



An improved deep echo state network inspired by tissue-like P system forecasting for non-stationary time series

Xiaojian Yang^{1,2} · Qian Liu^{1,2} · Xiyu Liu^{1,2} · Jie Xue^{1,2}

Received: 22 February 2022 / Accepted: 20 July 2022 / Published online: 24 August 2022
© The Author(s) 2022

Abstract

As a recurrent neural network, ESN has attracted wide attention because of its simple training process and unique reservoir structure, and has been applied to time series prediction and other fields. However, ESN also has some shortcomings, such as the optimization of reservoir and collinearity. Many researchers try to optimize the structure and performance of deep ESN by constructing deep ESN. However, with the increase of the number of network layers, the problem of low computing efficiency also follows. In this paper, we combined membrane computing and neural network to build an improved deep echo state network inspired by tissue-like P system. Through analysis and comparison with other classical models, we found that the model proposed in this paper has achieved great success both in predicting accuracy and operation efficiency.

Keywords Deep echo state network · Membrane computing · Time series prediction

1 Introduction

With the development of modern information technology and the rapid growth of data, the value of data is getting more and more attention. Decision-making in various fields depends on data processing and analysis. A large number of time series data also exist in various fields of life, such as the financial field, the transportation field, and the astronomical field. Time series analysis is the process of processing dynamic data. It needs to be analyzed on the basis of existing data to obtain the useful information contained in the data and realize the extraction of value [1, 2]. For data with time series, there are usually difficulties, such as large amount of data, high complexity, high storage cost and low calculation efficiency, so it is of great significance for the mining and analysis of time series data [3], especially for nonstationary time series.

After decades of development, time series analysis has formed a complete system. Machine learning has made great progress in the application of time series analysis and

prediction, especially the prediction of a small amount of data with low dimensions. However, with surging data volume in today's society, traditional machine learning algorithms showed their obvious deficiencies. Theoretically speaking, thanks to a good ability of nonlinear mapping and self-adaptation, recurrent neural network (RNN) is an ideal tool for dealing with modeling and forecasting problems of time series [4, 5]. This model is a network with a recurrent structure. Its working principle is to transport the information obtained from the network layer at the previous time to the network layer at the next time. The output of the hidden layer is determined by the sequence information of the past time. Echo state network (ESN) is a new type of RNN, which input weights and reservoir weights are randomly generated on the basis of a certain probability distribution, and fixed, the only that need to be trained are output weights, by simple linear regression. This training method of ESN can achieve the global optimal of weights, simplify the training process of classical RNN. Therefore, compared with the gradient-based traditional RNN, ESN overcomes the problems of low computational efficiency and easy to fall into local optimum [6]. In spite of success in time series prediction due to its higher generalization ability, ESN performed poorly when processing stochastic, nonlinear, non-stationary time series. The rapid development of deep learning and its application in ESN has solved this problem [2, 7], at the cost of adding

✉ Xiyu Liu
xyliu@sdu.edu.cn

¹ Business school, Shandong Normal University, Jinan, China

² Academy of Management Science, Shandong Normal University, Jinan, China

model complexity, thus compound the difficulty of model construction.

In 1998, Academician Gheorghe Păun proposed membrane computing, a computing method inspired by nature [8, 9]. Its development starts from the observation of cells, and this new computing model is built by abstracting from the structure and function of cells, also known as the P system. tissue-like P system is the abstract of living organisms, so contains a number of cells, cells have connected channels through which they can communicate directly, and cells without connecting channels can communicate indirectly through the environment. Therefore, tissue-like P system is easier to implement information exchange [10, 11].

Therefore, our motivation is to design a deep neural network based on tissue-like P system while achieving better performance than traditional deep neural networks. In this paper, we focus on the NARMA signal, as these are the most common time series in correlational research, yet verify our model performance.

The rest of this paper is organized as follows. Section 2 introduces the related work. Section 3 describes an overview of the background works and proposes a deep echo state network (DESN) based on tissue-like P systems. The prediction performance of the model and the analysis of experimental results are presented in Sect. 4. Finally, some conclusions are given in Sect. 5.

