlier detection framework based on time convolution network automatic encoder, which can detect outliers in time series data, such as identifying equipment failures, dangerous driving behaviors of vehicles, etc. A feature extraction method is used to transform the original time series into a time series with rich features. The proposed TCN-AE is used to reconstruct time series data with rich characteristics, and the reconstruction error is used to calculate outliers. Although this method can complete the detection and analysis

of outliers, the root-mean-square error of the analysis results of this method is large, and the performance of this method has some room for improvement.

In order to solve the problem of large average absolute error and root mean square error in the traditional methods mentioned above, an intelligent analysis method of multidimensional time series data based on deep learning is proposed.

2   Intelligent Analysis of Multi-dimensional Time Series Data

2.1   Mining Multidimensional Time Series Data Outliers

Due to the small scale and wide range of time series data, it is very easy to regard abnormal values as wrong or invalid data during analysis, which will also affect the overall accuracy of multidimensional time series data, cause misunderstanding and increase the difficulty of analysis. Therefore, we use deep learning network technology to mine outliers of time series data. The deep learning network model is shown in Fig. 1.
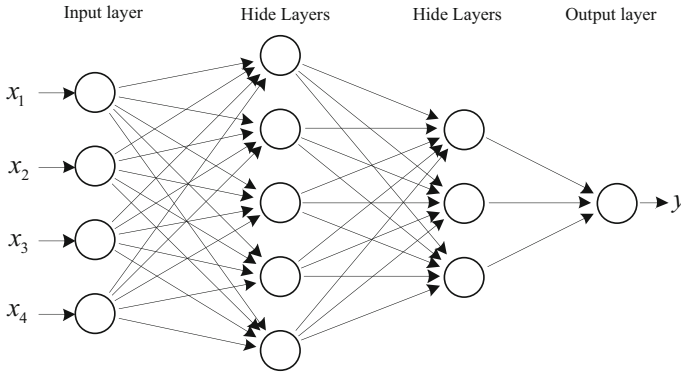

Fig. 1. Deep learning network model

From the above figure, it can be found that the composition of the deep learning network model mainly includes the input layer, the hidden layer and the output layer, corresponding to the input value, output value, weight and output function. The basic relationship between the different components is shown in the following formula:

$$y^* = g(\xi x^* + \phi) \tag{1}$$

In the formula, y^* represents the output value, g represents the transfer function, ϕ represents the offset, ξ represents the weight, and x^* represents the input value.

The steps to mine multidimensional time series data outliers using deep learning networks are as follows:

Step 1: Initialize the deep learning network, randomly initialize the weights and biases of each layer, and the number of neurons in the input layer is determined by the number of data attributes in the dataset. The neighborhood range of the detection object has been obtained through the above process. Assuming that there are M attributes in

the data set within the neighborhood range, the number of neurons in the input layer is set to M;

Step 2: Obtain the input and output vectors through the given training set, and set them as vectors x^* and y^* respectively;

Step 3: Specify the number of nodes, hide and output the number of nodes;

Step 4: Obtain the actual output value of the deep learning network according to the given data forwarding output data;

Step 5: Process the output value, which can fully reflect the distribution of the data set. According to the output value of the deep learning network, the multi-dimensional time series data can be judged by the entropy value. The entropy value represents the multi-dimensional time series data in a certain category. Uncertainty [6]. The greater the entropy, the higher the uncertainty of multi-dimensional time series data, and the more likely abnormality of multi-dimensional time series data. It is proposed that when the entropy value exceeds a certain threshold, the multidimensional time series data is an outlier. When the threshold is small, set a threshold value of f, whose range is between 0 and 1.

Therefore, the evaluation function f is given, which is related to the number of multidimensional time series data of two categories, namely:

$$f = \gamma \left(\alpha P_{true}, \beta P_{false} \right) \tag{2}$$

In the formula, α and β represent the weights of multi-dimensional time series data respectively, P_{true} represents the multi-dimensional time series data classified as correct, P_{false} represents the multi-dimensional time series data classified as wrong, and γ represents the validity of using a certain threshold to judge abnormal points.

The value of γ is closely related to the mining effect of outliers. The larger the value, the better the mining effect. On the contrary, the worse the mining effect.

