Hindawi
Complexity
Volume 2023, Article ID 9369888, 11 pages
https://doi.org/10.1155/2023/9369888

Research Article

Prediction for the Inventory Management Chaotic Complexity System Based on the Deep Neural Network Algorithm

Tengfei Lei,[1] **Rita Yi Man Li**,[2] **Nuttapong Jotikastira**,[1] **Haiyan Fu**,[3] **and Cong Wang**[3]

[1]*Rattanakosin International College of Creative Entrepreneurship, Rajamangala University of Technology Rattanakosin, Nakhon Pathom, Thailand*
[2]*Sustainable Real Estate Research Center, Department of Economics and Finance, Hong Kong Shue Yan University, North Point, Hong Kong*
[3]*Collaborative Innovation Center of Memristive Computing Application, Qilu Institute of Technology, Shandong, Jinan 250200, China*

Correspondence should be addressed to Haiyan Fu; fuhy413@126.com

Received 6 February 2022; Revised 15 July 2022; Accepted 12 September 2022; Published 12 May 2023

Academic Editor: Chun-Biao Li

Precise inventory prediction is the key to goods inventory and safety management. Accurate inventory prediction improves enterprises' production efficiency. It is also essential to control costs and optimize the supply chain's performance. Nevertheless, the complex inventory data are often chaotic and nonlinear; high data complexity raises the accuracy prediction difficulty. This study simulated inventory records by using the dynamics inventory management system. Four deep neural network models trained the data: short-term memory neural network (LSTM), convolutional neural network-long short-term memory (CNN-LSTM), bidirectional long short-term memory neural network (Bi-LSTM), and deep long-short-term memory neural network (DLSTM). Evaluating the models' performance based on RMSE, MSE, and MAE, bi-LSTM achieved the highest prediction accuracy with the least square error of 0.14%. The results concluded that the complexity of the model was not directly related to the prediction performance. By contrasting several methods of chaotic nonlinear inventory data and neural network dynamics prediction, this study contributed to the academia. The research results provided useful advice for companies' planned production and inventory officers when they plan for product inventory and minimize the risk of mishaps brought on by excess inventories in warehouses.

1. Introduction

Researchers have found chaos in physics, chemistry, ecology, geography, and economics data [1], and the discrete nonlinear management system has been widely studied by many researchers [2–8]. The concept of chaotic strategic management dates back to 1983. In 1994, Feichtinger [4] studied chaotic planning, queuing, and scheduling in management operations. Murphy [5] used chaos to study public relations' problems and crises. After reviewing the chaos management research, Joseph [6] pointed out that chaos management requires a change in rules and adaptability [1].

The main purpose of inventory is to meet the demand, so demand forecasting is the basic premise of inventory management. Boardman and others used a clustering algorithm to compare new and existing similar products and predict sales volume of new products [9]. Van der Auweraer et al. utilized auxiliary installed base data to predict the spare parts demand [10]. Yu et al. proposed a support vector machine (SVM) model to predict the newspaper demand of different stores by including 32 features in the model [11]. Shimmura and Takenaka used the SVM method to forecast the demand for convenience store inventory data by reducing the feature dimension and data quantity [12]. Tanizaki et al. used POS, Bayesian linear regression, and other methods to predict hotel passenger flow [13].

In the era of big data, the cost of acquiring, storing, and processing a large amount of data is significantly reduced. Decision makers can observe historical demand

and acquire data such as weather, prices, holidays, promotion information, and demographic information to improve demand forecasting accuracy [14, 15]. In recent years, the advantages of machine learning in processing large datasets and high-dimensional feature data have attracted the attention of scientists. The rapid increase in data changes the prediction algorithm from traditional forecasting approaches to deep learning [16–31]. For example, Kong et al. used the restricted Boltzmann machine (RBM) algorithm based on deep learning to predict traffic flow. The phase space reconstruction of the RBM algorithm constructed the polymorphic long-term model of chaotic time series [17]. Wei and Wang proposed an anomaly detection method of hierarchical spatiotemporal feature learning network based on deep learning [18]. Zhang et al. used the residual neural network framework to model time proximity, period, and trend characteristics of crowd flow [19]. Haq et al. [29] utilized the multilayer bidirectional LSTM algorithm to identify the mitochondrial protein of the Plasmodium falciparum parasite. Khan et al. [30] used deep learning algorithms to predict residential and commercial energy consumption. Azar and Vaidyanathan [1] used a new deep learning algorithm to predict and analyze renewable energy power generation. However, as a typical nonlinear system, the complex inventory management presented a chaotic and nonlinear phenomenon with high complexity and small amplitude change during the time series change. It is impossible to make accurate predictions by using traditional machine learning. Thus, finding a suitable deep learning algorithm for prediction is necessary. Having said that, however, the above mentioned deep learning algorithm can also be used in other chaotic systems [32–35].

This paper aims to:

(1) Analyze the nonlinear characteristics of inventory management using the nonlinear dynamics theory;

(2) Verify the inventory data characteristics and forecast the inventory by using LSTM, bi-LSTM, and DLSTM algorithms.

This paper predicted inventory data under complex, chaotic systems. The prediction results concluded that the bi-LSTM algorithm is better for chaotic nonlinear datasets and provided a reference for other chaotic datasets. The rest of this paper is organized as follows: in Section 2, the chaotic inventory management system, the inventory data, and the data irregularity are nonlinear y 0-1 test. Section 3 introduces prediction models: LSTM, bi-LSTM, CNN-LSTM, and DLSTM. Section 4 verifies the abovementioned algorithms by experiments, and the optimal model is obtained by comparing three indexes. Finally, the results are summarised in Section 5.