2 Related work

2.1 Echo state network

As shown in Fig. 1, ESN is the structural basis of deep ESN. They have exactly the same reservoir structure and the same training mechanism. Echo state network (ESN) is one of RC (reservoir computing) algorithms, which is developed on the basis of recurrent neural network. To solve the problems of large training resource consumption and long running time of cyclic neural network, ESN came into being [12, 13].

The echo state network was proposed in 2001. It is a special type of RNN, and it is also composed of input layer, hidden layer and output layer. The difference between ESN and

traditional neural networks is that it adds a randomly connected reserve pool to replace the original hidden layer. The connection state of neurons in the reserve pool is random, and the connection weight is fixed. This allows it to effectively reduce the amount of calculation during the training process, and to a certain extent avoid the phenomenon of local minima during the gradient descent process [14]. The reservoir accepts two directions of input, one from the input layer, and the other from the output of the previous state of the reservoir, where the state feedback weight is the same without training, and it is determined by the random initial state [15, 16]. ESN uses randomly connected neurons in the reserve pool to generate a complex state space. The input data on the left is linearly combined with the state space to obtain the output data on the right.

The classical ESN contains three layers: a input layer, a reservoir and a output layer, as shown in Fig. 1. The external input, the reservoir and the output vector are denoted by the vector $u(n)$ of size M , $x(n)$ of size N , $y(n)$ of size P , respectively. ESN contains three relatively independent topology structures accordingly: the connection weight from the input layer to the reservoir is the matrix W^{in} of size $M \times N$; the recursive connection weight inside the reservoir is the matrix W of size $N \times N$; the output connection weight from the reservoir to the output layer is W^{out} . Sometimes the neurons of the input layer and the output layer can be connected. The recursive formula for the state of each part of ESN over time is as follows:

$$x(n) = \gamma f(W \cdot x(n-1) + W^{in} \cdot u(n)) + (1 - \gamma)x(n-1) \tag{1}$$

$$y(n) = g(W^{out}[u(n);x(n)]) \tag{2}$$

where the activation function $f(\cdot)$ is usually the tanh function, and $g(\cdot)$ is the linear function. γ is the leaky rate and contains a value between 0 and 1, representing the proportion of the state $x(n-1)$ of the reservoir at the previous time to the state $x(n)$ of the reservoir at the current time. $\gamma = 1$ means we completely ignore the state of the reservoir at the previous time. $\gamma = 0$ means that all the state $x(n-1)$ of the reservoir at the previous time is assigned to the state $x(n)$ of the reservoir at the current time. The training method of ESN can adopt simple linear regression method, i.e., solving the formula linear equation:

$$\bar{y} = wB \tag{3}$$

where B is a T -rows matrix which each row vector is $[u(n); x(n)]$ (T represents the number of training samples). Each row of matrix \bar{y} is the output vector of the system, $t = 1, \dots, T$. is the output connection weight matrix that needs to be solved. Three parameters need to be set to construct an ESN: n , λ_{max} and ρ . As mentioned above, n is the

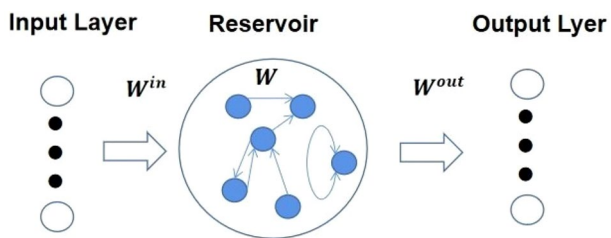


Fig. 1 Structure of the echo state network

number of neurons in the reservoir. The spectral radius λ_{\max} of the reservoir is defined by the formula:

$$\lambda_{\max} = \max\{\text{abs}(w)\} \quad (4)$$

The ESN's echo state property can be guaranteed only if $\lambda_{\max} < 1$. The echo state property means that the state of the neurons in the reservoir can not be affected by the initial value after some iterations.

The sparsity ρ of the reservoir determines the percentage of non-zero terms. The training goal of ESN is to minimize the error between the output of the model and the actual output of the system. This situation can also be extended to the data that has not been trained.

