Information Systems and e-Business Management
https://doi.org/10.1007/s10257-021-00508-5

ORIGINAL ARTICLE

Research on the application of conditional generative adversarial nets in economic time series data analysis

Li Weiping[1] · Wang Weihan[1]

Received: 25 April 2020 / Revised: 23 June 2020 / Accepted: 14 November 2020

Abstract
The processing of time series data is the key technical field of financial data analysis. With the continuous development of computing science, deep learning has a revolutionary impact on the traditional computing model. Among them, the generative adversarial nets GANs has achieved desirable results in the field of data generation. Revolving around the conditional generative adversarial nets cGANs, an effective Bi-LSTM generator, CNN discriminator and data processing method are designed in this paper. Also, the experiments on two economic datasets including the stock and commodity price are implemented. The results show that compared with the traditional model, the prediction performance of this research method witnesses a great improvement and it can be employed to better deal with the analysis task of non-stationary data, which is a significant point contributed by the research. In addition, the details associated with the generator mode and GANs model optimization are reported and discussed in combination with the actual situation of the experiment, and the existing problems are further explained and discussed.

Keywords Time series data · Economics · Financial engineering · Machine learning · Deep learning · Generative adversarial nets

1 Introduction

Nowadays, with the continuous development of global economy, the complexity of economic system and financial system is increasing, and the high correlation and tight coupling between economic behavior and data are increasing, which makes the traditional methods meet various problems and bottlenecks in dealing with economic data analysis. These problems pose new challenges to the ability

✉ Wang Weihan
wangweihan@pku.edu.cn

Li Weiping
wpli@ss.pku.edu.cn

[1] School of Software and Microelectronics, PKU, Beijing, China

of calculation, accuracy and scale data processing in the field of economic data analysis (Powney et al. 2014; Newman 2014).

With the continuous development of computing science, the mainstream data analysis and calculation mode has evolved from the earlier empirical model method and statistical model method to the machine learning model method, in which deep learning model method attracts more and more researchers' attention in data analysis. The results show that the loop neural networks such as LSTM, GRU and convolution neural networks such as CNN have excellent performance in the analysis of time series data. These methods can be widely and further studied in financial data analysis, and can be applied to practice quickly after gaining sufficient experimental verification.

In the deep learning method, the emergence of generative adversarial nets GANs provides a new impetus for the development of deep learning. The proposal of GANs provides researchers with a better choice of data generation methods in addition to flow generation model and autoregressive generation model. In fact, GANs has made great achievements in the field of image recognition. However, the application of GANs is still in its infancy in the analysis of sequential data (such as time series data, natural language data, audio data, etc.). In a broad sense, there are two main reasons. The one is attributed to the discreteness of sequence data, the other one is due to the high dimension of sequential data. As a result, it is required for researchers to optimize the GANs model and solve practical problems based on sequential data. Whereas, it is worth affirming that there have been targeted research orientations and solutions for the two bottlenecks.

Most of the existing economic and financial data analyses focus on the analysis of single sequential data, for example, forecasting the closing price by virtue of the closing price and forecasting the commodity price by virtue of the commodity price. However, it is unreasonable to describe a certain kind of economic behavior or financial phenomenon based on a single indicator. Only by establishing a multivariate model to implement expression and analysis, can the results be more accurate. Compared with the traditional method, in which the ability and accuracy of multi-sequence analysis are limited to a great extent, further investigation and discussion are conducted by virtue of multi-variable deep learning modeling in this research.

In addition to the introduction of this part, the background information of analysis approach of time series data, deep learning method and the characteristics of economic and financial data is introduced in the corresponding contents of the second part. In the third part, the establishment and training methods of sequence model based on LSTM as well as the selection of hyper-function are introduced, and at the same time, the establishment, training of CGANs model and the selection of hyper-function are introduced and discussed. As for the fourth part, data-associated situations are introduced and there are two kinds of databases are employed in this research, which include stock market database and commodity price database. Then, the research results are reported and discussed in the fifth part and ultimately, the overall research is summarized retrospectively and expected further.

The related fields of this research include econometrics, time series analysis, machine learning and deep learning, financial engineering and complex systems.

2 Related work

2.1 Analysis of time series data

Time series analysis is one of traditional and important kinds of data analysis and data mining. Time series data extensively exists in various fields of human production and life, such as theoretical research, scientific practice, engineering technology, production and management, daily life, etc. (Wei et al. 2018; Liang et al. 2017; Blackwell et al. 2014).

Generally speaking, a time series is a set of random variables sorted according to the time order, which is usually the result of observation of a potential process in accordance with a given sampling rate within an equally spaced period of time. Different from other data, time series data has its own unified and meaningful data structure. Such characteristics determines the interoperability and universal applicability of the analysis method of time series data in different fields. The analysis of time series data includes the classification and prediction of time series data.

Observation and measurement are implemented for a variable or a set of variables x(t), and the sequence set composed of discrete numbers obtained in a series of time t1, t2,…, tn is called the time series. In detail, a set of random variables in chronological order:

$$\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \ldots, \boldsymbol{x}_t \tag{1}$$

Can be employed to represent the time series of a random event, which is abbreviated as:

$$\{\boldsymbol{X}_t, t \in \boldsymbol{T}\} \tag{2}$$

For example: the closing price of a stock A in each trading day from June 1, 2015 to June 1, 2016 can form a time series; also, the highest temperature in a day in a place can form a time series.

Definition of the analysis task of time series data.

The two basic questions of time series analysis are prediction and classification respectively. The following is the definition for the two questions and the first on lies in the basic definition of the prediction of time series.

A time series with a given input length of n is defined as:

$$\{\boldsymbol{X}_t, \boldsymbol{t} = 1, 2, \ldots, \boldsymbol{n}\} \tag{3}$$

The prediction length of time series is defined as m, the prediction task can be expressed as:

$$\boldsymbol{f}(\boldsymbol{X}_t) \rightarrow \{\boldsymbol{x}_{n+1}, \boldsymbol{x}_{n+2}, \ldots, \boldsymbol{x}_{n+m}\} \tag{4}$$

2.2 Traditional method of time series analysis

The traditional methods of time series analysis (whose range is the method of time series data before the appearance of machine learning) include observation method, statistical method and model method, in which such methods as ARIMA method, ARCH method that has won the Nobel Prize in economics, etc. are relatively famous. Whereas, such models also have shortcomings and limitations to some extent. Firstly, the parameters of the model are determined by subjective judgment, which may lead to the local optimal solution rather than the global optimal solution. Secondly, the enlargement of the model order will improve the prediction effect, but this practice may be at the cost of tremendous computing resources because of the poor computing performance. Thirdly, the effect of multi sequence analysis is rather poor. These are also the bottlenecks of traditional methods of time series analysis (Wu 2002; Hron et al. 2011; Stekhoven and Bühlmann 2012).