2. Inventory Management Systems and Datasets

2.1. Inventory Management Model. Many enterprises face inventory peoblems whih can be represented in form of complicated chaotic systems of equations as follows [36]:

$$\begin{cases} x_{i+1} = s + pz_{i+1}, \\ y_{i+1} = qx_{i+1} + ry_i z_i, \\ z_{i+1} = 1 - x_i - y_i + z_i, \end{cases} \tag{1}$$

where s, p, q, and r are the system parameters, s represents the initial sales base, p represents the inventory fund transfer rate, q represents the product resource rate, and r represents the inventory efficiency. x_i represents the resources for sales in period i, y_i represents the number of customers in period i, and z_i represents the inventory capital of the company in period i. Normalizing the parameters of the inventory management model [36], the results would be: $0 < x_i < 10 < y_i < 1$ and $0 < z_i < 1/r$. Where $p = 0.43$, $q = 0.38$, $s = 0.11$, and $r = 0.72$. The attractors of a system (1) are shown in Figure 1.

2.2. 0-1 Test. This study implemented the 0-1 test to investigate whether the data is chaotic. He et al. used 0-1 test algorithm to make correlation analysis on the time series of fractional order system [8]. If $\varphi(n)$ $(n = 1, 2, 3, \ldots)$ represents a one-dimensional observable iterative data, then the two real-valued functions would be [36]:

$$\begin{cases} p(n) = \sum_{i=1}^{n} \varphi(i) \cos(\theta(i)), \\ s(n) = \sum_{i=1}^{n} \varphi(i) \sin(\theta(i)), \end{cases} \tag{2}$$

where $\theta(i) = i\omega + \sum_{j=1}^{i} \varphi(j)$, the trajectories are visualised in Figure 2. If the bounded trajectory in the Figure 2 is a regular cloud shape, then the unbounded trajectory follows Brownian motion and the data is chaotic. This method was used to study the y and z sequence of the system (1). Its parameters were the same as those in Figure 2. The p-s relationship is displayed in Figure 3. The change of inventory safety threshold due to the change in stocks of goods with time is irregular, which cannot be accurately predicted by traditional algorithms [36].

3. Research Method

3.1. LSTM Model. LSTM network improves RNN. RNN neurons are shown in Figure 3. Cell memory unit structure is added to the hidden layer of RNN, which allows the model to learn the information for a long time and effectively overcome the problem of gradient disappearance or explosion [29]. LSTM introduces a memory cell structure in the hidden layer, including three gate controllers: input, forgetting, and output gates [37], allowing the network to forget historical information and update the memory state with new information. The structural diagram of LSTM neurons is shown in Figure 4.

The three gates adopt the sigmoid function, and all of them are nonlinear summation units. At the same time, the activation functions inside and outside the module are included. The multiplication operation is used to control the

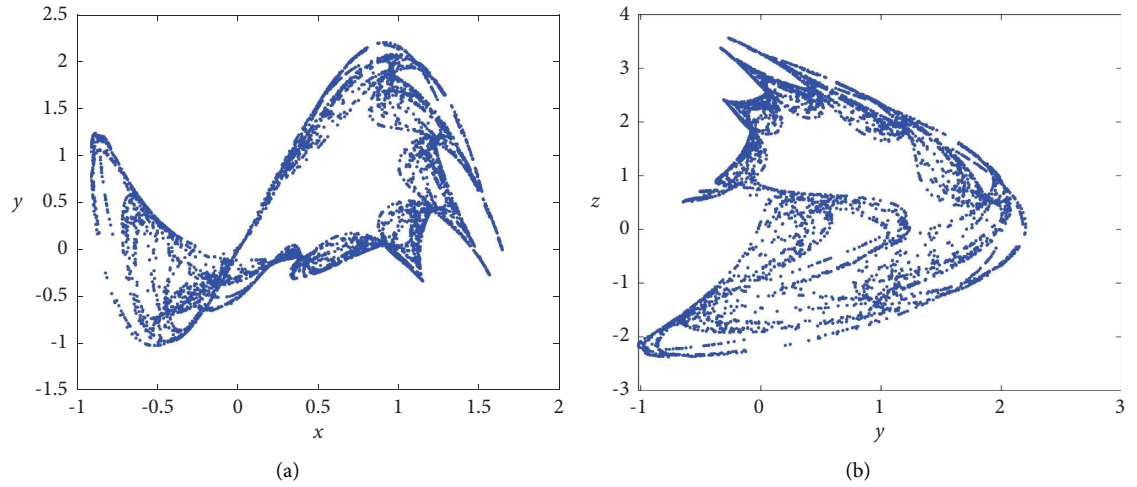

Figure 1: Attractors of system (1) with $p = 0.43$, $q = 0.38$, $s = 0.11$, and $r = 0.72$. (a) x-y phase. (b) y-z phase.

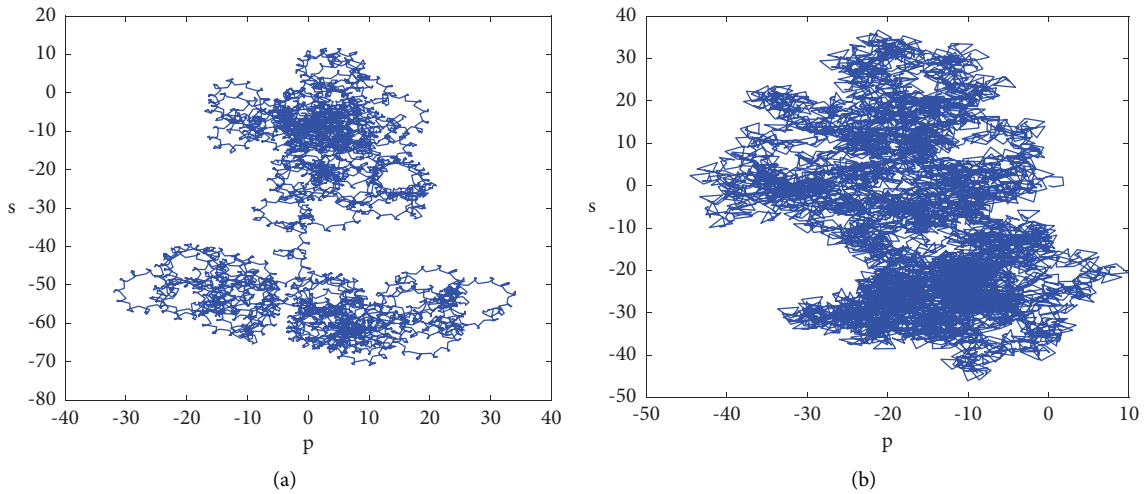

Figure 2: p-s phase diagrams of inventory system (1). (a) p-s plot of y sequence. (b) p-s plot of z sequence.