The value of γ is inversely proportional to the number of multidimensional time series data with correct classification, that is, the number of multidimensional time series data with wrong classification is proportional to the value of γ. Therefore, the formula of evaluation function f is as follows:

$$f = -\alpha P_{true} + \beta P_{false} \tag{3}$$

In order to improve the mining accuracy, the following formula is used to reduce the f error, and its expression is:

$$\varepsilon = -\delta \frac{\partial f}{\partial \varpi_{jk}} \tag{4}$$

In the formula, δ represents the deep learning coefficient, which is the learning speed in the process of deep learning network training, namely the learning rate.

Assume that $f_a(x)$ represents the multidimensional time series data density function with correct classification in the dataset, and $f_b(x)$ represents the multidimensional time series data density function with incorrect classification in the dataset, as shown in Fig. 2.

According to the distribution function in Fig. 2, we get:
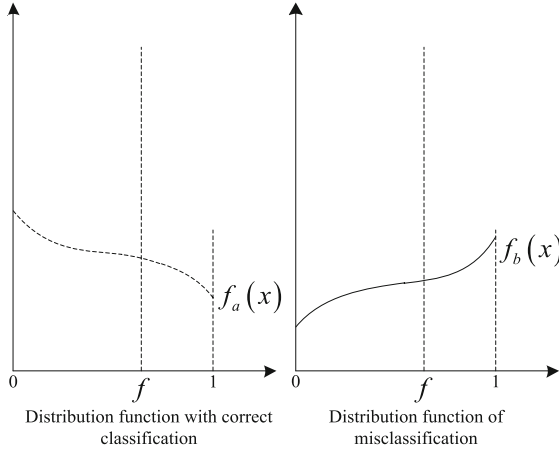
$$P_a = \int_f^1 f_a(x)dx \tag{5}$$

Fig. 2. Distribution function of correctly classified and incorrectly classified data

$$P_b = \int_f^1 f_b(x)dx \qquad (6)$$

According to the distribution function of formula (5) and formula (6), we can get:

$$P(f) = -\alpha P_a + \beta P_b \qquad (7)$$

Judging the efficiency of outlier mining of multi-dimensional time series data through $P(f)$, the value of this value is proportional to the mining efficiency. The larger the value is, the higher the mining efficiency is, and the worse the mining effect is. Therefore, when $P(f)$ takes a maximum value, the value of entropy is the best.

Step 6: According to the actual output and expected output of the deep learning network, calculate the output error of the deep learning network and judge the stop condition of the deep learning network. If yes, stop training and exit the deep learning network to evaluate the outliers of multidimensional time series data; if not, return to Step 2.

Step 7: Evaluation of multidimensional time series data outliers, evaluate the detected outliers, and find out the reasons for the multidimensional time series data outliers. After the outliers are identified and verified, the outliers need to be post-processed in order to accurately serve educational decision-making [7]. First, the causes of outliers are analyzed from a technical point of view; if there are technical reasons or human input errors, such abnormal data needs to be eliminated to reduce the difficulty of post-processing and improve the accuracy of multi-dimensional time series data. Second, the influence of subjective assumptions eliminates technical error factors, and appropriate intelligent mining algorithms are used to mine abnormal points, establish an analysis model, and determine an appropriate abnormal range to reduce the subjectivity of abnormal points and reduce the correlation of abnormal points. error effects. Third, the analysis results of abnormal phenomena are presented in an intuitive form, so that the causes of abnormal phenomena can be analyzed in detail in combination with the specific education and

teaching situation, and corresponding measures and plans can be put forward in a targeted manner to make multi-dimensional time series data outliers. The point detection algorithm plays a greater practical value.

The above calculation process based on deep learning network is iterated continuously until all multidimensional time series data outliers are mined, and then the algorithm designed this time is stopped, so as to complete the outlier mining of multidimensional time series data based on deep learning technology through the above process.