2.3 Method of time series analysis of deep learning

Although the traditional method of time series analysis has some limitations in the application process, its guiding ideology is of guiding significance for the selection, design and optimization of subsequent algorithms. The development of the method of time series analysis based on machine learning can be divided into two paths. The one is to machine-learn the classical method; the other one is the application of machine learning methods, which include linear regression, decision trees, support vector machines, Bayesian networks, matrix decomposition, Gaussian processes, etc. The results indicate that the effect of machine learning method is better than that of traditional analysis method of time series data. At the same time, more indicators and characteristics are put forward for the modeling of time series and effect evaluation and performance optimization (Devlin et al. 1810; Twisk et al. 2013; Che et al. 2018).

2.3.1 Neural network of long and short time memory (LSTM)

Since the artificial neural network was put forward, a large number of machine learning research has emerged, and achieved desirable achievements in dealing with practical problems. Among many neural networks, recurrent neural network (RNN) is a neural network model dedicated to the study of sequence data. In the process of application, RNN faces the problems of gradient disappearance and sequence generation. In order to solve the problem of long-range correlation and sequence generation in RNN, researchers developed long and short time memory (LSTM) and Encoder-Decoder model. The results show that LSTM model performs well in dealing with sequential data, and the effect will be further improved by adding dual-way processing. In addition, the LSTM model also has some improved forms with better effect and performance, such as GRU model. The mathematical expression of LSTM model is as follows:

$$\boldsymbol{f_t} = \boldsymbol{\sigma_g}\left(\boldsymbol{W_f x_t} + \boldsymbol{U_f h_{t\text{-}1}} + \boldsymbol{b_f}\right) \tag{5}$$

$$i_t = \sigma_g\left(W_i x_t + U_i h_{t-1} + b_i\right) \tag{6}$$

$$o_t = \sigma_g\left(W_o x_t + U_o h_{t-1} + b_o\right) \tag{7}$$

$$c_t = f_t * c_t - 1 + it * \sigma_c\left(W_c x_t + U_c h_{t-1} + b_c\right) \tag{8}$$

$$\boldsymbol{f_t} = \boldsymbol{\sigma_g}\left(\boldsymbol{W_f x_t} + \boldsymbol{U_f h_{t-1}} + \boldsymbol{b_f}\right) \tag{9}$$

The basic structure of LSTM is shown in Fig. 1:

LSTM is a deep learning model with the property of time correlation. The length of time correlation is related to the parameters of the model. LSTM solves the problems of gradient dissipation and remote transmission loss in traditional deep learning models by introducing mechanisms such as forgetting gates. LSTM has the characteristics of long-range correlation, but compared to CNN and other models, the long- range correlation of LSTM is weak. The reason is that the segment data in the sequence received by the LSTM every cycle depends on the memory gate mechanism for the long-range correlation. The parameters in the memory gate are greatly affected by the accumulation in the long sequence. This is a common feature of local regression learning models (Wei et al. 2018; Feng et al. 2018).

2.3.2 Convolutional neural network (CNN)

Convolutional neural network (CNN) is also an extended application of multi-layer perceptron, whose structures include convolutional layer, down-sampling layer and full link layer. There are multi feature maps in each layer and each feature map extracts a feature of input by virtue of a kind of convolution filter and in addition, each feature map possesses multi neurons. CNN is widely used in image recognition and has achieved good results. On account that CNN can express the overall rule of sequence, it

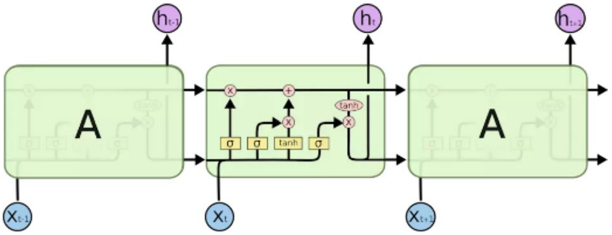

Fig. 1 Classic Structure of LSTM

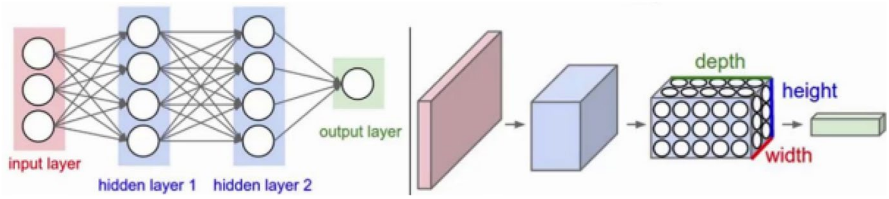

Fig. 2 Structure of CNN model

has been widely adopted in the classification of sequence data processing. The structure of CNN model is shown in Fig. 2 (Sezer and Ozbayoglu 2019).

The mathematical expression of CNN model is:

$$s(t) = (x * w)(t) = \sum_{a=-\infty}^{\infty} x(a)w(t-a) \tag{10}$$

$$S(i,j) = (I * K)(i,j) = \sum_{m}\sum_{n} I(m,n)K(i-m,j-n). \tag{11}$$

$$zi = \sum_{m} Wi,jxi + bi \tag{12}$$

$$y = softmax(z) \tag{13}$$

$$softmax(zi) = \frac{\exp(zi)}{\sum_j \exp(zj)} \tag{14}$$

$$s(t) = (x * w)(t) = \sum_{a=-\infty}^{\infty} x(a)w(t-a) \tag{15}$$

2.3.3 Generative adversarial nets (GANs)

The appearance of generative adversarial nets provides researchers with a new and better choice of the construction method of model generation. GANs can be employed to locate the optimal equilibrium between certainty and randomness, real data and noise data, flow model and autoregressive model, so as to facilitate the generated model to be better restore the laws of reality. The basic structure of generative adversarial nets is shown in Fig. 3:

The objective function is expressed as:

$$\min_{G}\max_{D} V(D,G) = E_{x\sim P_{data}}\left[logD(x)\right] + E_{x\sim P_G}\left[log(1-D(x))\right] \tag{16}$$

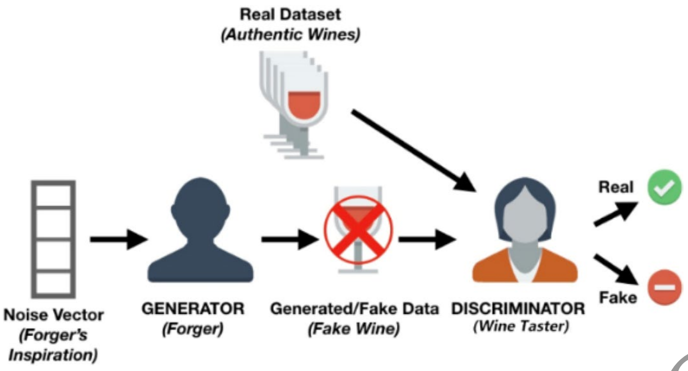

Fig. 3 The basic structure of GANs

It can be seen that the function itself represents the optimization of Max–Min problem, which is also corresponding to the process of optimization training of G and D. In fact, the optimization training of GANs cannot be completed by virtue of a simple procedure. Generally speaking, the D shall be optimized at first, followed by the optimization of G. When the discrimination network D is optimized, the optimization function D can be expressed as:

$$\max_{D} V(D,G) = E_{x\sim P_{data}}\left[logD(x)\right] + E_{x\sim P_G}\left[log(1-D(x))\right] \tag{17}$$

There are a great variety of researches related to the optimization model for improving the performance of GANs. Researches put forward conditional generative adversarial nets (CGANs) to make it meet the task of multi-variable generation, and the conditional probability is also introduced into the establishment of GANs model. As a consequence, the objective function is transformed from the accuracy rate to the conditional probability (Miyato and Koyama 2018). The detailed situation is shown as follows:

$$\max_{D} V(D,G) = E_{x\sim P_{data}}\left[logD(x|y)\right] + E_{x\sim P_G}\left[log(1-D(x|y))\right] \tag{18}$$

where y refers to the condition sequence of objective value. As a whole, CGANs is a game antagonism process with the noise concerning conditional probability. Besides, the researchers also proposed GANS models such as ForGAN, SeqGan, MAD-GANs to deal with the problem of time series data processing (Goodfellow 2016; Arjovsky and Bottou 2017; Zhou et al. 2018; Li et al. 2019).