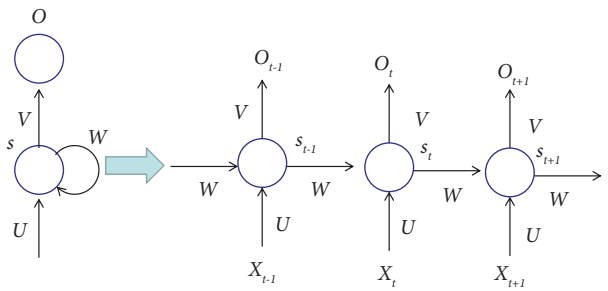

Figure 3: Framework of RNN.

activation functions of the units. The calculation consists of the following steps:

We calculate the value f_t of the forgotten gate as follows:

$$f_t = \sigma\left(W_f \cdot \left[h_{t-1}, x_t\right] + b_f\right). \tag{3}$$

We calculate the value of the input gate as follows:

$$\begin{aligned} i_t &= \sigma\left(W_i \cdot \left[h_{t-1}, x_t\right] + b_i\right), \\ \tilde{c}_t &= \tan h\left(W_C \cdot \left[h_{t-1}, x_t\right] + b_C\right). \end{aligned} \tag{4}$$

We calculate the current time memory unit state value C_t as follows:

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t. \tag{5}$$

We calculate the output gate and memory output h_t of the LSTM unit as follows:

$$\begin{aligned} o_t &= \sigma\left(W_o\left[h_{t-1}, x_t\right] + b_0\right), \\ h_t &= o_t * \tan h\left(C_t\right). \end{aligned} \tag{6}$$

LSTM and RNN speculate backward data through forwarding information. Forward and backward information is used to predict the current time, strengthening the

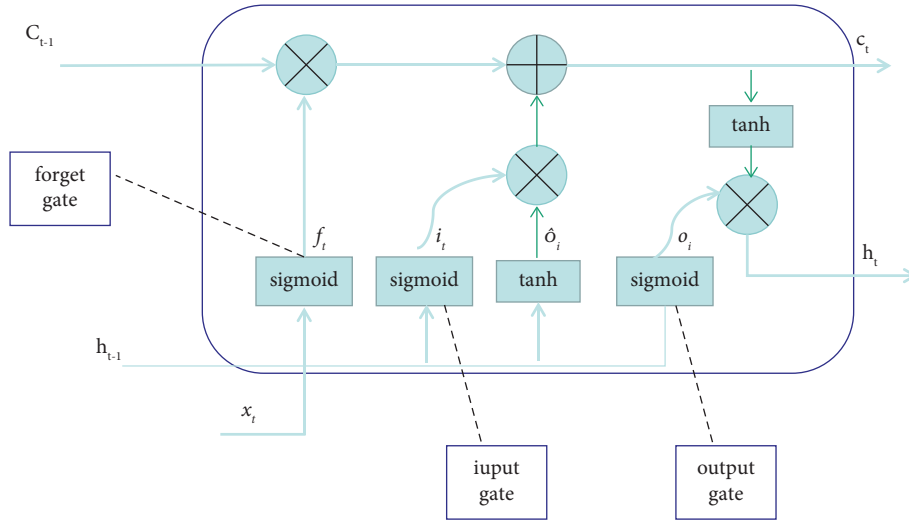

Figure 4: General framework of LSTM.

connection between feature information and predicted value and improving the model's prediction accuracy. The research shows that the LSTM network has positive results in multivariate classification and prediction.

3.2. Bi-LSTM. The LSTM prediction model only predicts through the law of unilateral data, and it cannot fully mine the time feature information, so the prediction accuracy needs further improvement. Targeting the LSTM model's deficiency, a bidirectional-LSTM (bi-LSTM) prediction model is proposed. The structural diagram of Bi-LSTM neurons is shown in Figure 5. Bi-LSTM [37] uses two unrelated LSTM models to predict data from the front and back. The output of the hidden layers of the two models is used as the input of the output layer, and finally, the built-in function of the output layer outputs the final predicted value.

$$\begin{cases} \overrightarrow{h}_t = \mathrm{LSTM}\left(x, \overrightarrow{h}_{t-1}\right), \\ \overleftarrow{h}_t = \mathrm{LSTM}\left(x, \overleftarrow{h}_{t-1}\right), \\ y_{t-1} = g\left(W_{\overrightarrow{h}y}\overrightarrow{h}_t + W_{\overleftarrow{h}y}\overleftarrow{h}_t + b_y\right). \end{cases} \tag{7}$$

Bi-LSTM, based on the time window method, refers to the prediction of the next time step by using the historical value of the time window length of data. The parameter value of the time window step represents the historical data for predicting the future value. For example, if the current value x_t and the previous values x_{t-1} and x_{t-2} are used to predict the value of the next period x_{+1}.

Regularization avoids overfitting in prediction. $L1$ and $L2$ regularization methods introduce a penalty for the problem of too large parameters in the model. The most used regularization technique for deep learning is dropout, which randomly inactivates some neurons. Each training session is equivalent to a different weak classifier, thus improving the model's generalization ability and using the dropout method to improve the model's applicability.

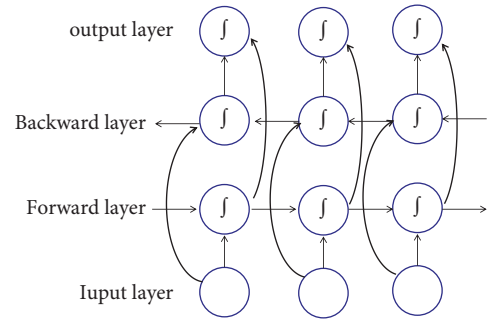

Figure 5: General framework of bi-LSTM.