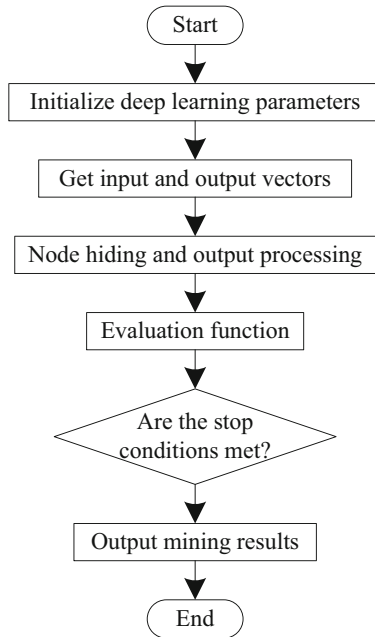The specific process of outlier mining in multidimensional time series data is shown in Fig. 3.

Fig. 3. Outlier mining process of multidimensional time series data

2.2 Dimensionality Reduction Processing Multidimensional Time Series Data

First, the k-nearest neighbor method is used to determine the domain of sampling points in the z-dimensional time series, and then the nearest neighbor graph G is constructed. The nodes in Figure G correspond to the points in $\{x_t\}$, and the edges in Figure G represent the nearest neighbor relationship between points. Calculate the approximate geodesic distance L_{ij}, select two nodes x_i and x_j in Figure G, and measure the approximate geodesic distance of the shortest path between them to obtain the matrix $L^2 = \left[L_{ij}^2 \right] \in \gamma^{N \times N}$.

The process of constructing the nuclear matrix $\tilde{H}$ is shown in formula (8):

$$\tilde{H} = H\left(L^2\right) + 2\zeta H(L) + \frac{1}{2}\zeta^2 \tag{8}$$

In the formula, $\zeta \geq \zeta^*$, so as to ensure that the matrix $\tilde{H}$ is positive semi-definite. After calculating the r eigenvalues of $\tilde{H}$, the eigenvalue matrix $\partial \in \gamma^{r \times x_r}$ can be obtained, and the corresponding eigenvector is $q \in \gamma^{r \times r}$. Finally, the embedded coordinates of the n sampled data points in the r-dimensional space can be obtained as:

$$\hat{X} = (\hat{x}_1, \hat{x}_2, \cdots, \hat{x}_n)^T = \partial^{\frac{1}{2}} q^T \tag{9}$$

According to the calculation, $\tilde{H}$ is the kernel matrix, so the (i, j) element of $\tilde{H}$ can be expressed as:

$$\tilde{H}_{ij} = k(x_i, x_j) = \Phi^T(x_i)\Phi(x_j) \tag{10}$$

Then in the low-dimensional feature space, the formula for calculating the covariance matrix of multi-dimensional time series data is:

$$J = \frac{1}{N}(\varphi D)^T \tag{11}$$

where, $\Omega = [\Phi(x_1), \Phi(x_2), \Phi(x_n)]$, the node coordinates in the low dimensional feature space, can be obtained by mapping the central matrix to the eigenvector of matrix Ω. Select a new test sample $x_l \in \gamma^b$ whose coordinate in the low dimensional feature space is $\hat{x}_l \in \gamma^a$, then:

$$\left[\hat{x}_l\right]_j = \sum_{i=1}^{n} q_{ij} k(x_i, x_l) \tag{12}$$

Among them, q_{ij} is the i element of the feature vector q_j, and $\left[\hat{x}_l\right]_j$ represents the j element of x_l, $k(.)$ represents the coordinate transformation function.

According to the above process, the k-nearest neighbor method is used to reduce the dimension of multi-dimensional time series data.

2.3   Extracting Multi-dimensional Time Series Data Features

The multi-objective decision-making theory in operational research is a challenging and active branch field. Based on the decision-making background, multi-objective decision-making considers several evaluation indicators that may conflict and diverge with each other, and combines optimization theory, statistics theory, management philosophy and operational research methods to optimize and rank multiple alternatives [8]. The proposed method extracts the characteristics of time series data on the basis of multi-objective decision theory, and the specific steps are as follows:

The standard decision matrix C is constructed according to the extracted interval extreme point sequence. The rows and columns in the decision matrix C are the extreme

points existing in the time series data and the object attributes corresponding to the extreme points, that is, the multi-objective criterion. Let the vector $A = (a_1, \cdots, a_n)$ be a set composed of n extreme points, and the vector $B = (b_1, \cdots, b_m)$ be a set composed of m extreme point attributes, which is the evaluation index. After standardizing the decision matrix, the decision objects corresponding to different indicators are compared by the following formula:

$$d_k(a_i, a_j) = c_k(a_i) - c_k(a_j) \tag{13}$$

In the formula, $d_k(a_i, a_j)$ represents the difference between extreme points a_i and a_j on evaluation index c_k.

