

## Research Article

# A Novel Long- and Short-Term Memory Network with Time Series Data Analysis Capabilities

Mu Qiao<sup>1</sup> and Zixuan Cheng<sup>2</sup> 

<sup>1</sup>*School of Mathematics, Jilin University, Changchun 130021, China*

<sup>2</sup>*School of Electronic and Information Engineering, Changchun University, Changchun 130022, China*

Correspondence should be addressed to Zixuan Cheng; [zixuan\\_cheng@163.com](mailto:zixuan_cheng@163.com)

Received 3 September 2020; Revised 26 September 2020; Accepted 30 September 2020; Published 14 October 2020

Academic Editor: Yi-Zhang Jiang

Copyright © 2020 Mu Qiao and Zixuan Cheng. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Time series data are an extremely important type of data in the real world. Time series data gradually accumulate over time. Due to the dynamic growth in time series data, they tend to have higher dimensions and large data scales. When performing cluster analysis on this type of data, there are shortcomings in using traditional feature extraction methods for processing. To improve the clustering performance on time series data, this study uses a recurrent neural network (RNN) to train the input data. First, an RNN called the long short-term memory (LSTM) network is used to extract the features of time series data. Second, pooling technology is used to reduce the dimensionality of the output features in the last layer of the LSTM network. Due to the long time series, the hidden layer in the LSTM network cannot remember the information at all times. As a result, it is difficult to obtain a compressed representation of the global information in the last layer. Therefore, it is necessary to combine the information from the previous hidden unit to supplement all of the data. By stacking all the hidden unit information and performing a pooling operation, a dimensionality reduction effect of the hidden unit information is achieved. In this way, the memory loss caused by an excessively long sequence is compensated. Finally, considering that many time series data are unbalanced data, the unbalanced *K*-means (*UK*-means) algorithm is used to cluster the features after dimensionality reduction. The experiments were conducted on multiple publicly available time series datasets. The experimental results show that LSTM-based feature extraction combined with the dimensionality reduction processing of the pooling technology and cluster processing for imbalanced data used in this study has a good effect on the processing of time series data.

## 1. Introduction

Time series data are a common type of data in work and life. The time series dataset is a collection of observations at different moments collected with a certain collection technology and at certain time intervals. Therefore, each observation result in time series data is often time stamped. With the continuous improvement of computer technology and storage capabilities, storage devices store many time series data. Time series data are widely generated and exist in various industries or fields. For example, camera systems in shopping malls collect a large amount of consumer-related information. The data storage departments of large financial centers and security companies collect a large amount of stock information. In terms of environmental data, real-time

weather information can be observed by artificial satellites, and a large amount of geological and mineral data can also be detected by related instruments. In meteorology, the recorded data of precipitation, temperature, and air quality pollution in each city or region are all time series data. In e-commerce consumption, customer consumption habits, commodity transaction volume, logistics data, and commodity evaluation data are usually time series data. Therefore, research on time series data exists in all walks of life. Extracting the required information from time series data has important practical significance.

Time series data have the characteristics of large data volume, high dimensionality, and continuous updating; so, time series data are relatively complicated. Because of the complexity of time series data, analysis and research on time

series data have not achieved great results. The initial study of time series data analysis can be mainly categorized into several major periods, such as descriptive time series analysis, statistical time series analysis, frequency domain analysis, time domain analysis, and time series data mining. In the past 20 years, the analysis of time series data has received widespread attention, and various time series data research methods have been proposed. Related research mainly includes similar time series research [1–3], time series search and query [4], dimensionality reduction [5, 6], segmentation [7, 8], anomaly detection [9, 10], topic discovery [11], prediction [12, 13], clustering [14–22], classification [23–25], and segmentation [26, 27].

This article focuses on the clustering analysis of time series data using LSTM combined with pooling technology for feature extraction and the UK-means for clustering. The work of this research is summarized as follows:

- (1) The general method of using an RNN for feature extraction is to select the last hidden unit of the network to express the data. However, the original time series data cannot be expressed well when using only the last layer, so this article will use pooling technology to combine the hidden layer expression of all time steps. This method effectively reduces the data dimension while preserving the original information to the greatest extent.
- (2) A  $K$ -means clustering algorithm suitable for imbalanced data is introduced. The algorithm first oversamples the dataset to construct multiple balanced training subsets. Second, traditional  $K$ -means is used to cluster each subset and get the clustering result of each training subset. Finally, the integrated strategy is used to get the final clustering result.
- (3) The dimensionality-reduced feature data are input into the UK-means model to obtain the clustering results for the time series data. By comparing the experimental results of different feature extraction methods, dimensionality reduction techniques, and clustering models, it is verified that the method used in this study has the best analysis effect on time series data.