According to Khan et al. [38], the hybrid network DB-Net, is proposed by combining the extended convolutional neural network (DCNN) with the bidirectional long-term and short-term memory (bi-LSTM). Sagheer and Kotb [39] put forward "CL-Net" based on a new hybrid structure T of ConvLSTM and LSTM. All the above improve LSTM and bi-LSTM deep learning models.

3.3. CNN-LSTM. A convolutional neural network (CNN) comprises five parts: input layer, convolution layer, pooling layer, full connection layer, and output layer.

$\mathbf{X} = [x_1, x_2, \ldots, x_n]$ is the input data matrix, where n represents the length of the time series and m represents the number of data features. The time-series data are convolved to obtain the following equation:

$$o_c = f_c(\mathbf{X} \otimes \mathbf{W}_c + b_c), \tag{8}$$

where $\otimes$ is the convolution operation, convolution kernel $\mathrm{WC} \in \mathbf{R}^{j*m}$ is the weight vector, j is the convolution kernel size, and b_c is the bias of this layer. $f_c(\cdot)$ represents the convolution layer activation function. o_c is the convolution kernel feature mapping result.

Pool operation selects the most critical features of the convolution layer sequence to form the pooling layer. There are two kinds of pooling operations: maximum pooling and average pooling. The commonly used pooling method is maximum pooling, and the maximum global pooling is used in the last pooling operation. The expression is:

$$\begin{cases} o_p(k) = \max\left(o_c(2k-1), o_c(2k)\right), \\ o_p = \max\left(o_c\right), \end{cases} \tag{9}$$

where $o_p(k)$ is the output result of the k^{th} pool; o_p is the output result of maximum global pooling.

A combination of timing features is realized through the full connection layer:

$$o_d = f_d\left(o_p \times \mathbf{W}_d + b_d\right). \tag{10}$$

Among them, $\mathbf{W}_d$ is the weight matrix of the full connection layer, b_d is the bias, and the activation function $f_d(\cdot)$ of the full connection layer includes ReLU, tanh, and sigmoid.

The output layer outputs the results of the full connection layer:

$$y = f_o\left(o_d \times \mathbf{W}_o + b_o\right), \tag{11}$$

$\mathbf{W}_o$ is the weight matrix of the output layer, b_o is the bias, and the activation function f_o is the softmax function.

CNN-LSTM is a combination of CNN and LSTM, which is divided into four layers:

(1) Input layer: data input after normalization.

(2) CNN layer: this layer extracts the data features through CNN, where the convolution layer and pooling layer can extract the features that more clearly reflect the inventory changes and reduce overfitting. The full connection layer can summarise and output the abovementioned features.

(3) LSTM layer: the extracted features are converted into the corresponding data format of LSTM, and time series data mining is carried out through three gate mechanisms in LSTM to obtain the internal change rule and the prediction model.

(4) Output layer: the activation function of the output layer is the Sigmoid function, and the LSTM prediction result is the output.

3.4. DLSTM. In the Deep LSTM (DLSTM) architecture, as shown in Figure 6 [40], the input at time t, x_t is introduced to the first LSTM block along with the previous hidden state $S_{t-1}{}^{(1)}$, and the superscript (1) refers to the first LSTM. The hidden state at time t, $s_t^{(1)}$ is computed and moves forward to the next step and up to the second LSTM block. The second LSTM uses the hidden state $s_t^{(1)}$ along with the previous hidden state $s_{t-1}^{(2)}$ to compute $s_t^{(2)}$, which goes forward to the next step and up to the third LSTM block and so on until the last LSTM block is compiled in the stack.

The benefit of such stacked architecture is that each layer can process some part of the desired task and subsequently pass it on to the next layer until the last accumulated layer finally provides the output. Another benefit is that such architecture allows the hidden state at each level to operate differently. The previous two benefits have a significant impact in scenarios showing the use of data with long-term dependency or in the case of handling multivariate time series datasets.

The prediction results of Bi-LSTM can be compared with LSTM. The model structure of LSTM itself is relatively complex, and training is more time-consuming than CNN. The characteristics of RNN networks determine that they cannot process data in parallel. Furthermore, although LSTM can alleviate the long-term dependence of RNN to some extent, it is difficult for longer sequence data.

4. Experimental Results

4.1. Data Sources. The experimental data in this paper come from dynamic equation (1). According to the definition of the state variable of dynamic system (1), the state variable Z is the inventory data. The first 70000 datasets were used as training datasets and the last 3000 test datasets, totalling 10000. In this paper, system (1) state Z was adopted, and 10000 samples were selected, as shown in Figure 7. The abovementioned analysis showed that the inventory data are chaotic. To fully use the time series between the data, this paper predicts and evaluates the inventory data and verifies it with the actual data.

4.2. Evaluation Index and Model Parameters. This paper used LSTM, bi-LSTM, GRU, CNN-LSTM, and other algorithmic models for prediction. To evaluate the effectiveness of these methods, mean square error (MSE), root mean square error (RMSE), and mean absolute error (MAE) were used to evaluate the model. These indicators are defined as follows [19]:

$$\begin{aligned} \text{MSE} &= \frac{1}{N}\sqrt{\sum_{i=1}^{N}\left(y_i - \widehat{y}_i\right)^2}, \\ \text{RMSE} &= \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(y_i - \widehat{y}_i\right)^2}, \\ \text{MAE} &= \frac{1}{N}\sum_{i=1}^{N}\left|y_i - \widehat{y}_i\right|, \end{aligned} \tag{12}$$

where $\widehat{y}_i$ is the observed inventory quantity, y_i is the forecast quantity of the inventory, and N is the number of test samples.

In this paper, LSTM, DLSTM, GRU, CNN-LSTM, and bi-LSTM algorithms were adopted, and the main parameter values in the algorithms are shown in Table 1.