The difference $d_k(a_i, a_j)$ is replaced by the standardized preference $\vartheta_k(a_i, a_j)$ through the preference function, namely:

$$\vartheta_k(a_i, a_j) = \psi_k(d_k(a_i, a_j)) \tag{14}$$

In the formula, ψ_k represents the preference function. The time series data feature extraction algorithm based on multi-objective decision-making selects the preference function on the basis of linear features:

$$\psi_k(x) = \begin{cases} 0, & \text{if } x < v_k \\ \frac{x - v_k}{u_k - v_k}, & \text{if } v_k < x < u_k \\ 1, & \text{if } x > u_k \end{cases} \tag{15}$$

where, u_k represents the preference threshold and v_k represents the indifference threshold. The above two thresholds can predict the distribution of preference.

The preference matrix W is constructed according to the calculated preference degree corresponding to the extreme point on the rating index. The time series data feature extraction algorithm based on multi-objective decision-making introduces the weight index [9] to measure the weight relationship between targets. The preference matrix W is weighted and normalized by the following formula:

$$W = \sum_{k=1}^{v} f_k(x) \cdot \delta_k \tag{16}$$

In the formula, δ_k represents the relative weight corresponding to the evaluation index c_k. Usually the relative weights are in line with $\delta_k > 0$ and $\sum_{k=1}^{W} \delta_k = 1$.

Through the above analysis, it can be seen that the multi-objective preference degree existing between decision object a and decision object b conforms to the following formula:

$$\begin{cases} \vartheta(a_i, a_j) > 0 \\ \vartheta(a_i, a_j) + \vartheta(a_j, a_i) < 1 \end{cases} \tag{17}$$

Let Λ^+ represent the positive preference flow. On all decision objectives, the preference level corresponding to the positive preference flow Λ^+ of decision object a_i is the

highest, and Λ^- represents the negative preference flow. On all decision objectives, the preference level corresponding to the negative preference flow Λ^- of decision object a_i is the lowest. The final decision results are obtained through positive preference flow Λ^+ and negative preference flow Λ^-, and the preferences are sorted. The calculation formulas of positive preference flow Λ^+ and negative preference flow Λ^- are as follows:

$$\begin{cases} \Lambda^+(a) = \frac{1}{n-1} \sum_{x \in A} \vartheta(a_i, a_j) \\ \Lambda^-(a) = \frac{1}{n-1} \sum_{x \in A} \vartheta(a_j, a_i) \end{cases} \tag{18}$$

In extreme cases, the corresponding negative preference flow value and positive preference flow value of the optimal decision object are 0 and 1 respectively; if the corresponding negative preference flow value and positive preference flow value are not 0 and 1, it is the worst decision object. Time series data features based on multi-objective decision-making Other algorithms realize time series data feature extraction according to the results of preference flow sorting. Obtain the net preference flow from the negative preference flow and the positive preference flow:

$$\Lambda(a) = \Lambda^+(a) - \Lambda^-(a) \tag{19}$$

The net preference flow generally meets the following conditions:

$$\begin{cases} \Lambda(a_i) \in [-1, 1] \\ \sum_{a_i \in A} \Lambda(a_i) = 0 \end{cases} \tag{20}$$

The extreme value points are sorted according to the calculated net preference flow $\Lambda(a_i)$. The higher the value of the net preference flow, the higher the preference level corresponding to the extreme value points. The time series data are classified according to the level to achieve feature extraction of multi-dimensional time series data.