In previous studies, generators and discriminators often used deep learning neural network models, such as DNN, CNN, RNN, LSTM, etc. The specific model selection depends on the characteristics of the research problem and the characteristics of the data set (Liu et al. 2016,2009; Lazar and N A. 2002; Husson and Josse 2014).

2.4 Probability distribution of economic data

Financial dynamics is an independent research field in financial analysis and economic analysis. By virtue of the description of volatility, yield, correlation and diffusion dynamics, it analyzes and investigates the internal dynamics of financial phenomena. Financial dynamics can deal with researches and application work in the combination with the traditional method of time series and machine learning. It has interdisciplinary research fields and similar goals with econometrics and other disciplines, but the methodology and technical paths are different (Honaker and King 2010).

In the related researches revolving around randomness, the description of economic data done by researchers include Pareto estimation, random estimation, mixed Gaussian distribution and Levy distribution. At present, it is generally accepted that the distribution of economic data conforms to the truncation of Levy flight distribution, which is expressed as:

$$\boldsymbol{P_L(x)} = \begin{cases} 0 & x > l \\ \boldsymbol{cP_L(x)} & -l \geq x \leq l \\ 0 & x < -l \end{cases} \tag{19}$$

where $P_L(x)$ is a symmetric Levy distribution, c is a normalization constant. This conclusion is helpful for the design and selection of training strategy and activation function in the construction of deep learning model.

The Levy flight censored distribution characteristics of economic data are unique, and this feature will affect the design of the algorithm and the selection of critical parameters. We must also carry out model research in conjunction with the distribution characteristics of the data when dealing with other types of data (such as physical data, meteorological data, medical data).

3 Model and method

3.1 Overall structure

The overall structure of the model is described in this part. The model adopts the generative adversarial nets, and the objective function partially adopts the objective function with conditional probability. In combination with the Levy ending distribution property of economic data and experimental test, the tanh function is regarded as the final output layer of conditional probability. The input of the whole model is the conditional vector matrix v. Since GAN training is very sensitive to dimensions, the dimension of the conditional vector matrix is set to 3. For the case where the original data is of high dimension, the automatic coding machine can be employed for dimensionality reduction, so as to achieve the goal of dimensionality reduction without causing data loss. Bidirectional LSTM model is selected as the trainer, which takes both long range memory and short range memory into account, and the streaming mode is employed to generate data. In addition, the discriminator adopts CNN model and autoregressive mode to judge the generator. In order to meet the requirements of CNN data input, the original data needs to be preprocessed and constructed three-dimensional matrix, which is connected and unified with the process of automatic encoder. In the preprocessing process, first, we calculate the volatility of key data indicators. The volatility rate is the change range of the sequence data relative to the previous sequence data. This indicator is calculated within the sequence data without large-scale fluctuations and can be combined with the data normalization preprocessing process. The overall structure is shown in Fig. 4.

3.2 Model of Bi-LSTM generator

The whole structure of the generator adopts bidirectional LSTM model. The data of the input conditional state matrix is the matrix sequence with dimension 3, and the input result matrix is the random generated dimension noise sequence with dimension 1. In order to ensure that the model adopted by the generator can guarantee the achievement of the generation goal, and in order to determine the selection of the hyper-function the model, the Bi-LSTM model is implemented preliminary study at first. The input data is firstly studied by pre-processing and preserving the volatility, and the value volatility rate is defined as:

$$\boldsymbol{R}(t) = \boldsymbol{lnP}(t+\Delta t) - \boldsymbol{lnP}(t) \tag{20}$$

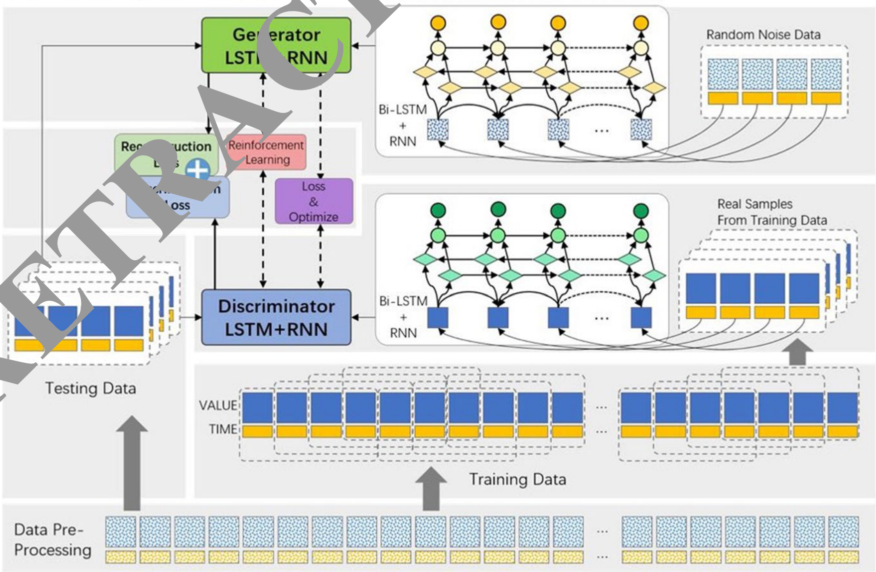

Fig. 4 The overall structure of research

Table 1 The algorithm of Bi-LSTM generator

Algorithm 1 Bi-LSTM generator algorithm

```
$gettimeseriesdata\{x\},\{v\}fromdatasource$
$datapreprocessing : reshapedimensionto3byAEsalgorithmas\{v^*\}$
$datapreprocessing : \{x'\},\{v'\}isnormalizedvolatilityrateof\{x\},\{v^*\}$
$createBiLSTMmodel, setlayers = 10, OptimizedbyAdam, LosssetbyMSE$
$Learningratesetbydecaylearningin[0.001 \sim 0.05]$
$initializeBiLSTMbyseed, settrain_{set}andtest_{set}fromdata$
$trainBiLSTMbytrain_{set}$
$testBiLSTMbytest_{set}$
$outputresult$
```