## 2. Related Information

**2.1. Introduction to Time Series Data.** Time series data refer to data composed of sequence values or events that change over time. Sequence values are usually data measured at the same time interval, such as the daily closing price of stocks, daily fluctuations in material prices, futures trading data, and medical data. There are two main types of time series data in reality: continuous and numerical. Continuous data mainly describe the continuous occurrence of events or business activities in time. For example, the sales and development time and the development and production time of certain products. There are many types of continuous numeric values generated in specific commercial marketing activities, such as categorical variables, numeric values, date types, currency amounts, duration, and counts. The subtype

values mainly describe the relationship between a time series and other business activities or other derivative data, for example, whether the appearance or function of a certain product plays a key role in the sales of the entire enterprise.

**2.2. Time Series Data Analysis Technology.** The time series data analysis is mainly for prediction, classification, and anomaly detection. Regardless of the specific purpose, the technology used can be roughly divided into two categories. One is based on traditional analysis techniques. The second is based on deep learning technology.

**2.2.1. Traditional Analysis Technology.** Traditional analysis methods can be divided into two categories: qualitative analysis and quantitative analysis. Qualitative analysis is often used to predict trends, and judgments can be made without referring to industry values. These values include an industry's previous sales volume, market competition intensity, product planning strategy, and many other issues. Industry experts use a large amount of existing data and, then, use their own personal analysis to synthesize conclusions and judge the trend in a certain indicator in the next stage in the future [28]. When market data are not accurate and complete, especially when there are no numerical measurement data, this method is the main reference method for product sales analysis in the industry. This method relies on the long-term experience and intuition of industry experts and can give results quickly, but it is not accurate enough. Qualitative forecasting is a common time series analysis method in practice [29].

Quantitative analysis is a numerical analysis method. The biggest difference is that quantitative analysis relies on objective numbers, while qualitative analysis relies on intuition. Quantitative analysis is more accurate than qualitative analysis, and of course, the corresponding time complexity increases accordingly. Quantitative analysis is more convincing for more precise issues such as changes in product sales and fluctuations in traffic flow. It can objectively discover inherent statistical laws through numbers. Currently, quantitative analysis mainly includes regression analysis methods [30], traditional statistical analysis methods [31–33], and machine learning [15–18, 34–39] methods.

**2.2.2. Deep Learning Technology.** The theory of deep learning was developed based on the artificial neural network model. Deep learning methods introduce deeper network structures based on artificial neural networks and conduct deeper analyses of extracted features and timing relationships in data. For example, the pretraining method proposed in a deep belief network model allows an artificial neural network to initialize the weights before optimizing training so that the values will not be too far from the ideal value. This processing method can significantly save computing resources. After pretraining, fine tuning is used for training. A deep network can be realized through these two methods. For deep network models, a layer-by-layer training



can also prevent overfitting. Its purpose is to use a value to represent a small area. Commonly used methods include maximum pooling, average pooling, and additive pooling. Maximum pooling is often applied to images because this method can ensure rotational invariance and expansion of the image. For time series data, to make the information of each time dimension supplement the last dimension, it is most appropriate to use average pooling or additive pooling [44]. Because it is easy to obtain a relatively large value and increase the amount of calculation in additive pooling, this study uses average pooling. The principle comparison diagrams of the maximum pooling and average pooling operations are shown in Figure 2. A  $2 \times 2$  window is selected in the figure.

Figure 2 shows that average pooling takes the average value in the window, and maximum pooling takes the maximum value in the window. In the pooling operation, in addition to selecting the size and value of the window, it is also necessary to select the span of the window, that is, the stride of the window moving on the matrix. The span in Figure 2 is 2. In actual applications, the span is sometimes selected as 1. This can easily capture the relationship of translational invariance.

**2.4. Unbalanced K-Means Algorithm.** The traditional  $K$ -means algorithm is mainly aimed at clustering analysis of balanced data. The time series data are mostly unbalanced data. When traditional  $K$ -means processes such data, the effect will drop a lot. In order to improve the clustering performance of the  $K$ -means algorithm on unbalanced data, this paper uses an unbalanced  $K$ -means algorithm ( $UK$ -means). The principle of the algorithm is shown in Figure 3.

It can be observed from Figure 3 that  $UK$ -means first oversampled the dataset and randomly selected  $M$  data from the majority of categories, and each data subset contained  $N$  samples. Each majority class sample subset and the minority class sample set are fused into  $M$  datasets with a sample size of  $2N$ . Secondly,  $K$ -means clustering is performed on each training subset to obtain the clustering result of the training subset. Finally, the results of each training subset are averagely weighted to obtain the final clustering result.