2.4   Designing Multi-dimensional Time Series Data Analysis Algorithms

On the basis of completing the feature extraction of multi-dimensional time series data, grid is introduced as index calculation to grid the activity space of multi-dimensional time series data in the network, that is, the division method of longitude and latitude is used to discretize the activity space of network multi-dimensional time series data into For several grids of size $\varpi \times z$, after the grid division is completed, the active position points of the network multi-dimensional time series data are mapped to the corresponding cells one by one. After the grid division, the length reference value $\Delta\varpi$ and width reference value Δz of each cell can be Calculated by the following formula:

$$\Delta\varpi = \frac{\max|p_i \times x - p_j \times y|}{\varpi} \tag{21}$$

$$\Delta z = \frac{\max|p_i \times x - p_j \times y|}{z} \tag{22}$$

Among them, $\forall p_i, p_j \in \Sigma, i \neq j$ and Σ represent the movement trajectory of the multi-dimensional time series data of the network, p_i and p_j represent the coordinate points of the multi-dimensional time series data in the activity space, and $(p_i \times x, p_i \times y)$ and $(p_j \times x, p_j \times y)$ represent the coordinate values of p_i and p_j in the latitude and longitude directions, respectively.

After completing the grid division of the network activity space, the binary normal density kernel function is used to calculate the density estimation value of any cell 5 [10]. The specific calculation formula is as follows:

$$d(\chi) = \frac{1}{n\lambda^2} \sum_{i=1}^{n} \frac{1}{2\pi} \exp\left(-\frac{|\chi - p_i|^2}{2\lambda^2}\right) \tag{23}$$

$$\lambda = \frac{1}{2} n^{-\frac{1}{6}} \left(\varepsilon_x^2 + \varepsilon_y^2\right)^{\frac{1}{2}} \tag{24}$$

where, λ represents the moving track smooth parameter of dynamic time series data, and n represents the total number of dynamic time series data contained in the network; $|\chi - p_i|$ represents the distance between cells χ and p_i, and the standard deviation of x coordinate and y coordinate corresponding to all position points in Table Σ of ε_x and ε_y.

According to the above calculation, the abnormal area of the network can be regarded as the area connected by adjacent cells with the same or similar density values that meet certain conditions. The judgment formula for the abnormal area is as follows:

$$|d(\chi_1) - d(\chi_2)| > \sigma \tag{25}$$

Among them, $d(\chi_1)$ and $d(\chi_2)$ represent the density estimates of two adjacent cells in the network, and σ represents the abnormal area judgment threshold.

After obtaining a series of abnormal areas of the network according to the above steps, it is necessary to further obtain the time-varying law of the activity position of multi-dimensional time series data in an abnormal area, so as to complete the intelligent analysis of multi-dimensional time series data in the abnormal area. Assuming that O_i and R_j represent any multi-dimensional time series data and any abnormal region in the network, they will be converted into binary sequences, namely:

$$D = d_1 d_2 \cdots d_i \cdots d_n \tag{26}$$

where, $d_k = 1$ represents that the multi-dimensional time series data visited the abnormal area R_j at time k, and $d_k = 0$ is the opposite.

The spectrum sequence function $X\left(f_{k/N}\right)$ can be obtained by discrete Fourier transform of the binary sequence D in the above abnormal area. The specific calculation formula is as follows:

$$X\left(f_{k/N}\right) = \frac{\sum_{n=0}^{N-1} d_n e^{-\frac{2\pi}{N} kn}}{\sqrt{N}} \tag{27}$$

In the formula, the subscript k/N represents the frequency captured by the coefficients of each discrete Fourier series, and n represents the imaginary unit, then the

periodogram of the abnormal area of the network can be obtained as:

$$\Theta_k = \left\| X\left(f_{k/N}\right) \right\|^2 \tag{28}$$

A series of candidate periodic charts have been obtained due to the spectrum leakage of multidimensional time series data in the network or other reasons. In order to avoid incorrect and false alarms, the candidate periodic charts are checked by introducing an autocorrelation function to determine the change cycle of multidimensional time series data in abnormal areas, and the intelligent analysis of multidimensional time series data in abnormal areas of the network is completed. The specific calculation formula is as follows:

$$\Gamma_f(\tau) = \sum_{n=1}^{N-1} d_\tau d_{n+\tau} \Theta_k \tag{29}$$

In summary, deep learning is used to mine outliers in multidimensional time series data, process multidimensional time series data through dimensionality reduction, extract features of multidimensional time series data, and combine multidimensional time series data analysis algorithm design to realize intelligent analysis of multidimensional time series data.