Because the structure of the echo state network enriches the theory of traditional neural networks, and the parameter learning is simpler and faster, it has become an effective tool for studying time series data. Coulibaly et al. used the echo state network to predict the monthly average water level of four lakes in the United States as an example. In the hydrological time series prediction problem, it proved that the echo state network method is superior to the traditional recurrent neural network [17]. Regarding the incoming traffic load of the mobile network, Bianchi et al. used the echo state network to predict and obtained a better prediction effect [18]. Through the improvement of the echo state network, Liu et al. applied it to the production process of iron and steel enterprises, and they were also satisfied with the prediction results of the amount of blast furnace coal. A hybrid echo state network with complex network characteristics was proposed by Cui et al. The complex network theory is introduced on the basis of the traditional echo state network, which effectively improves the accuracy of time series prediction [19]. Najibi et al. proposed three new types of echo state networks. These three networks use K-means, PAM and Ward algorithms to construct the structure of the reservoir. The prediction effect on the chaotic time series is significantly better than that of the traditional network [20].

2.2 Tissue-like P system

P system is distributed computational parallel models, inspired by the structure and functions of cells, tissues and organs [21]. The P system contains communication rules to realize various functions [22]. Communication between cells is realized by the exchange between objects. The execution of rules within cells meets the maximum parallelism, and each cell can operate independently, which makes P system with the maximum parallelism [23, 24]. The original tissue-like P system was defined as

$$\Pi = (O, \omega_1, \omega_2, \dots, \omega_m, \text{ch}, i_0) \quad (5)$$

where O represents finite non-empty alphabets of objects; $\text{ch} \subseteq \{1, 2, \dots, m\} \times \{1, 2, \dots, m\}$, ch represents the communication channel between cell i and cell j ; $i_0 \subseteq \{1, 2, \dots, m\}$ represents that cell i is the output cell of the system, used to output the results of the computation; $\omega_1, \omega_2, \dots, \omega_m$ are strings within m cells, m is the number of cells, and its specific form is defined as

$$\omega_i = (Q_i, s_{i,0}, w_{i,0}), 1 < i < m \quad (6)$$

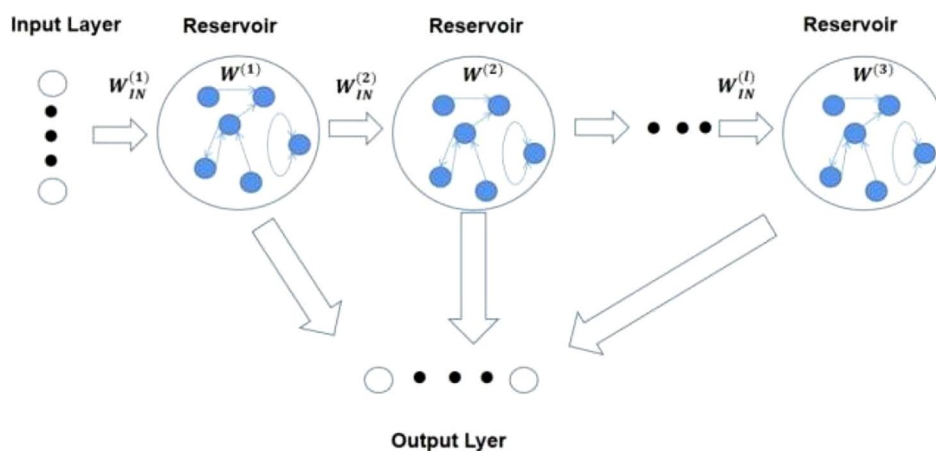
where Q_i is a finite set of states; $s_{i,0} \subseteq Q_i$ represents the initial state of the cell; $w_{i,0} \subseteq O$ represents the multiple sets of objects contained in cell i in the initial state; R_i is the set of rules inside the cell.