### 3. Time Series Data Clustering Method

**3.1. Framework of the Method Used.** For cluster analysis of time series data, this paper uses a clustering framework that combines LSTM and pooling technology. Figure 4 shows a framework diagram of the cluster analysis used in this study. In Figure 4, the original sequence is first input one by one into the LSTM for feature extraction, and the hidden states obtained at each moment are gathered. Second, average pooling technology is used to reduce the dimensionality to obtain the transformed data expression. Finally, the unbalanced data clustering algorithm is used to cluster the converted data. This study uses the  $UK$ -means model for clustering operations. The introduction of  $UK$ -means is in section 2.4. In summary, the main idea of this research is to use LSTM to extract features and reduce dimensionality of

datasets. Then,  $UK$ -means clustering feature data are used to get the final clustering result. The flow chart of this method is shown in Figure 5.

**3.2. Parameter Update in the LSTM Model.** The mean square error is used to obtain the target expression as follows:

$$F = \frac{\sum_{i=1}^N \sum_{t=1}^T (y_t^i - y_t^{i'})^2}{N * T}, \quad (10)$$

$$\begin{aligned} \Delta_t &= \frac{\partial F}{\partial y_t}, \\ \delta_j^t &= \frac{\partial F}{\partial \alpha_j^t}, \\ \epsilon_c^t &= \frac{\partial F}{\partial b_c^t}, \\ \epsilon_s^t &= \frac{\partial F}{\partial s_c^t}. \end{aligned} \quad (11)$$

We differentiate the input gate, forget gate, output gate, and internal state and obtain their expressions as follows:

Input gate,

$$\delta_j^t = \text{act}'(\alpha_j^t) \sum_{c=1}^C \tanh(\alpha_c^t) \epsilon_s^t. \quad (12)$$

Forget gate,

$$\delta_\phi^t = \text{act}'(\alpha_\phi^t) \sum_{c=1}^C s_c^{t-1} \epsilon_s^t. \quad (13)$$

Output gate,

$$\delta_w^t = \text{act}'(\alpha_w^t) \sum_{c=1}^C \tanh(s_c^t) \epsilon_s^t. \quad (14)$$

Internal state,

$$\delta_c^t = b_j^t \tanh'(\alpha_c^t) \epsilon_s^t. \quad (15)$$

The cell state can be obtained from the hidden layer or from the cell state at a later time. Therefore, the cell state gradient is as follows:

$$\epsilon_s^t = b_w^t \tanh'(s_c^t) \epsilon_c^t + b_\phi^{t+1} \epsilon_s^{t+1}. \quad (16)$$

The gradient update formula of the hidden layer is

$$\epsilon_c^t = w_{c1} \Delta_t^i + \sum_{s \in \{l, w, c, \phi\}} \sum_{l=1}^H w_{cs} \delta_s^{t+1}. \quad (17)$$

The weight parameters mainly include  $w_{h1}$ ,  $w_{il}$ ,  $w_{i\phi}$ ,  $w_{ic}$ ,  $w_{iw}$ ,  $w_{hl}$ ,  $w_{h\phi}$ ,  $w_{hc}$ , and  $w_{hw}$ . These weight parameters can be divided into 3 categories, namely,  $w_{h1}$ ,  $w_i$  ( $w_{il}$ ,  $w_{i\phi}$ ,  $w_{ic}$ ,  $w_{iw}$ ), and  $w_h$  ( $w_{hl}$ ,  $w_{h\phi}$ ,  $w_{hc}$ ,  $w_{hw}$ ). The gradient descent method is used to update these three types of weight parameters, and the expression is as follows:

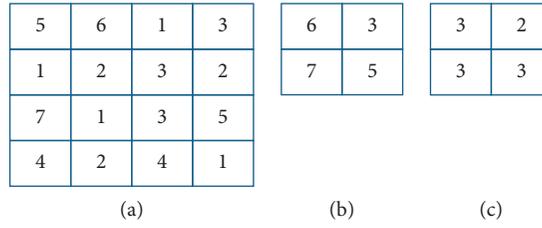


FIGURE 2: Schematic diagram of pooling operation. (a) Operation matrix to be pooled. (b) Max pooling. (c) Average pooling.

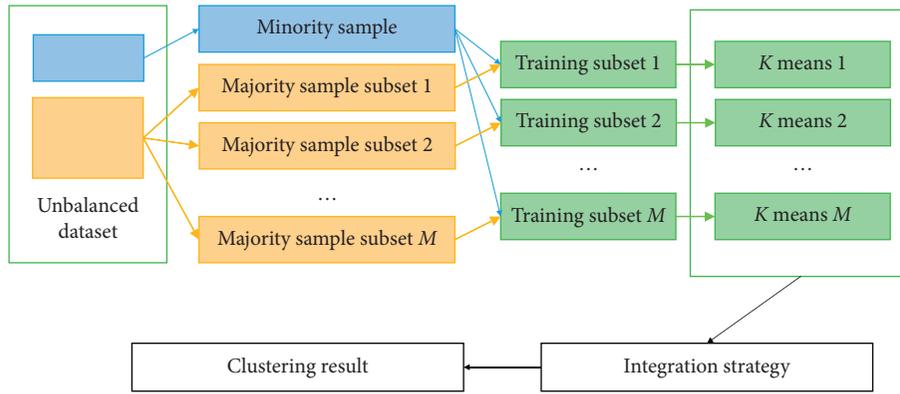


FIGURE 3: UK-means schematic diagram.