3 Experimental Analysis

3.1 Experimental Environment and Settings

The experiment was run on Windows 10 operating system, with AMD Ryzen3600X CPU, 16 GB memory and NVIDIA Ge Force RTX 2060 graphics card. The deep learning platform Keras2.0.8 is used for network model building and training, and Tensor Flow GPU 1.4 is called to complete the accelerated operation. The parameter settings of each layer of Bi ConvLSTM Bi LSTM predictor are shown in Table 1, where filters and kernel size represent the number and size of convolution cores, activation is the activation function used, and merge mode represents the combination mode of bidirectional RNN output. In order to prevent accidental experiment, all experimental results are the average values obtained from five experiments.

3.2 Experimental Dataset

The Yahoo Webscope dataset is an open source time series data analysis dataset. The dataset consists of 4 sub-datasets A1 to A4, with a total of 367 time series, each of which consists of 1420 to 1680 instances. This paper uses 50 time series data in the A2 sub-dataset with strong data periodicity as the training set and 50 time series data as the test set to compare the performance of our method and other data analysis methods.

Since the A2 sub data set only contains point exceptions, in order to increase the test demand for context exceptions, this paper randomly selects an instance in each dimension of the test set as the starting position, replaces the original data in this dimension with continuous random values that fluctuate up and down no more than 30% of the value,

Table 1. Bi-ConvLSTM-Bi-LSTM predictor parameter settings

layer name	units	filters	kernel size	return sequences	activation	merge mode
Bidirectional Conv LSTM2D	-	32	(3,3)	true	Re LU	concat
Bidirectional Conv LSTM2D	-	32	(3,3)	true	Re LU	concat
flatten	-	-	-	-	-	-
Repeat Vector	No. of prediction	-	-	-	-	-
Bidirectional LSTM	80	-	-	true	-	ave
Time Distributed Dense	No. of prediction	-	-	-	-	-

and simulates context exceptions in practical applications, and the length of all context exceptions is between 50 and 120 instances. Use the above method to add 0–2 context exceptions to each dimension of the test set, and all context exceptions are independent in each dimension. The modified test set contains 99 point exceptions and 51 context exceptions.

3.3 Evaluation Indicators

In order to evaluate the performance of intelligent analysis of multidimensional time series data, three error indicators, mean absolute error (MAE) and root mean square error (RMSE), were used to evaluate the results of intelligent analysis of multidimensional time series data. Its formula is as follows:

$$MAE = \frac{1}{n} \sum_{i=1}^{n} |X - \hat{X}| \tag{30}$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{t=1}^{n} (X - \hat{X})^2} \tag{31}$$

where, X is the real time series data, $\hat{X}$ is the analyzed time series data, and n is the length of the time series data.

3.4 Result Analysis

In order to highlight the advantages of the method in this paper, the analysis method based on bidirectional LSTM, the fusion statistics method and the bidirectional convolution

LSTM analysis method are introduced for comparison, and the following results are obtained.
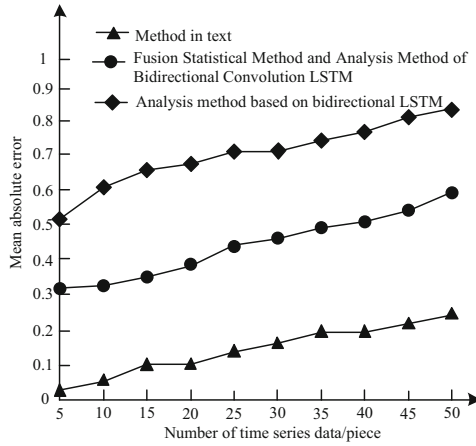

Fig. 4. Mean absolute error test results

From the results in Fig. 4, it can be seen that in the mean absolute error test of the intelligent analysis of multi-dimensional time series data, compared with the analysis method based on bidirectional LSTM, the fusion statistical method and the analysis method of bidirectional convolution LSTM, the method in this paper has better performance. The average absolute error value is within 0.25, indicating that the method in this paper is more suitable for intelligent analysis of multi-dimensional time series data. The reason why the method in this paper has lower average absolute error is that it accurately excavates outliers of multidimensional time series data through deep learning network, and constructs covariance matrix in the feature space to complete the dimensionality reduction of data. Extract the characteristics of multidimensional time series data and complete the intelligent analysis of data. However, traditional methods ignore the process of dimension reduction and feature extraction, resulting in high average absolute error of analysis.