In the study of tissue P system, a large number of scholars have studied the computing power of this system and its variants, which has a strong computing efficiency in solving practical problems [25]. Song et al. [26] proposed a one-way tissue P system with symport rules (MTS P system), in which two regions communicate only in one direction. It is proved that the MTS P system still has strong computing capability under the restriction of unidirectional. Most rule application strategies in P system adopt maximum parallelism. This rule synchronization strategy is considered in [27], and the tissue P system with synchronous symport/antiport rules is proposed. It is proved that the system can also solve the SAT problem under the rule of cell division. Cells can also work asynchronously. Pan et al. [28] studied the computing power of local synchronization in asynchronous tissue P systems with symport/antiport rules at the three levels of rules, channels and cells. The steady-state mechanism is introduced in [29]. The environment no longer provides energy to cells, and the multi-set rewriting rules are introduced into tissue P system, a steady tissue P system is constructed, and proved to be an effective solution to the NP-complete problem.

2.3 Time series predictions

Time series prediction models have evolved from early linear models, such as autoregressive moving average models [30], to better nonlinear models, such as neural network models [31]. The development from linear regression modeling method with clear mathematical relations to black box nonlinear modeling method requires researchers to be able to adopt more effective theoretical methods, such as machine learning [32], fuzzy reasoning [33], heuristic, neural network [34] and other artificial intelligence methods. There are obvious differences between various linear and nonlinear modeling methods. On the one hand, the difference lies in the different mathematical methods adopted by each prediction method. On the other hand, the difference lies in the significant differences in the theoretical methods used by

Fig. 2 Structure of the deep echo state network



different prediction methods. In the implementation of the algorithm, the computational resources required by different prediction methods are also very different. Different prediction methods have different mechanisms for extracting data features. The methods for extracting data features can be divided into two forms: explicit and implicit. Explicit feature extraction method is to build features by directly transforming finite length historical data at each time step. In contrast, the implicit feature extraction method is to construct the internal dynamic features contained in the historical data through machine learning method. This method does not need to strictly consider the time relationship of the data in the time series. Different forecasting methods can be distinguished by forecasting model data characteristics and training methods. A variety of common time series forecasting methods will be introduced in the following sections.

3 Methods

3.1 Framework of deep echo state network

Hinton et al. proposed an unsupervised greedy layerwise training algorithm, opening the door of deep learning research. Deep ESN is the result of ESN combined with deep learning thought. Its essence is multi-layer artificial neural network, with simple input layer and output layer, and multiple reservoir structures as hidden layer. DESN has more reservoir structures and can map more complex time-series applications. Depth the echo state network (DESN) structure, as shown in Fig. 2, the left is the input layer, its function is to the external data into the depth of the echo state network, the bottom is output layer, its function is the depth of the echo state network output the generated data to the external, is among multiple hidden layer, each layer is a dynamic structure of reservoir. Refer to Ying-Chun Bo's

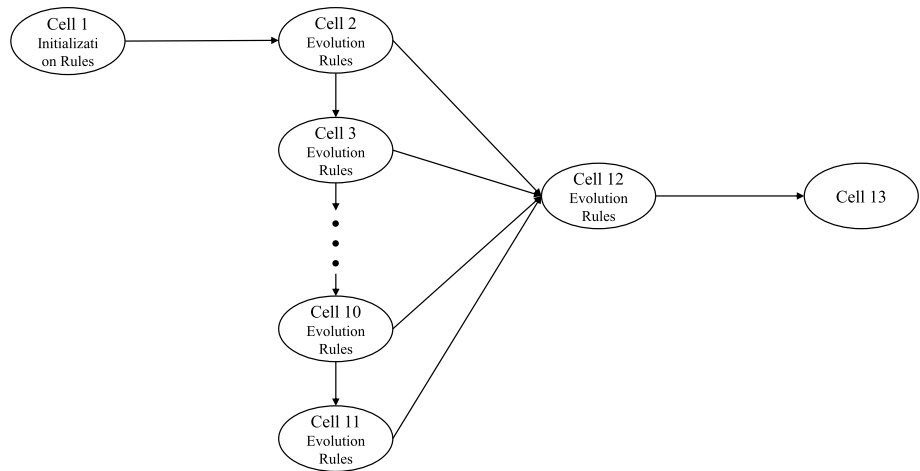
research, we stacked ten reservoirs as hidden layers of the DESN.