Table 2 The pre-training results of Bi-LSTM

Model	Result (MSE*100)
ARIMA	12.9153
ARIMA-GARCH	10.8762
SVM	9.8034
LSTM	8.0080
LSTM (in train_set)	7.5401
Bi-LSTM (epochs = 100)	7.5305
Bi-LSTM (epochs = 2000)	6.9332
Bi-LSTM (epochs = 2,in train_set)	6.7299

where $P(t)$ refers to the price condition at a certain moment, and Δt is equivalent to 1 in the continuous window-based analysis. Shanghai stock index serves as the data in preliminary study and according to the method of order selection of ARIMA model, the selection range of best order is 21–43. Generally, ARIMA order model does not exceed 6, which also shows the limitations of traditional models such as ARIMA, etc. On account that long-range memory exists in LSTM sequence, the length of sliding window is set to 20 in the preliminary study. In addition, the multi-layer neural network is set in the model, the loss function adopts MSE, the update strategy adopts Adam, the activation function of the output layer adopts tanh, and the activation function of other layers adopts softmax function. Attenuation learning rate is adopted and selected from the gradient [0.001–0.05]. The pseudo code of the algorithm process is shown as follows (Table 1).

In the preliminary training, better results were achieved at epoch = 2000, when MSE was reduced to less than 0.07. Compared with baseline models ARIMA, ARIMA-GARCH, SVM and unidirectional LSTM, the result is much better. Among them, the parameters of the unidirectional LSTM model are the same as those of Bi-LSTM. Whereas, the comparison of training and test results subjected to multiple parameters shows that the performance of the model in the training set is always better than that in the result set, which can be attributed to two reasons by means of preliminary analysis. Firstly, LSTM is a generation method of flow mode data, which focuses on the law of micro fluctuation, but cannot reflect the global fluctuation of data; secondly, when LSTM is trained, the training dataset of time series data cannot reflect the specific data laws of the data of test set. The first problem can be addressed with the GANs model and the second problem can be alleviated by the Bi-LSTM model. The results show that compared with the training dataset, the error of LSTM model in Bi-LSTM test decreases by 56.55% (Table 2).

The training situations are shown in Fig. 5. It can be seen from the training situations that, with the increase of epoch, the error decreased gradually. (d.1) and (d.2) indicate that the forward and reverse fitting results of the model are similar, which represents the effectiveness of Bi-LSTM in extracting the overall regularity of the sequence.

3.3 CNN discriminator model

In this study, CNN is used to construct discriminator. CNN network can generalize and extract global information by convolution of local information, which is essentially a structured multi-layer neural network. The construction of the discriminant model mainly depends on the following steps. The first step is to carry out data preliminary processing. The preliminary processing here is to convert the data after AEs dimensionality reduction from the matrix of t×3 to the matrix of 20×20×3, where t refers to the length of the time series. The converted data can be expressed

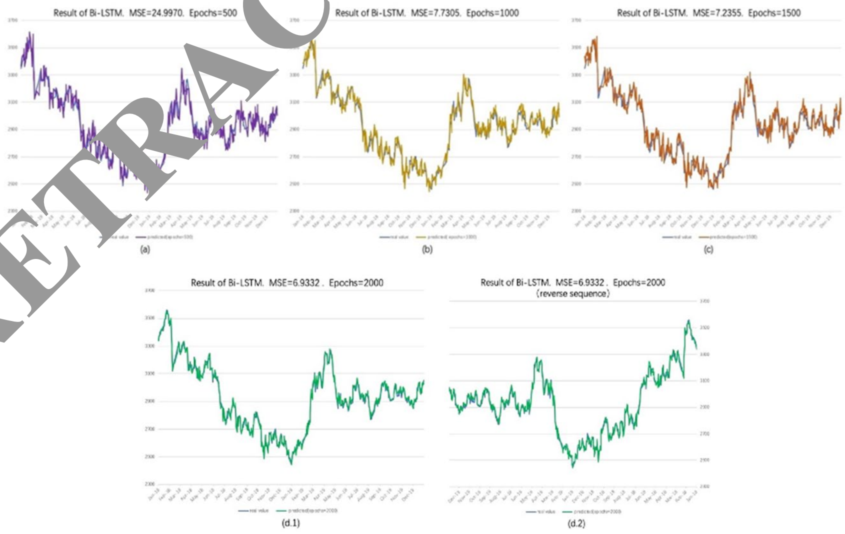

Fig. 5 The predicive simulation of Bi-LSTM

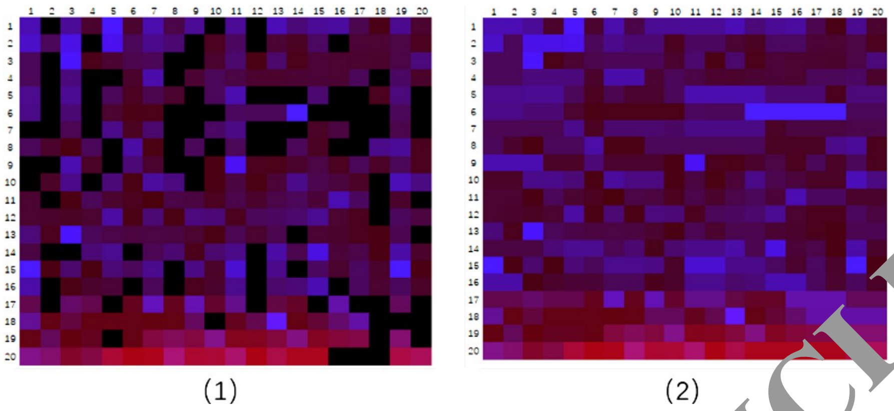

Fig. 6 Schematic diagram of data after matrix conversion preprocessing

Table 3 The algorithm of CNN discriminator model

Algorithm 2 CNN discriminator model algorithm	
1	$gettimeseriesdata\{x\},\{v\}fromdatasource$
2	$datapreprocessing : reshape\{x,v\}dimensionto3byAEsalgorithmas\{x'\}$
3	$datapreprocessing : reshape\{x'\}to20\times20\times3as\{img\}$
4	$createCNNmodel:setinput_{layers},conv_{layers},pooling_{layers},output_{layers}$
5	$OptimizedbyAdam,LossbyL1$
6	$initializeCNNbyseed, settrain_{set}andtest_{set}fromdata\ preprocessing$
7	$trainCNNbytrain_{set}$
8	$testCNNbytest_{set}$
9	$outputresult$

by images, and each image contains 20×20 pixel data. The second step is to mark the image, in which each image will be equipped with a corresponding label. The mark from the real dataset is 1, and the mark from the noise dataset is 0. The third step is the preliminary training of the image classification and in the fourth step, the model is applied to the generative adversarial training in the adversarial nets. After preprocessing, the data of the target sequence is converted into formatted data. We can use image processing for further research or use other data analysis methods for further research. In the process of data conversion, there is no loss in the data's own change rules and data distribution.The image figure that has been preprocessed by data is shown in the following figure, in which the left image is the case when data is missing, and the right image is the case when data completion is finished by virtue preprocessing (Fig. 6).