The test results of the three methods in the intelligent analysis of the root mean square error of multi-dimensional time series data are shown in Fig. 5.

It can be seen from the results in Fig. 5 that the root mean square error of the method in this paper when analyzing multidimensional time series data is between 0.1 and 0.3. When the analysis method based on bidirectional LSTM, fusion statistics method and bidirectional convolution LSTM are used, the minimum root mean square error of multidimensional time series data is 0.4 and 0.6 respectively. With the increase of data volume, the root mean square error of multidimensional time series data analysis gradually increases. When the data volume reaches 50, the root mean square error of data analysis reaches 0.77 and 0.95, which shows that the method in this paper has better performance in analyzing root mean square error. The reason why the method in this paper has lower root mean square error in the process of data analysis is that the method
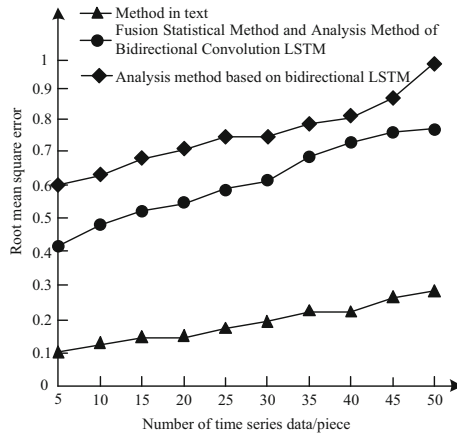
Fig. 5. Root mean square error test results

in this paper constructs a deep learning network model, accurately excavates the outliers of multi-dimensional time series data based on the distribution function of correctly classified and incorrectly classified data, thus reducing the root mean square error of data analysis.

4   Conclusion

In this paper, deep learning is applied to the intelligent analysis of multi-dimensional time series data. Experimental tests show that this method can reduce the average absolute error and root mean square error of data analysis when analyzing multi-dimensional time series data. However, there are still many deficiencies in this research. In the future research, we hope to introduce wavelet coefficients to decompose and reconstruct multidimensional time series data to avoid data distortion.

References

1. Shaukat, F., Mir, J., Linsen, L., et al.: Interactive visual analytics tool for multidimensional quantitative and categorical data analysis. Inf. Vis. **19**(3), 234–246 (2020)
2. Zhao, P., Chang, X., Wang, M.: A novel multivariate time-series anomaly detection approach using an unsupervised deep neural network. IEEE Access **9**(2), 109025–109041 (2021)
3. Tao, T., Zhou, X., Ma, B., et al.: Abnormal time series data detection of gas station by Seq2Seq model based on bidirectional long short-term memory. J. Comput. Appl. **39**(3), 924–929 (2019)
4. Xia, Y., Han, X.: Multi-dimensional time series data anomaly detection fusing statistical methods and bidirectional convolutional LSTM. Appl. Res. Comput. **39**(05), 1362–1367+1409 (2022)
5. Meng, C., Jiang, X.S., Wei, X.M., et al.: A time convolutional network based outlier detection for multidimensional time series in cyber-physical-social systems. IEEE Access **56**(2), 451–455 (2020)

6. Zhao, L., Shi, Z., Shi, Q., et al.: Network traffic classification method based on time series features. J. North Univ. China (Nat. Sci. Edn.) **43**(03), 221–228 (2022)
7. Lu, W., Shan, D., Pedrycz, W., et al.: Granular fuzzy modeling for multidimensional numeric data: a layered approach based on hyperbox. IEEE Trans. Fuzzy Syst. **27**(4), 775–789 (2019)
8. Fei, G., Shaoxu, S., Jianmin, W.: Time series data cleaning under multi-speed constraints. J. Softw. **32**(03), 689–711 (2021)
9. He, Y., Liu, C., Yang, Z.: Temporal dependency mining from multi-sensor event sequences for predictive maintenance. J. Chin. Comput. Syst. **42**(11), 2307–2312 (2021)
10. Gan, X., Tang, X.: Mining model of association rules for temporal data based on CNN. Comput. Simul. **38**(03), 282–285+326 (2021)