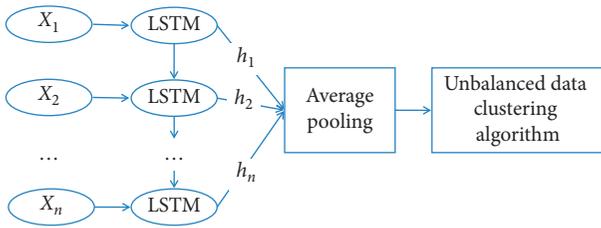


FIGURE 4: Framework of the method.

$$w_{h1} = w_{h1} - \mu \sum_{t=1}^T \Delta_t b_h^t, \quad (18)$$

$$w_i = w_i - \mu \sum_{t=1}^T x_i^t \delta_\alpha^t, \quad (19)$$

$$w_h = w_h - \mu \sum_{t=1}^T b_h^{t-1} \delta_b^t. \quad (20)$$

### 4. Experimental Results and Analysis

4.1. *Experimental Data.* This article conducts experiments on the UCR public dataset. The UCR dataset provides multiple sets of datasets based on real scenes, each of which contains training sets and test sets in the same format. Each row in each dataset is recorded as a piece of data, the first number of a piece of data is the category, and the remaining part is a time series. For ease of use, all data in the UCR are standardized using z-scores. All the experiments use the UCR’s default training set and test set without redivision.

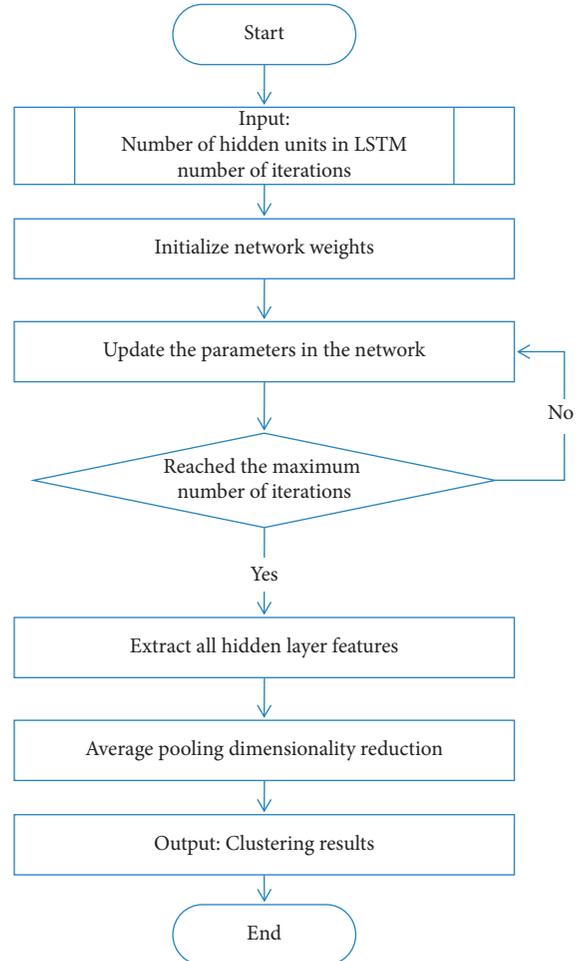


FIGURE 5: Flow chart of this method for clustering time series data.

The introduction of the dataset used in the experiment is shown in Table 2.

*4.2. Experimental Environment and Evaluation Index.* The implementation language for all algorithms in this study is python 2.7. In the hardware environment, the CPU is an Intel core i7-7700k 4.4 G, the memory is 16 G, and the graphics card is an NV GTX 1070. This study uses the accuracy (Acc) and *F*-score as two evaluation indicators to evaluate the clustering performance of the clustering framework used on time series data. The calculation formulas for the evaluation indices are shown as follows:

$$\text{Acc} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}, \quad (21)$$

$$F\text{-score} = \frac{2 * P * R}{P + R}, \quad (22)$$

$$P = \frac{\text{TP}}{\text{TP} + \text{FP}}, \quad (23)$$

$$R = \frac{\text{TP}}{\text{TP} + \text{FN}}, \quad (24)$$

The larger the Acc value, the better the clustering performance. When the values of P and R are both high, the larger the *F*-score value is, the better the clustering effect is. The *F*-score is more suitable for the classification and evaluation of unbalanced data. A description for each parameter in the abovementioned formula is shown in Table 3. In Table 3, P stands for positive, N stands for negative, T stands for true, and F stands for false.