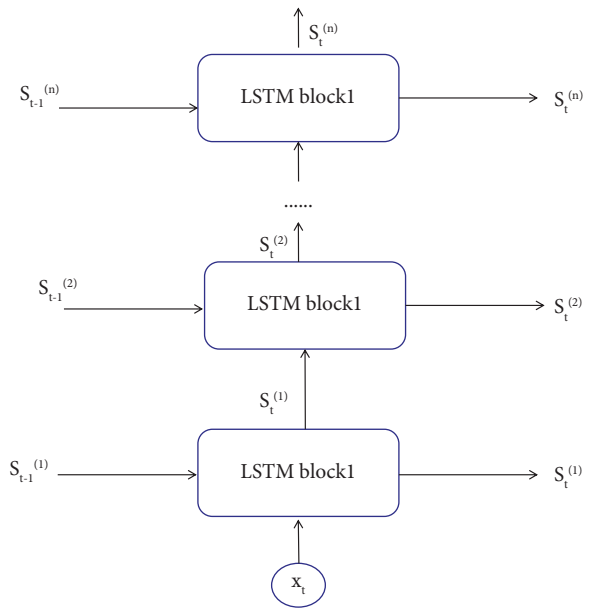

Figure 6: General framework of DLSTM.

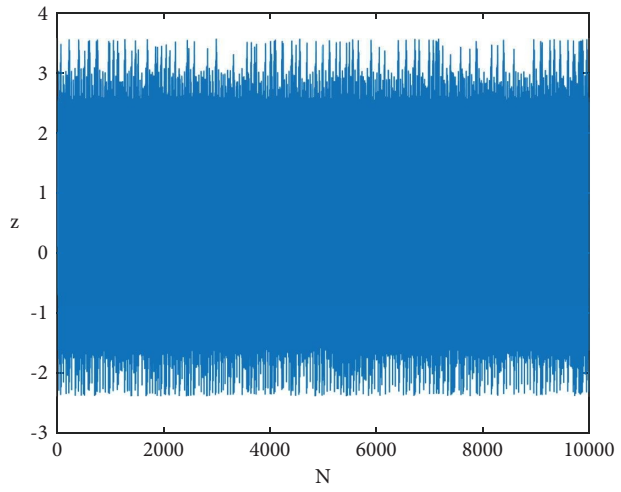

Figure 7: Inventory quantity of system (1).

4.3. Results. The inventory forecasting model adopted the LSTM algorithm, and the comparison between the predicted result and the actual value is shown in Figure 8. The change of the Loss function after 50 cycles is displayed in Figure 9. Figure 8 shows the last 150 data of the test set, allowing the readers to check the predicted and actual values. MSE was 0.005315, RMSE was 0.072905, and MAE was 0.060346. All in all, the prediction errors were quite small.

The comparison between the predicted result by using the bi-LSTM algorithm and the actual value is shown in Figure 10. The change of the Loss function after 50 cycles is shown in Figure 11. Figure 10 shows the last 150 data of the test set for the convenience of readers to check the predicted and actual values. MSE was 0.001475, RMSE was 0.038405, MAE was 0.029732, and the forecasting errors were small.

The inventory forecasting model adopted the CNN-LSTM algorithm. The comparison between the predicted result and the actual values is shown in Figure 12. The change of the Loss function after 50 cycles is shown in Figure 12. Figure 12 shows the last 150 data of the test set for the convenience of readers to check the predicted and actual values. MSE is 0.027766, RMSE is 0.166631, MAE is 0.117720, and the forecasting errors are relatively small.

The inventory forecasting model adopted Figure 13 the DLSTM algorithm. Figure 14 shows that the last 150 data of the test set were used for the convenience of readers to check

Table 1: Parameters of four models.

	LSTM	DLSTM	CNN-LSTM	Bi-LSTM
Number of neurons	80	2	80	80
Dropout	0.3	0.3	0.3	0.3
Loss function	mean_squared_error	mean_squared_error	mean_squared_error	mean_squared_error
Optimizer	Adam	Adam	Adam	Adam
Training times	50	50	50	50
Batch_size	64	1	64	64
Training set	7000	7000	7000	7000
Test set	3000	3000	3000	3000

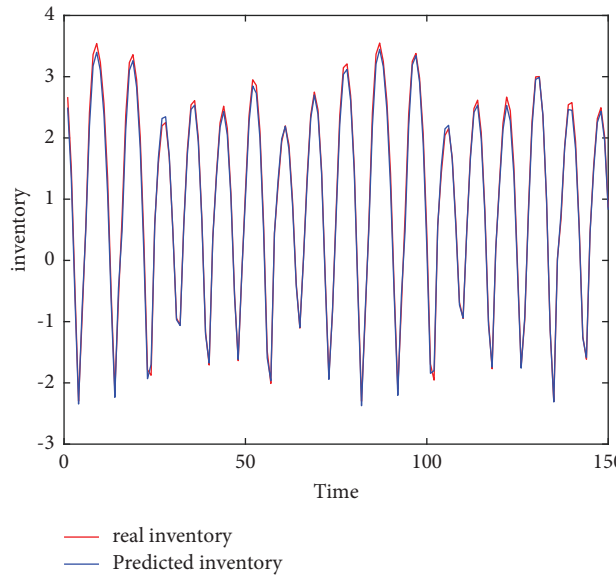

Figure 8: Timing diagram of real and predicted value (LSTM).

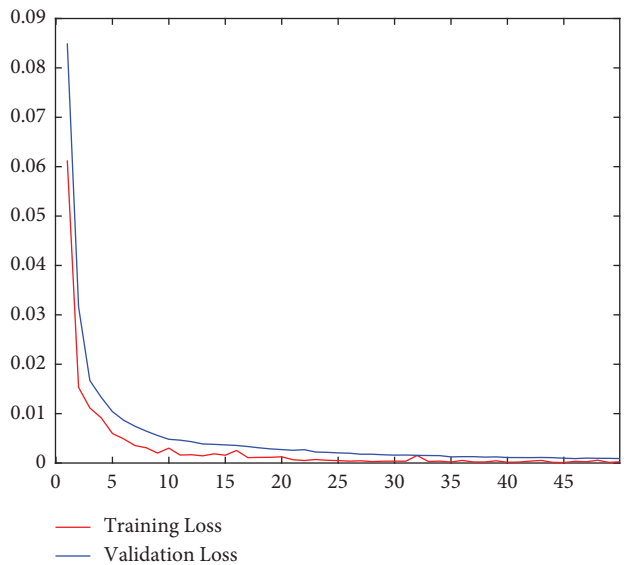

Figure 9: Training and validation loss (LSTM).