The model structure is set to the input layer. There are four 3×3 convolvers, and a pooling layer is prepared after each two convolvers. Finally, the data is streamed to the output layer and the output result is processed by softmax. There

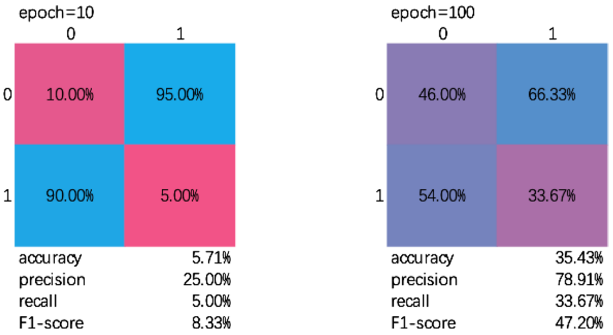

Fig. 7 CNN verification test (1)

are two reasons why the model does not take the usage of a larger convolver into account. On the one hand, the use of small kernel convolver is expected to enhance the overall training efficiency of the model; on the other hand, the order of long-range influence of economic data is about 20, which is exactly the range of data induction corresponding to the 3 × 3 kernel convolver. Compared with the 3 × 3 kernel convolver, the convolver with larger kernel may acquire worse performance in the extraction of long-range correlation law (Table 3 and Fig. 7).

The model performs well in the preliminary training of datasets in which there are 300 pictures for the data of commercial price and 50 pictures among them with noise data. The training results are shown in the following figures. It can be seen that the accuracy of 98.94% and F1 score of 0.96 can be achieved after the training of 500 cycles (Fig. 8).

In order to evaluate and compare the performance and results of the model, this paper uses a Binary classification machine to analyze the model. In the verification data set, the data is divided into two categories, labeled 0 and 1.

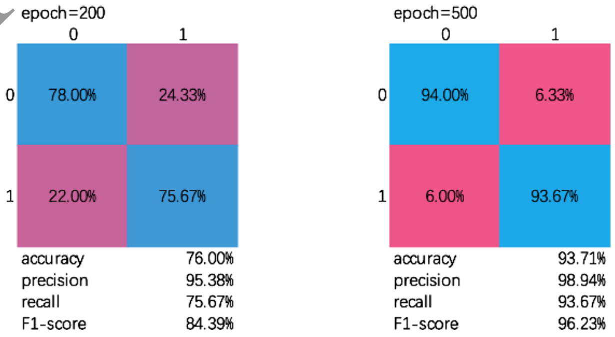

Fig. 8 CNN verification test (2)

3.4 Automatic coder AEs

In this study, the automatic coder is employed to higher and lower the dimension of multi-dimensional data. Automatic coders are a branch of unsupervised learning, and the core architecture is composed of an encoder and a decoder. By means of the encoder, the features of the input data are extracted by virtue of the multi-layer neural network to form the feature vector. Then the feature vector can be returned to the raw dimensional data by means of the decoder-based reverse change. The model uses the error between input and output as the objective function for training, and the final purpose is to make the output signal as similar as possible to the input single, that is, the information contained in the raw data is not lost during the mapping process of the feature vector. The coding target structure of the automatic coder is 400 × 3, which is determined by the characteristics of the dataset itself and the requirements of the model construction. The output structure of the decoder is the same as that of the raw data. The mathematical expression of the encoding and decoding processes is respectively shown as follows:

$$\boldsymbol{encode} : \boldsymbol{y} = \boldsymbol{s}(\boldsymbol{Wx} + \boldsymbol{b}) \tag{21}$$

$$\boldsymbol{decode} : z = \boldsymbol{s}\left(\boldsymbol{W}^{T}\boldsymbol{y} + \boldsymbol{b}^{T}\right) \tag{22}$$

where the data matrix input as x is output as the matrix y through the multi-layer neural network. s represents the nonlinear activation function. In this study, tanh activation function is adopted, W is the weight matrix and b is the bias quantity of the neural network. During the process of decoding, W^T and b^T are the transposed matrixes of W and b, respectively.

The dimensionality raising and reduction of the raw data to feature vectors and the restoration of the feature vector to the raw data are achieved by means of encoder and decoder, respectively. Automatic coder is a mature machine learning technology with stable performance in terms of coding and decoding. Within the range of error acceptance, it is generally believed that the output z of the decoder is exactly the same as the input x of the encoder. Also, such a process can be employed to implement the dimensionality reduction as well as the dimensionality raising of the input data.

3.5 Conditional generative adversarial nets (CGANs) model

Compared with the previous GANS in which the unconditional constrain generation was used, the input of D and G in cGANs was granted with conditions, so that the generated model can describe the probability distribution under different kinds of conditions, so as to achieve a better effect. Each component of the model has a clear division of labor. The generated model G is used to extract the distribution of data, and the discriminant model D is used to determine the probability that an input sample comes from real data rather than the generated sample. For the convenience of the following discussion, the input of the real dataset is recorded as x and the input of the noise dataset as z. The objective function of the model is expressed as follows:

$$\underset{G}{\boldsymbol{min}}\,\underset{D}{\max}\,\boldsymbol{V}(\boldsymbol{D},\boldsymbol{G}) = \boldsymbol{E}_{x\sim P_{data}}\left[\boldsymbol{logD}(\boldsymbol{x}|\boldsymbol{y})\right] + \boldsymbol{E}_{z\sim P_G}\left[\boldsymbol{log}(1-\boldsymbol{D}(z|\boldsymbol{y}))\right] \tag{23}$$

The optimization procedure of discriminator D can be expressed as:

$$\underset{D}{\max}\,\boldsymbol{V}(\boldsymbol{D},\boldsymbol{G}) = \boldsymbol{E}_{x\sim P_{data}}\left[\boldsymbol{logD}(\boldsymbol{x}|\boldsymbol{y})\right] + \boldsymbol{E}_{z\sim P_G}\left[\boldsymbol{log}(1-\boldsymbol{D}(z|\boldsymbol{y}))\right] \tag{24}$$

The optimization procedure of generator G can be expressed as:

$$\underset{G}{\min}\,\boldsymbol{V}(\boldsymbol{D},\boldsymbol{G}) = \boldsymbol{E}_{z\sim P_G}\left[\boldsymbol{log}(1-\boldsymbol{D}(z|\boldsymbol{y}))\right] \tag{25}$$

In the function *V*(*D*, *G*), the first term is the entropy of the data from real distribution *pdata*(*x*) through discriminator and the discriminator tries to make it maximized up to 1; the second term is the entropy of the data from random input *p*(*z*) through generator. A false sample was generated by the generator and it is recognized by the discriminator. As for this term, the discriminator tries to make it minimized to 0. In general, the discriminator tends to maximize the function *V*(*D*, *G*), but the generator tends to minimize the function *V*(*D*, *G*), so that the disparity between the real data and false data can be narrow to a great extent. When the classification accuracy of discriminator D converges to 50%, this practice indicates that the generator G has made discriminator D unable to distinguish data sources, which can be regarded as the completion of the model training. At the moment, generator G can be employed to generate new data or perform prediction tasks.