### 4.3. Experimental Results and Analysis

*4.3.1. Experiments on Different Clustering Algorithms.* The samples are input to the LSTM network for feature extraction, and then, the average pooling technology is used to reduce the dimensionality. Finally, different clustering algorithms are used for the transformed features to explore the influence of different clustering algorithms on the clustering effect on time series data. The comparison algorithms used are *K*-means [17], fuzzy *C*-means clustering (FCM) [16], soft subspace clustering (SSC) [18], and unbalanced *K*-means clustering (*UK*-means) [38]. The experimental results are shown in Table 4.

The experimental results in Table 4 verify the following three conclusions. (1) Under the premise of the same feature extraction and dimensionality reduction technology, different clustering algorithms will have different clustering effects. (2) On the Adiac, Beef, CBF, and ChlorineCon datasets, the *UK*-means algorithm in the sixth column has the best clustering effect. This shows that the clustering effect obtained by using the unbalanced clustering algorithm is better. For time series data, an unbalanced clustering algorithm is more suitable for clustering analysis. (3) On the CinCECGTorso and Coffee datasets, the clustering effect under the SSC algorithm is the best. This is because the SSC

TABLE 2: Dataset introduction.

Dataset	Number of categories	Training set	Test set	Time series length
Adiac	37	390	391	176
Beef	5	30	30	470
CBF	3	30	900	128
ChlorineCon	3	467	3840	166
CinCECGTorso	4	40	1380	1639
Coffee	2	28	28	286

TABLE 3: Description of each parameter in the evaluation index.

		Prediction	
		P	N
Actual	P	TP	FN
	N	FP	TN

algorithm not only reflects the relationship between the sample attributes and clusters but also reflects differences in the related attributes. The clustering results obtained by the *UK*-means on these two datasets are slightly lower than those obtained by SSC, but the performance difference is not very large. This shows that the *UK*-means can also be used on unbalanced datasets without significantly reducing the clustering performance.

*4.3.2. Experiments on Different Dimensionality Reduction Techniques.* After the sample is output by the LSTM network, the feature vector dimension at this time is still very large. This study uses an average pooling technique to reduce its dimensionality. To verify the effectiveness of this dimensionality reduction method, the method of no dimensionality reduction, the maximum pooling method, and the average pooling method are compared here. The final clustering algorithm uses the *UK*-means. The comparative experimental results are shown in Table 5.

The third column in Table 5 shows the clustering results obtained without adding dimensionality reduction technology. The fourth column shows the clustering results obtained by using the maximum pooling technique. The fifth column shows the clustering results obtained by dimensionality reduction using average pooling technology. Clearly, the value in column 5 is significantly greater than that in column 3. This shows that the method of using pooling technology to enhance the expression ability of the hidden layer is very effective. The values in column 5 are slightly larger than the values in column 4, which shows that the average pooling technique is more suitable for dimensionality reduction processing on time series data.

*4.3.3. Feature Extraction Experiment.* This research uses the LSTM model to extract the features from the input samples. To verify its effectiveness, the contrast feature extraction method used is the wavelet transform [45]. In the framework of the entire clustering method, the dimensionality reduction technology still uses the average pooling technology, and the clustering algorithm uses the *UK*-means algorithm. The final experimental results are shown in Table 6.

TABLE 4: Clustering results of different clustering algorithms.

Dataset	Index	<i>K</i> -means	FCM	SSC	<i>UK</i> -means
Adiac	Acc	0.6717	0.6803	0.6987	0.7097
	<i>F</i> -score	0.6385	0.6417	0.6542	0.6686
Beef	Acc	0.5895	0.5998	0.6231	0.8110
	<i>F</i> -score	0.6589	0.6676	0.6889	0.7874
CBF	Acc	0.6367	0.6432	0.8129	0.8724
	<i>F</i> -score	0.5764	0.5810	0.7675	0.8289
ChlorineCon	Acc	0.5577	0.5618	0.5820	0.6887
	<i>F</i> -score	0.6349	0.6446	0.6527	0.6510
CinCECGTorso	Acc	0.6726	0.6823	0.6956	0.6854
	<i>F</i> -score	0.5527	0.5598	0.5721	0.5938
Coffee	Acc	0.6832	0.6788	0.6890	0.6882
	<i>F</i> -score	0.7276	0.7142	0.7289	0.7245

TABLE 5: Clustering results using different dimensionality reduction techniques.