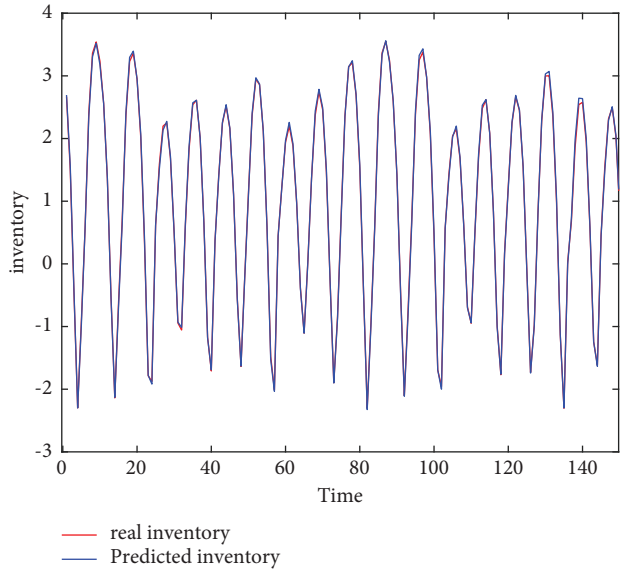

Figure 10: The timing diagram of the actual and predicted values (bi-LSTM).

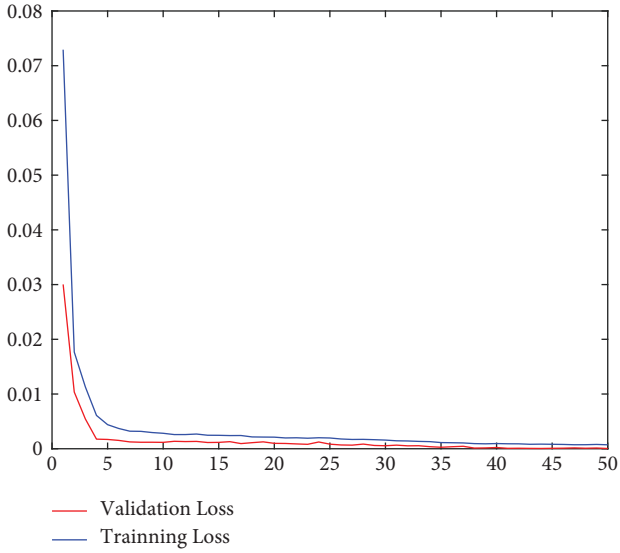

Figure 11: Training and validation loss (bi-LSTM).

the predicted and actual values. The comparison between the predicted result and the actual value is shown in Figure 14. MSE was 0.462163, RMSE was 0.6798, and MAE was 0.570947.

By comparing the abovementioned evaluating indicator, the results are shown in Table 2. The results obtained by bi-LSTM were the best with the slightest error, despite all other algorithms being used due to relatively small errors. Because the data fluctuation was not particularly large, DLSTM had no apparent advantages in this scenario. At the same time, we found no correlation between the complexity and performance of the model. For example, the DLSM algorithm is more responsible but is not the best for inventory safety prediction.

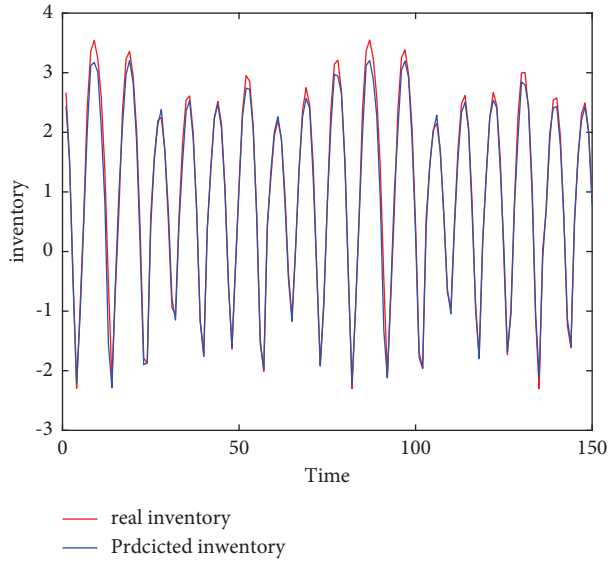

Figure 12: Timing diagram of actual and predicted value (CNN-LSTM).

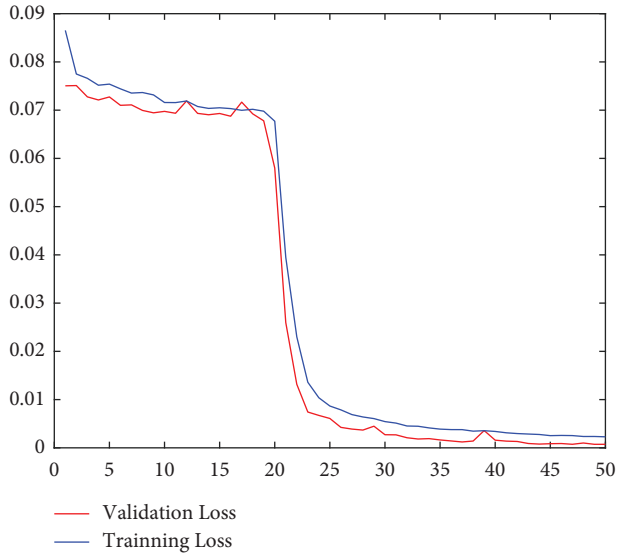

Figure 13: Training and validation loss (CNN-LSTM).

There are often uncertain factors in the production process, such as many sudden orders, temporary consumption increases, the sudden advance of delivery, late delivery, and so on. Through the abovementioned four algorithms, we can see that the bi-LSTM algorithm accurately predicted the inventory capacity, and it is of substantial value for enterprises to make purchase and demand plans.