cGANs is the core link of the whole research. In terms of the selection of hyper-function, the model parameters are optimized in this research in combination with the improvement strategy of WGAN and the characteristics of economic data. As

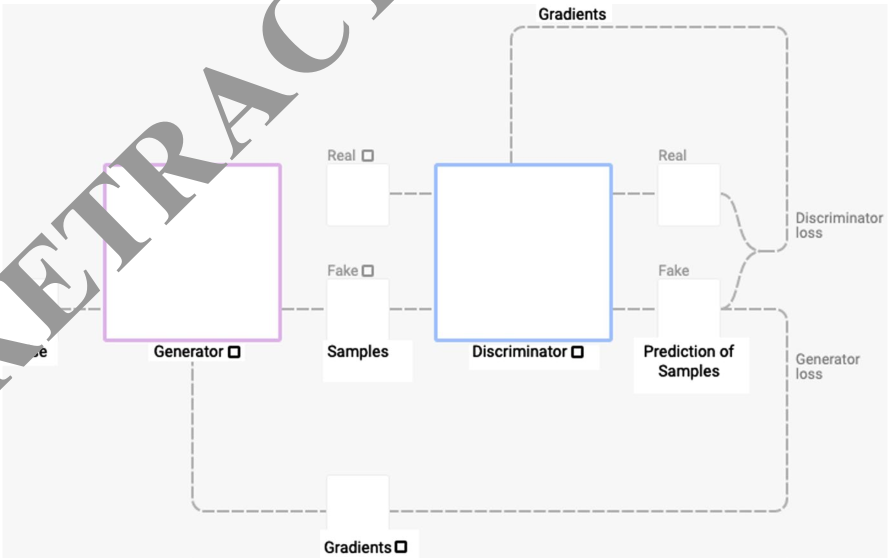

Fig. 9 Framework of Cgan

Table 4 The algorithm of cGAN model

Algorithm 3 cGANs model algorithm

```
$gettimeseriesdata\{x\}, \{z\}, \{v\}fromdatasource$
$datapreproces\sin g : reshape\{x, v\}to400 \times 3byAEsa\lg orithmas\{x'\}$
$datapreproces\sin g : reshape\{v\}to\dim ension3byAEsa\lg orithmas\{v^*\}$
$datapreproces\sin g : reshape\{x'\}to20 \times 20 \times 3as\{img_{real}\}$
$datapreproces\sin g : \{z'\}, \{v'\}isnormalizedvolatilityrateof\{x\}, \{v^*\}$
$createcGANswithgenerater = a\lg orithm1asG, discri\min ator = a\lg orithm$
$createcGANswithdiscriminator = algorithm2asD, OptimizedbyRSMProp$
$repeatcGANs.trainuntilD_{rate} = 50\%orepoch = epoch_{input} :$
$repeatcGANs.traincGANs.G.trainandoutputgeneratedata\{g\}, \{v\}$
$repeatcGANs.trainreshape\{g, v\}to400 \times 3byAEsa\lg orithmas\{g'\}$
$repeatcGANs.trainreshape\{g'\}to20 \times 20 \times 3as\{img_{fake}\}$
$repeatcGANs.traincGANs.D.trainby\{img_{real}\}$
$repeatcGANs.trainD_{rate} = cGANs.D.testby\{img_{real}, img_{fake}\}$
$repeatcGANs.traincontinuetooptimizecGANs.D, cGANs.GandcGANs$
$outputresult$
```

for the activation function of the last layer, the tanh function close to Levy truncated distribution is adopted and the optimization algorithm is replaced by the RSMProp algorithm. The expression of the algorithm of CGANs model is shown as follows (Fig. 9 and Table 4).

4 Datasets and experiments

Two datasets were used in this study, one of which was taken from the data associated with the stock index and stock price of Shanghai stock exchange, and the other one is related to the data of commercial prices of a city Two types of data are introduced below.

4.1 Dataset of stock index and stock price

The data comes from the Shanghai stock exchange and is publicly available. The dataset contains more than 7000 trading days of the Shanghai composite index and stocks since 1990. The data fields include closing price, highest price, lowest price, opening price, previous closing price, up/down amount, up/down amount, volume and transaction amount. In the actual time series analysis, it is necessary to conduct correlation analysis or causative analysis on the data of each field in order to determine the impact of each index on the results. The commonly used methods include principal component analysis (PCA), empirical mode decomposition (EDM) and XGBoost algorithm in machine learning. In this study, two indexes including up and down amount and up and down range were eliminated, and the other seven indexes are input as state data. Data for a total of 400 trading days from August 2001 to April 2003 were selected, and a total of 50 indexes and stocks including the Shanghai stock index were used as research data.

4.2 Date set of commercial prices

This data is derived from the commercial price in a city and contains a large amount of price data. The available data indicators include transaction price, transaction quantity and total transaction volume, which are all calculated and stored on a daily basis. In this study, the price fluctuation of 50 items within 400 consecutive days was selected for data analysis and model research in this paper.

4.3 Evaluation function

In this study, MSE pair is used as the evaluation index to evaluate the final results of the generator. The evaluation process includes the value of the given sequence length, the random starting position, and the prediction of the next sequence position. This evaluation is for the overall output of the entire model. In the process of model training, the training status of discriminator can be monitored by classification accuracy F1-score, and the overall training status can be monitored by loss function, JS divergence, KL divergence, etc. It is necessary to monitor the situation during the training process. Compared with other deep learning models, it is more likely for the training process of GANs to encounter difficulties. On account that the initial state is random, the random initial state is conducive to the training of the model and the development of the training towards convergence. Whereas, if the random initial state and the initial training are far away from the real sample, it will be difficult to carry out the training process. At the moment, the parameters can be reset to implement retraining.

4.4 Experiment and model training

Model training and data experiment are carried out for the design and dataset of the above mentioned model. In the process of model training, the effect of initial noise on model training was studied. Figure (a) shows that the training process encountered a long period of oscillation, it can be seen that the oscillation continued at epoch = 2000. Figure (b) shows the deviation from the real sample caused by the noise data encountered in the training, which has not entered the convergence trend after the training to epoch = 3000. Figure (c) shows a better training process, which entered the convergence trend around epoch = 1200. During the training process, the model learning rate was reduced at epoch = 2050, and then the model gradually converged and produced good results. Many researchers have proposed solutions to address the difficulties of GANs training. The improvement of the preliminary training process for the generator and the construction of mixed noise data in the combination of actual data and noise data are effective solutions.

For the difficulties encountered in the training, the parameters were studied and optimized in the experiment. During the training process, on account that the output of GANs is processed by tanh, the distribution of loss function tends to the same trend at the edge. If the edge of big error is reached during the distributed training, it is difficult to jump out of the big error interval in the case of low learning rate. As a consequence, the situation shown in figure (a) is presented. Similarly, the training shown in figure (b) also demonstrates such a problem. In addition, the training shown in figure (b) also reveals another problem, that is, since the overall objective function reflects the optimal game of G and D, when D continues to be optimized, G develops backward due to the random deviation of noise, and thus develops the process into a training task that refuses to converge. For the above two problems, in the subsequent training, the input noise data is processed into mixed noise data. At the same time, the method of attenuating learning rate was adopted to accelerate the convergence rate of training, and a good effect was obtained, which is shown in figure (c) (Fig. 10).