Dataset	Index	No dimensionality reduction	Maximum pooling technology	Average pooling technology
Adiac	Acc	0.6477	0.7012	0.7097
	<i>F</i> -score	0.6039	0.6594	0.6686
Beef	Acc	0.5787	0.7658	0.8110
	<i>F</i> -score	0.6452	0.8356	0.7874
CBF	Acc	0.6212	0.8423	0.8724
	<i>F</i> -score	0.5644	0.7790	0.8289
ChlorineCon	Acc	0.5456	0.5478	0.6887
	<i>F</i> -score	0.5238	0.5276	0.6510
CinCECGTorso	Acc	0.5998	0.6212	0.6854
	<i>F</i> -score	0.5485	0.5645	0.5938
Coffee	Acc	0.6335	0.6021	0.6882
	<i>F</i> -score	0.6186	0.5897	0.7245

TABLE 6: Clustering results using different feature extraction methods.

Dataset	Index	Wavelet transform	LSTM
Adiac	Acc	0.6870	0.7097
	<i>F</i> -score	0.6589	0.6686
Beef	Acc	0.7806	0.8110
	<i>F</i> -score	0.8535	0.7874
CBF	Acc	0.8832	0.8724
	<i>F</i> -score	0.8398	0.8289
ChlorineCon	Acc	0.6736	0.6887
	<i>F</i> -score	0.6062	0.6510
CinCECGTorso	Acc	0.6321	0.6854
	<i>F</i> -score	0.5752	0.5938
Coffee	Acc	0.6967	0.6882
	<i>F</i> -score	0.7345	0.7245

The fourth column of Table 6 shows the experimental results of feature extraction using the LSTM model. The third column shows the experimental results obtained by using the wavelet transform for feature extraction. On the datasets Adiac, Beef, ChlorineCon, and CinCECGTorso, the values in the fourth column are all greater than those in the third column. This shows that the LSTM-based feature extraction method works better on these 4 datasets. On the datasets CBF and Coffee, the values in the third column are greater than the values in the fourth column, which shows

that the clustering results obtained by the wavelet transform are slightly better than those obtained by LSTM. However, the difference in the effect is not very large. The clustering effect when using the LSTM network for feature extraction on 4 datasets (Adiac, Beef, ChlorineCon, and CinCECGTorso) is better than that based on the wavelet transform. On the remaining 2 datasets (CBF and Coffees), the effects of the two feature extraction methods are not much different. Considering the details, it is more sensible to use the LSTM method for feature extraction.

## 5. Conclusions

Because time series data have many special properties, commonly used clustering algorithms cannot achieve satisfactory results when clustering time series data. The purpose of this research is to find suitable models for various time series data. The research on time series data generally focuses on the chronological nature of time series data. To capture this property, this study uses an RNN that can process the data in chronological order to train the data. Due to the gradient problem in traditional RNNs, they have shortcomings in practical applications. This study uses a special CNN, namely, LSTM, to learn the dimensionality reduction expression of a time series. For long time series, the hidden layer of the network cannot remember all of the time information. As a result, it is difficult to compress the global information in the last layer. In response to this problem, this research performs an average pooling operation after stacking all the hidden unit information to further complete the dimensionality reduction of the data. Finally, the *UK-means* algorithm is used to cluster the feature data after dimensionality reduction. The experiments are conducted on multiple UCR public datasets. The experimental results verify the effectiveness of the clustering algorithm used. The method used in this research involves a large number of matrix operations, such as matrix addition and matrix multiplication. This requires high hardware performance. The next step will be to study how to improve the efficiency of the algorithm.

## Data Availability

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

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This study was supported by the Scientific Research Project of Jilin Education Department.