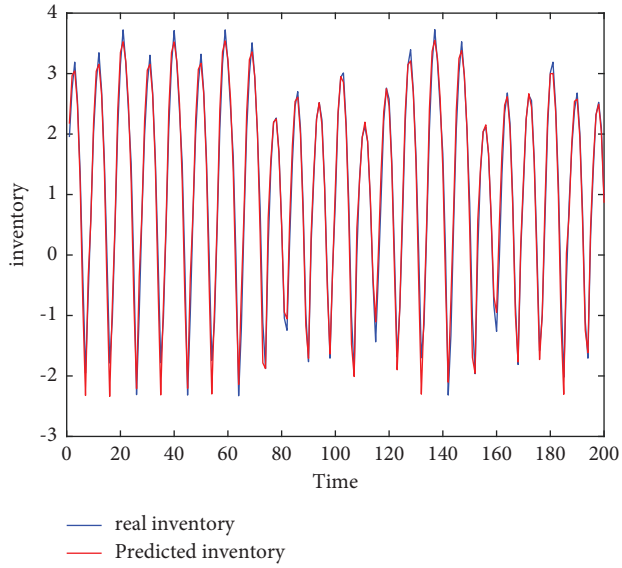

Figure 14: Timing diagram of actual and predicted value (DLSTM).

Table 2: Model results of four deep learning algorithms.

Evaluate	LSTM	Bi-LSTM	CNN-LSTM	DLSTM
MSE	0.005315	0.001475	0.027766	0.462163
RMSE	0.072905	0.038405	0.166631	0.6798
MAE	0.060346	0.029732	0.117720	0.570947

5. Conclusion

Excessive inventory capacity causes inventory backlog, directly affecting the company's production efficiency. In this paper, we focused on the prediction of inventory capacity. It used an inventory management dynamics system to obtain 10000 inventory data and used four prediction algorithms in artificial intelligence: LSTM, BI-LSTM, CNN-LSTM, and DLSTM to train and predict. The prediction results showed that bi-LSTM had the best prediction results.

This study contributed to the academic circle by comparing different forms of neural network prediction of dynamics and chaotic nonlinear inventory management data. It also provided theoretical support for other predictions. The predicted results offered practical suggestions for enterprises' planned production and inventory officers when they decide on the optimal inventory of goods and reduce the likelihood of accidents due to excessive amounts of goods in warehouses. In future work, other algorithms, such as CNN-BILSTM and CNN-DLSTM, as well as AutoML as per Li et al. [41, 42], could be used to predict inventory and compare with the four deep learning methods in this research.

Data Availability

The data used to support the findings of this study are available from the corresponding authors upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the Major Scientific and Technological Innovation Projects of Shandong Province (Grant no. 2019JZZY010111), the Natural Science Foundation of Shandong Province (Grant no: ZR2017PA008), the Key Research and Development Plan of Shandong Province (Grant no: 2019GGX104092), and the Science and Technology Plan Projects of Universities of Shandong Province (Grant no: J18KA381).

References

[1] A. T. Azar and S. Vaidyanathan, *Chaos Modeling and Control Systems Design*, Springer, Cham, Switzerland, 2015.

[2] J. P. Pinder, "Non-linear dynamical systems and inventory management," *Managerial and Decision Economics*, vol. 17, no. 1, pp. 27–43, 1996.

[3] D. Levy, "Chaos theory and strategy: theory, application, and managerial implications," *Strategic Management Journal*, vol. 15, no. 2, pp. 167–178, 2007.

[4] G. Feichtinger, C. H. Hommes, and W. Herold, "Chaos in a simple deterministic queueing system," *ZOR - Methods and Models of Operations Research*, vol. 40, no. 1, pp. 109–119, 1994.

[5] P. Murphy, "Chaos theory as a model for managing issues and crises," *Public Relations Review*, vol. 22, no. 2, pp. 95–113, 1996.

[6] E. C. Joseph, "Chaos & postmodernism forecasting & futuring insights," *Futurics,Summer-Fall*, pp. 1–12, 1994.

[7] Y. Ding and J. Cao, "Bifurcation analysis and chaos switchover phenomenon in a nonlinear financial system with delay feedback," *International Journal of Bifurcation and Chaos*, vol. 25, no. 12, Article ID 1550165, 2015.

[8] S. Wang, S. He, A. Yousefpour, H. Jahanshahi, R. Repnik, and M. Perc, "Chaos and complexity in a fractional-order financial system with time delays," *Chaos, Solitons & Fractals*, vol. 131, Article ID 109521, 2020.

[9] L. Baardman, I. Levin, G. Perakis, and D. Singhvi, "Leveraging comparables for new product sales forecasting," *Production*

and Operations Management, vol. 27, no. 12, pp. 2340–2343, 2018.

[10] S. Van der Auweraer, R. N. Boute, and A. A. Syntetos, "Forecasting spare part demand with installed base information: a review," *International Journal of Forecasting*, vol. 35, no. 1, pp. 181–196, 2019.

[11] X. Yu, Z. Qi, and Y. Zhao, "Support vector regression for newspaper/magazine sales forecasting," *Procedia Computer Science*, vol. 17, pp. 1055–1062, 2013.

[12] T. Shimmura and T. Takenaka, "Selecting rows and columns for training support vector regression models with large retail datasets," *European Journal of Operational Research*, vol. 226, no. 3, pp. 679–683, 2013.

[13] T. Tanizaki, T. Hoshino, T. Shimmura, and T. Takenaka, "Demand forecasting in restaurants using machine learning and statistical analysis," *Procedia CIRP*, vol. 79, pp. 679–683, 2019.

[14] J. Meller, F. Taigel, and R. Pibernik, "Prescriptive analytics for inventory management: a comparison of new approaches," *SSRN Electronic Journal*, 2018.

[15] A. L. Beutel and S. Minner, "Safety stock planning under causal demand forecasting," *International Journal of Production Economics*, vol. 140, no. 2, pp. 637–645, 2012.

[16] A. Koesdwiady, R. Soua, and F. Karray, "Improving traffic flow prediction with weather information in connected cars: a deep learning approach," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 12, pp. 9508–9517, 2016.