Several baseline comparison models and change models were set up in the study. The baseline model includes the ARIMA-GARCH model representing the traditional time series analysis, excellent algorithm of machine learning SVM, sequential deep learning method LSTM and ordinary GANs. In the GANs model, the discriminator and generator are both set as the deep neural network with 5 layers and 64 cores, and MSE is selected as the index of comparative evaluation.

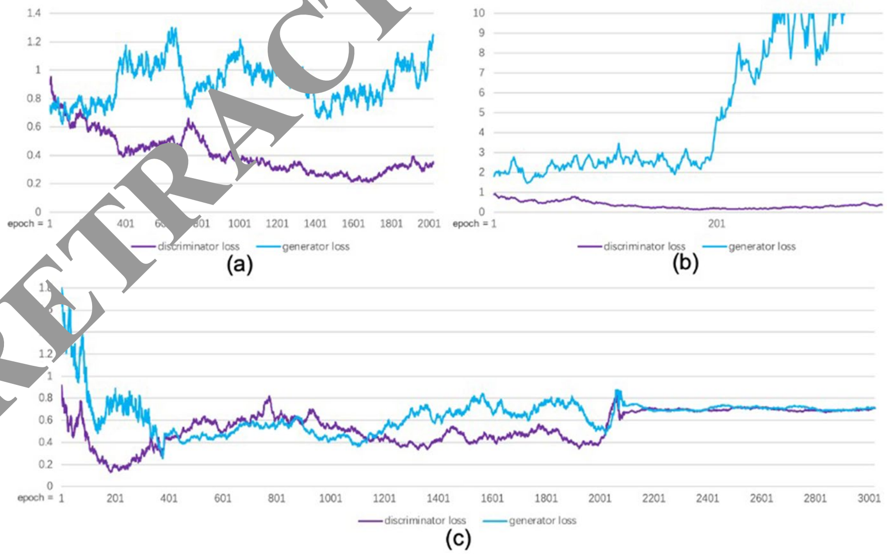

Fig. 10 The loss funtion in the training of GANs

5 Results

The index to evaluate the experimental results includes MSE, RMSE, MAE and MAPE. The formula of the four indexes is shown as follows:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(\hat{y}_i - y_i\right)^2 \tag{26}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\hat{y}_i - y_i\right)^2} \tag{27}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|\hat{y}_i - y_i\right| \tag{28}$$

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(\hat{y}_i - y_i\right)^2 \tag{29}$$

The value range of MSE, RMSE and MAE is $[0, +\infty)$. When the predicted value and the true value reach the ideal fitting, it is equal to 0, that is, the perfect model. In addition, the greater the error, the greater the value. The value range of MAPE is $[0, +\infty)$, in which 0% represents an ideal model, while MAPE higher than 100% represents a poor model. The evaluation target of MSE and RMSE is the predicted volatility after normalization, and the MAE and MAPE evaluation indexes are the predicted values.

The datasets were trained and tested on Stock price Dataset (Dataset I) and Commodity price Dataset (Dataset II). A representative result is selected for each of the two datasets, whose situation can be sorted in the table below. For dataset I, compared with the baseline model, the CGANs model achieved better results in all indexes. Among them, the MSE was decresed to close to 0.05, which indicates that the performance is increased by 48.75% compared with the traditional method ARIMA-GARCH. For another index MAE, CGANs was reduced to 20% of the error of the ARIMA-GARCH model, and the optimization range was higher than that in MSE. This phenomenon can be attributed to the volatility of the dataset, which makes the optimization range of MSE and MAE differ by about 4 times. In fact, it is related to the need for ARIMA to complete the difference to convert the data into a stationary sequence. The comparison results show that the neural network model (LSTM, GAN, cGANs) has better processing effect on volatility error than SVM and ARIMA-GARCH. According to the results of dataset II, the optimization amplitude is better than the dataset. In terms of the stationarity of dataset I and dataset II, the comparison shows that dataset I needs 6 differences to show stationarity, while dataset II needs 3 differences to show stationarity. This also proves that the stability and volatility of the dataset have impacts on the effect of the model and at the same time, the deep learning method has a strong ability to deal with non-stationary sequences.

Table 5 Comparison of prediction results of dataset I-based model

Data set	Model	MAE	MAPE	MSE	RMSE
Stock price Dataset	arima-garch	35.4479	2.2179%	0.1014	0.3184
	svm	19.9602	1.2488%	0.0953	0.3087
	lstm	10.6248	0.6634%	0.0769	0.2773
	gan	9.2477	0.5777%	0.0720	0.2684
	cgans	7.9081	0.4939%	0.0519	0.2279

Table 6 Comparsion of prediction results of dataset II-based model

Data set	Model	MAE	MAPE	MSE	RMSE
Commodity price dataset	arima-garch	27.0876	1.69%	0.0930	0.3050
	svm	15.5014	0.96%	0.0796	0.2821
	lstm	10.5897	0.66%	0.0766	0.2767
	gans	9.2449	0.57%	0.0831	0.2882
	cgans	7.8425	0.48%	0.0598	0.2444

The comparison of LSTM model pre-training in the third part shows that cGANs has better effect than Bi-LSTM, and the effect of test set and training set is relatively uniform. This result reveals that the introduction of the confrontation between the autoregressive model and the flow model will give consideration to the law of the overall sequence and the law of the microscopic sequence, thus making the generalization of the law more complete.

Making comparison of the effects of GANs and cGANs, it can be seen that the difference between both of them is reflected in two aspects. The one lies in the introduction of condition probability in cGANS and on the other hand, data preprocessing and targeted model construction for cGANs are added in this research. In addition, it can be seen from the results that the performance of GANs without optimization is basically equivalent to that of LSTM, and the performance demonstrated by some indexes is weaker than that of LSTM model (Tables 5, 6).

The following figures show the comparison of 100 data processing tasks in two datasets, in which the horizontal label represents the sample number and the vertical coordinate represents the result value of RMSE. It can be observed from the index image that the performance of ARIMA-GARCH and SVM is relatively ordinary, the performance of LSTM and GAN models is distributed in the middle range, while the performance of cGANs is generally better than that of various models and performs better in dataset I. The image distribution of dataset II shows that the performance of SVM is significantly improved in the data approaching to the stationary sequence, and the difference between each method is smaller than the comparison difference in dataset I. Such a consequence once again reveals that the model developed in this paper is relatively effective in dealing with non-stationarity.

What should be reported is that the training time cost by cGANs is much longer than that by other methods (Fig. 11).