## References

- [1] W. R. Pearson and A. J. Mackey, "Using SQL databases for sequence similarity searching and analysis," *Current Protocols in Bioinformatics*, vol. 59, no. 1, pp. 1–22, 2017.
- [2] J. Tong, R. I. Sadreyev, J. Pei, L. N. Kinch, and N. V. Grishin, "Using homology relations within a database markedly boosts protein sequence similarity search," *Proceedings of the National Academy of Sciences*, vol. 112, no. 22, pp. 7003–7008, 2015.
- [3] D. Berndt and J. Clifford, "Finding patterns in time series: a dynamic programming approach," *Advances in Knowledge Discovery & Data Mining*, pp. 229–248, 1996.
- [4] T. Kahveci and A. K. Singh, "Optimizing similarity search for arbitrary length time series queries," *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 4, pp. 418–433, 2004.
- [5] E. Keogh, K. Chakrabarti, M. Pazzani, and S. Mehrotra, "Dimensionality reduction for fast similarity search in large time series databases," *Knowledge and Information Systems*, vol. 3, no. 3, pp. 263–286, 2001.
- [6] J. Song, G. Yoon, H. Cho, and S. M. Yoon, "Structure preserving dimensionality reduction for visual object recognition," *Multimedia Tools and Applications*, vol. 77, no. 18, pp. 23529–23545, 2018.
- [7] A. Janos, F. Balazs, N. Sandor, and A. Peter, "Modified Gath–Geva clustering for fuzzy segmentation of multivariate time-series," *Fuzzy Sets and Systems*, vol. 149, no. 1, pp. 39–56, 2005.
- [8] N. Wang, X. Liu, and J. Yin, "Improved Gath-Geva clustering for fuzzy segmentation of hydrometeorological time series," *Stochastic Environmental Research and Risk Assessment*, vol. 26, no. 1, pp. 139–155, 2012.
- [9] G. M. Weiss, "Mining with rarity," *ACM SIGKDD Explorations Newsletter*, vol. 6, no. 1, pp. 7–19, 2004.
- [10] M. Hu, Z. Ji, K. Yan et al., "Detecting anomalies in time series data via a meta-feature based approach," *IEEE Access*, vol. 6, pp. 27760–27776, 2018.
- [11] P. Langley, "Scientific discovery, causal explanation, and process model induction," *Mind & Society*, vol. 18, no. 1, pp. 43–56, 2019.
- [12] A. Bauer, M. Züfle, N. Herbst, A. Zehe, A. Hotho, and S. Kounev, "Time series forecasting for self-aware systems," *Proceedings of the IEEE*, vol. 108, no. 7, pp. 1068–1093, 2020.
- [13] A. D. Shatashvili, I. S. Didmanidze, G. A. Kakhiani, and T. A. Fomina, "A method of preliminary forecasting of time series of financial data," *Cybernetics and Systems Analysis*, vol. 56, no. 2, pp. 296–302, 2020.
- [14] P. Qian, Y. Jiang, Z. Deng et al., "Cluster prototypes and fuzzy memberships jointly leveraged cross-domain maximum entropy clustering," *IEEE Transactions on Cybernetics*, vol. 46, no. 1, pp. 181–193, 2015.
- [15] Y. Jiang, D. Wu, Z. Deng et al., "Seizure classification from EEG signals using transfer learning, semi-supervised learning and TSK fuzzy system," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 25, no. 12, pp. 2270–2284, 2017.
- [16] P. Qian, Y. Jiang, S. Wang et al., "Affinity and penalty jointly constrained spectral clustering with all-compatibility, flexibility, and robustness," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 5, pp. 1123–1138, 2016.
- [17] Y. Jiang, F. L. Chung, S. Wang, Z. Deng, J. Wang, and P. Qian, "Collaborative fuzzy clustering from multiple weighted views," *IEEE Transactions on Cybernetics*, vol. 45, no. 4, pp. 688–701, 2014.
- [18] P. Qian, K. Zhao, Y. Jiang et al., "Knowledge-leveraged transfer fuzzy C -Means for texture image segmentation with self-adaptive cluster prototype matching," *Knowledge-Based Systems*, vol. 130, pp. 33–50, 2017.
- [19] Y. Jiang, F. L. Chung, H. Ishibuchi, Z. Deng, and S. Wang, "Multitask TSK fuzzy system modeling by mining intertask common hidden structure," *IEEE Transactions on Cybernetics*, vol. 45, no. 3, pp. 534–547, 2015.
- [20] P. Qian, S. Sun, Y. Jiang et al., "Cross-domain, soft-partition clustering with diversity measure and knowledge reference," *Pattern Recognition*, vol. 50, pp. 155–177, 2016.
- [21] Y. Zhang and J. Cai, "Fuzzy clustering based on automated feature pattern-driven similarity matrix reduction," *IEEE Transactions on Computational Social Systems*, vol. 99, pp. 1–10, 2020.