[17] F. Kong, J. Li, B. Jiang, and H. Song, "Short-term traffic flow prediction in smart multimedia system for Internet of Vehicles based on deep belief network," *Future Generation Computer Systems*, vol. 93, no. 4, pp. 460–472, 2019.

[18] G. Wei and Z. Wang, "Adoption and realization of deep learning in network traffic anomaly detection device design," *Soft Computing*, vol. 25, no. 2, pp. 1147–1158, 2021.

[19] J. Zhang, Y. Zheng, and D. Qi, "Deep spatio-temporal residual networks for citywide crowd flows prediction," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 31, no. 1, pp. 1655–1661, 2017.

[20] Y. Lv, Y. Duan, W. Kang, Z. Li, and F. Y. Wang, "Traffic flow prediction with big data: a deep learning approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 865–873, 2015.

[21] D. Zhang and M. R. Kabuka, "Combining weather condition data to predict traffic flow: a GRU-based deep learning approach," *IET Intelligent Transport Systems*, vol. 12, no. 7, pp. 578–585, 2018.

[22] S. Du, T. Li, X. Gong, Y. Yang, and S. J. Horng, "Traffic flow forecasting based on hybrid deep learning framework," in *Proceedings of the 12th International Conference on Intelligent Systems and Knowledge Engineering*, pp. 1–6, Nanjing, China, November 2017.

[23] A. S. Mihaita, H. Li, Z. He, and M. A. Rizoiu, "Motorway traffic flow prediction using advanced deep learning," in *Proceedings of the 2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pp. 1683–1690, Auckland, New Zealand, October 2019.

[24] L. Qu, W. Li, W. Li, D. Ma, and Y. Wang, "Daily long-term traffic flow forecasting based on a deep neural network," *Expert Systems with Applications*, vol. 121, no. 5, pp. 304–312, 2019.

[25] Y. Kim, P. Wang, and L. Mihaylova, "Scalable learning with a structural recurrent neural network for short-term traffic prediction," *IEEE Sensors Journal*, vol. 19, no. 23, pp. 11359–11366, 2019.

[26] Y. Chen, L. Shu, and L. Wang, "Traffic flow prediction with big data: a deep learning based time series model," in *Proceedings of the 2017 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 1010-1011, Atlanta, GA, USA, May 2017.

[27] H. F. Yang, T. S. Dillon, and Y. P. P. Chen, "Optimized structure of the traffic flow forecasting model with a deep learning approach," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 10, pp. 2371–2381, 2017.

[28] S. U. Khan and R. Baik, "MPPIF-net: identification of Plasmodium falciparum parasite mitochondrial proteins using deep features with multilayer Bi-directional LSTM," *Processes*, vol. 8, no. 6, p. 725, 2020.

[29] I. U. Haq, A. Ullah, S. U. Khan et al., "Sequential learning-based energy consumption prediction model for residential and commercial sectors," *Mathematics*, vol. 9, 2021.

[30] N. Khan, F. Ullah, I. U. Haq, S. U. Khan, M. Y. Lee, and S. W. Baik, "AB-net: a novel deep learning assisted framework for renewable energy generation forecasting," *Mathematics*, vol. 9, no. 19, p. 2456, 2021.

[31] X.. Gao, J. Mou, L. Xiong, Y. Sha, H. Yan, and Y. Cao, "A fast and efficient multiple images encryption based on single-channel encryption and chaotic system," *Nonlinear Dynamics*, vol. 108, no. 1, pp. 613–636, 2022.

[32] X. Ma, J. Mou, J. Liu, C. Ma, F. Yang, and X. Zhao, "A novel simple chaotic circuit based on memristor-memcapacitor," *Nonlinear Dynamics*, vol. 100, no. 3, pp. 2859–2876, 2020.

[33] X. Li, J. Mou, Y. Cao, and S. Banerjee, "An optical image encryption algorithm based on a fractional-order laser hyperchaotic system," *International Journal of Bifurcation and Chaos*, vol. 32, no. 03, 2022.

[34] X. Gao, J. Mou, S. Banerjee, Y. Cao, L. Xiong, and X. Chen, "An effective multiple-image encryption algorithm based on 3D cube and hyperchaotic map," *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 4, pp. 1535–1551, 2022.

[35] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[36] T. Lei, R. Y. M. Li, and H. Fu, "Dynamics analysis and fractional-order approximate entropy of nonlinear inventory management systems," *Mathematical Problems in Engineering*, vol. 2021, Article ID 5516703, 8 pages, 2021.

[37] Z. Li, J. P. Zhu, X. J. Xu, and Y. Yao, "RDense: a protein-RNA binding prediction model based on bidirectional recurrent neural network and densely connected convolutional networks," *IEEE Access*, vol. 8, pp. 14588–14605, 2020.

[38] N. Khan, I. U. Haq, S. U. Khan, S. Rho, M. Y. Lee, and S. W. Baik, "DB-Net: a novel dilated CNN based multi-step forecasting model for power consumption in integrated local energy systems," *International Journal of Electrical Power & Energy Systems*, vol. 133, Article ID 107023, 2021.

[39] N. Khan, I. U. Haq, F. U. M. Ullah, S. U. Khan, and M. Y. Lee, "CL-net: ConvLSTM-based hybrid architecture for batteries' state of health and power consumption forecasting," *Mathematics*, vol. 9, no. 24, p. 3326, 2021.

[40] A. Sagheer and M. Kotb, "Time series forecasting of petroleum production using deep LSTM recurrent networks," *Neurocomputing*, vol. 323, pp. 203–213, 2019.

[41] R. Y. M. Li, L. Song, B. Li, M. J. C. Crabbe, and X. G. Yue, "Predicting carpark prices using real estate indices and AutoML," *CMES-Computer Modeling in Engineering & Science*, vol. 134, no. 3, pp. 2247–2282, 2022.

[42] R. Y. M. Li, K. W. Chau, and D. C. W. Ho, *Current State of Art in Artificial Intelligence and Ubiquitous Cities*, Springer, Berlin, Germany, 2022.