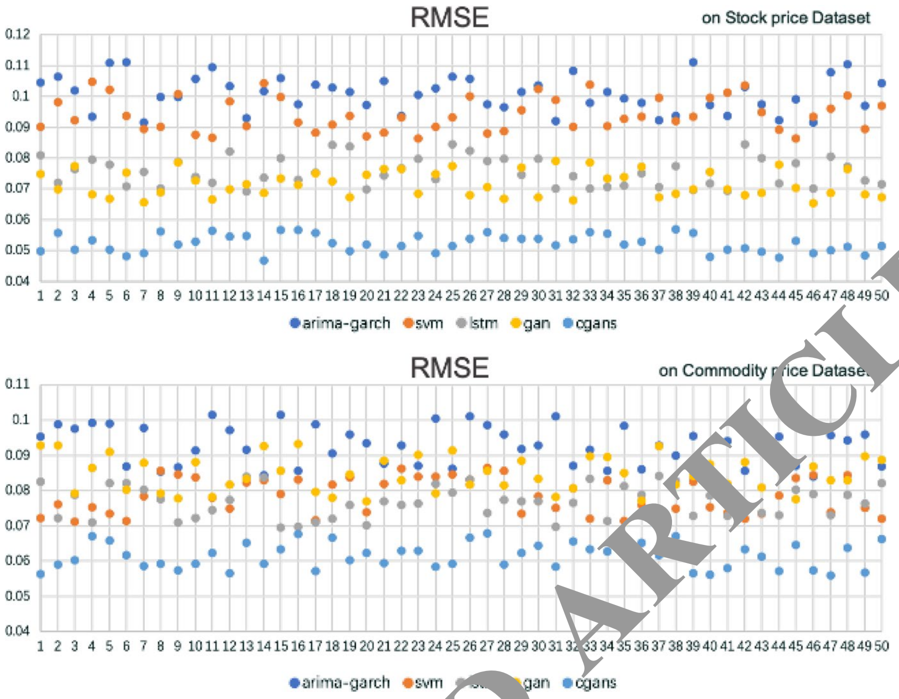

Fig. 11 RMSE results of 100 samples

6 Conclusion and expectation

In this paper, the author introduces the background information in terms of the time series processing of economic data, and puts forward tasks to predict the economic time series data based on cGANs model. Revolving around such a cGANs model, Bi-LSTM generator and CNN discriminator are constructed, and the preliminary processing of automatic coder AEs and sequential data visualization are added in combination with the preliminary analysis of data and existing research achievements. The comparison results indicate that these strategies can reap desirable effects. The performance of cGANs model is increased by 48.75% compared with traditional methods. In the part of model and experimental analysis, the performance and parameters of the generator and discriminator are discussed, and the training strategy and parameter selection of overall cGANs are analyzed, reported and discussed. In the part of result analysis, the comparison shows that the effect of the model is better than the baseline model and at the same time, its processing capacity in dealing with non-stationary series is superior to traditional methods and machine learning method. Such two conclusions can be regarded as the major points contributed by this research.

Considering the limitation caused by computing resource, no more sample data is introduced in the research. Whereas, the results acquired from studying the two kinds of economic data including stock data and commercial prices show that the method developed in this paper performs well in handling with the analysis of economy-related data. In addition to economic data, the research method of this paper can also be applied to data in other fields, such as physical sensor data, medical monitoring data, and other types of time series data. Besides, the method proposed in this paper can mine and learn the overall distribution of data under specific conditions, so this method can also be applied to further research and data tasks such as data missing filling and noise data recognition.

In this paper, when using the LSTM model, the LSTM can be used to describe this feature of the overall distribution of sequence data. This is different from the usual prediction tasks using LSTM, which is also a critical innovation point in this study. When LSTM is used to mine the overall distribution of data, there is a problem of fragmented data perspective. Setting a global deep learning model in the GAN model for adversarial training can enhance LSTM's ability to describe global data distribution rules.

In general, the issue that this research faces lies in the processing of single sequence data, whose method can be enlarged to the processing of multi-sequence data. The problem that the training time cost by cGANs was relatively long was encountered in this research. Although the convergence effect can be improved by virtue of adding strategies, the problem related to long training time shall be further studied and addressed.

References

Arjovsky M, Bottou L (2017) Towards principled methods for training generative adversarial networks

Blackwell M, Honaker J, King G (2017) A Unified approach to measurement error and missing data: details and extensions. Soc Methods Res 46(7):1–9

Che Z, Purushotham S, Cho K et al (2018) Recurrent neural networks for multivariate time series with missing values. Sci Rep 8(1):1

Devlin J, Chang M-W, Lee K, Toutanova K (2008) BERT: pre-training of deep bidirectional transformer for language understanding. arXiv preprint arxiv: 1810.04805

Feng Z, Hao-Min Z, Zhihua Y et al (2018) EMD2FNN: a strategy combining empirical mode decomposition and factorization machine based neural network for stock market trend prediction[J]. Expert Syst Appl 115:136–151

Goodfellow I (2016) NIPS 2016 Tutorial: generative adversarial networks

Honaker J, King G (2010) What to do about missing values in time- series cross-section data. Am J Political Sci 54(2):561–581

Hron K, Templ M, Filzmoser P (2011) Imputation of missing values for compositional data using classical and robust methods. Comput Stat Data Anal 54(12):3095–3107

Husson F, Josse J (2014) missMDA: Handling missing values with/in multivariate data analysis (principal component methods)[J]

Lazar NA (2002) Statistical analysis with missing data. Technometrics 45(4):364–365

Li D, Chen D, Shi L, et al (2019) MAD-GAN: Multivariate anomaly detection for time series data with generative adversarial networks[M]

Liang Y, Wu J, Ma Y (2017) A new approach to estimate the missing value based on neighbor granular domain in the incomplete information system. IEEE International Conference on Software Engineering and Service Science. IEEE, 813–816

Liu J, Musialski P, Wonka P, et al. (2009) Tensor completion for estimating missing values in visual data. In: IEEE, international conference on computer vision. IEEE, 2114–2121

Liu ZG, Pan Q, Dezert J et al (2016) Adaptive imputation of missing values for incomplete pattern classification. Pattern Recogn 52(C):85–95

Miyato T, Koyama M (2018) cGANs with Projection Discriminator [J]. arXiv:1802.05637. https://arxiv.org/abs/1802.05637

Newman DA (2014) Missing data: five practical guidelines. Organ Res Methods Organ Res Methods 17(4):372–411

Powney M, Williamson P, Kirkham J et al (2014) A review of the handling of missing longitudinal outcome data in clinical trials. Trials 15(1):237

Sezer OB, Ozbayoglu AM (2019) Financial trading model with stock bar chart image time series with deep convolutional neural networks [J]

Stekhoven DJ, Bühlmann P (2012) MissForest–non-parametric missing value imputation for mixed-type data. Bioinformatics 28(1):112

Twisk J, De BM, De VW et al (2013) Multiple imputation of missing values was not necessary before performing a longitudinal mixed-model analysis. J Clin Epidemiol 66(9):1022–1028

Wei R, Wang J, Su M et al (2018) Missing value imputation approach for mass spectrometry-based metabolomics data. Sci Rep 8(1):1–10

Wu X (2002) Learning missing values from summary constraints. ACM SIGKDD Explor Newsl 4(1):21–30

Zhou X, Pan Z, Guyu Hu, Tang S, Zhao C (2018) Stock market prediction on high-frequency data using generative adversarial nets. Math Probl Eng 2018:1–11

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