- [22] Y. Zhang, F.-L. Chung, and S. Wang, "A multiview and multiexemplar fuzzy clustering approach: Theoretical analysis and experimental studies," *IEEE Transactions on Fuzzy Systems*, vol. 27, no. 8, pp. 1543–1557, 2019.
- [23] B. R. Bakshi and G. Stephanopoulos, "Representation of process trends-IV. Induction of real-time patterns from operating data for diagnosis and supervisory control," *Computers & Chemical Engineering*, vol. 18, no. 4, pp. 303–332, 1994.
- [24] A. Fullah Kamara, E. Chen, L. Qi, and Z. Pan, "Combining contextual neural networks for time series classification," *Neurocomputing*, vol. 384, pp. 57–66, 2020.
- [25] Y. Lei and Z. Wu, "Time series classification based on statistical features," *EURASIP Journal on Wireless Communications & Networking*, vol. 2020, no. 1, pp. 1–13, 2020.
- [26] E. Keog and S. Kasetty, "On the need for time series data mining benchmarks: A survey and empirical demonstration," *Data Mining and Knowledge Discovery*, vol. 7, no. 4, pp. 349–371, 2003.
- [27] K. Joo-Chang and K. Chung, "Mining based time-series sleeping pattern analysis for life big-data," *Wireless Personal Communications*, vol. 105, no. 2, pp. 475–489, 2019.
- [28] A. Sengupta, Ap Prathosh, S. N. Shukla, V. Rajan, and K. Chandan, "Prediction and imputation in irregularly sampled clinical time series data using hierarchical linear dynamical models. Conference proceedings," in *Proceedings of the 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 3660–3663, Jeju island, Republic of Korea, July 2017.
- [29] S. S. Wulff, "Time series analysis: Forecasting and control, 5th edition," *Journal of Quality Technology*, vol. 49, no. 4, pp. 418–419, 2017.
- [30] X. Chen, A. T. K. Wan, and Y. Zhou, "Efficient quantile regression analysis with missing observations," *Journal of the American Statistical Association*, vol. 110, no. 510, pp. 723–741, 2015.
- [31] B. Seong, "Smoothing and forecasting mixed-frequency time series with vector exponential smoothing models," *Economic Modelling*, vol. 91, pp. 463–468, 2020.
- [32] G. Zhang, X. Zhang, and H. Feng, "Forecasting financial time series using a methodology based on autoregressive integrated moving average and Taylor expansion," *Expert Systems*, vol. 33, no. 5, pp. 501–516, 2016.
- [33] A. K. Singh, "Fractionally delayed Kalman filter," *IEEE/CAA Journal of Automatica Sinica*, vol. 7, no. 1, pp. 169–177, 2020.
- [34] P. Qian, Y. Chen, J.-W. Kuo et al., "mDixon-based synthetic CT generation for PET attenuation correction on abdomen and pelvis jointly using transfer fuzzy clustering and active learning-based classification," *IEEE Transactions on Medical Imaging*, vol. 39, no. 4, pp. 819–832, 2020.
- [35] P. Qian, C. Xi, M. Xu et al., "SSC-EKE: Semi-supervised classification with extensive knowledge exploitation," *Information Sciences*, vol. 422, pp. 51–76, 2018.
- [36] Y. Jiang, Z. Deng, F.-L. Chung et al., "Recognition of epileptic EEG signals using a novel multiview TSK fuzzy system," *IEEE Transactions on Fuzzy Systems*, vol. 25, no. 1, pp. 3–20, 2017.
- [37] P. Qian, J. Zhou, Y. Jiang et al., "Multi-view maximum entropy clustering by jointly leveraging inter-view collaborations and intra-view-weighted attributes," *IEEE Access*, vol. 6, pp. 28594–28610, 2018.
- [38] N. M. Faizah, L. F. Surohman, and R. P. Hendra, "Unbalanced data clustering with K-means and euclidean distance algorithm approach case study population and refugee data," *Journal of Physics: Conference Series*, vol. 1477, Article ID 022005, 2020.
- [39] Y. Zhang, S. Wang, K. Xia, Y. Jiang, and P. Qian, "Alzheimer's disease multiclass diagnosis via multimodal neuroimaging embedding feature selection and fusion," *Information Fusion*, vol. 66, pp. 170–183, 2021.
- [40] N. M. Rezk, M. Purnaprajna, T. Nordström, and Z. Ul-Abdin, "Recurrent neural networks: an embedded computing perspective," *IEEE Access*, vol. 8, pp. 57967–57996, 2020.
- [41] J. Cai, J. Hu, X. Tang, T.-Y. Hung, and Y.-P. Tan, "Deep historical long short-term memory network for action recognition," *Neurocomputing*, vol. 407, pp. 428–438, 2020.
- [42] C. Hubert, A. Rivera, M. Farhadloo, and M. A. Pedroza, "Grape detection with convolutional neural networks," *Expert Systems with Applications*, vol. 159, Article ID 113588, 2020.
- [43] Y. Zhang, H. Ishibuchi, and S. Wang, "Deep Takagi-Sugeno-Kang fuzzy classifier with shared linguistic fuzzy rules," *IEEE Transactions on Fuzzy Systems*, vol. 26, no. 3, pp. 1535–1549, June 2018.
- [44] L. Hubert and P. Arabie, "Comparing partitions," *Journal of Classification*, vol. 2, no. 1, pp. 193–218, 1985.
- [45] K. Stratis, C. Stavros-Richard, C. Alexander, and M. Fitzpatrick, "Detecting anomalies in time series data via a deep learning algorithm combining wavelets, neural networks and Hilbert transform," *Expert Systems with Applications*, vol. 85, pp. 292–304, 2017.