The DESN works like this: the input layer loads external data into the DESN, $u(n) \subseteq R^{k \times l}$, K is the dimension of the input data, which determines the number of neurons in the input layer. External data enter DESN and enter the first reservoir after weighted by input weight $W_{in}^{(1)} \subseteq R^{N_1 \times k}$, where N_1 represents the number of neurons in the first reservoir. In the reservoir neurons of the first layer, the state value of the previous historical timepoint $x^{(1)}(n-1)$ inside the reservoir is weighted by reservoir weight $W^{(1)} \subseteq R^{N_1 \times N_1}$. After summing with the received weighted input, a new state value is formed from the activation function of the neuron, thus updating the state value of the first reservoir, denoting as $x^{(1)}(n) \subseteq R^{N_1 \times N_1}$. Like the first reservoir layer, the status of the second reservoir is updated from $x^{(2)}(n-1)$ to $x^{(2)}(n) \subseteq R^{N_2 \times l}$. The model repeats the working process of the reservoir until the state value of the neurons in the last reservoir is updated. The state value of the last reservoir is denoted as $x^{(L)}(n) \subseteq R^{N_L \times l}$, where L represents the maximum number of layers, also known as the depth. At this point, all the status value of reservoir neurons has updated, $N = N_1 + N_2 + \dots + N_L$ is total number of the neurons of DESN reservoirs. All state values are sorted together and denoted as $x(n) = [(x^{(1)}(n))^T, (x^{(2)}(n))^T, \dots, (x^{(L)}(n))^T]^T \subseteq R^{N \times l}$, all of the status value weighted by the output weights $W^{out} \subseteq R^{(M \times N)}$ is feed to the output layer. The output layer outputs the final result to the network. M represents the dimension of output data, and is also the number of neurons in the output layer. So far, the flow of data in the network is finished.

3.2 The tissue-like P system based on deep echo state networks

The membrane structure inside the class organization forms a network structure, and the system objects change

Fig. 3 Structure of the tissue-like P system



their positions and states through communication rules and evolution rules. This paper applies the internal organization to the sonic state network, transfers the objects between the outer membranes through communication rules, and uses evolution rules to complete the state changes of the objects.

We constructed a tissue P system with 13 cells, as shown in Fig. 3, and its formal definition is

$$\Pi = (O, \omega_1, \dots, \omega_{13}, ch, i_0) \tag{7}$$

where

- (1) $O = \{x_1, x_2, \dots, x_n\}$ is a set of objects, where $x_j (j = 1, 2, \dots, n)$ is the j th vector;
- (2) $\omega_i (i = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13)$ represents the string inside the cell, its form is as follows:
 $\omega_i = (Q_i, s_{i,0}^t, \omega_{i,0}, R_i, r_{ij}), 1 \leq i \leq 13, 1 \leq j \leq 13, Q_i = (s_{i,1}^t, s_{i,2}^t, \dots, s_{i,13}^t)$, where $t = (0, 1, \dots, t_{\max}), s_{i,0}^t (i = 1, 2, \dots, 13)$ represents the state of the i th object in membrane i at time $t, \omega_{i,0} \in O^*$ represents the multiset of initial objects. R_i is a finite set of rules, which represents the evolution rules in cell i . It will change the state of objects in the cell. $r_{ij} = (i, u/j, \lambda)$ is a communication rule. This rule indicates that the string u can be transmitted from the cell i to the cell j , and the cell j cannot transmit the cell i . Because λ is an empty string. In this system, all communication rules are one-way transmission rules.
- (3) $ch = \{(1, 2), (2, 3), (3, 4), (4, 5), (5, 6), (6, 7), (7, 8), (8, 9), (9, 10), (10, 11), (2, 12), (3, 12), (4, 12), (5, 12), (6, 12), (7, 12), (8, 12), (9, 12), (10, 12), (11, 12), (12, 13)\}$ represents the connecting channel between different cells. For example, cell 2 can only receive information from cell 1, while cell 12 can receive information from cells 2 to 11;
- (4) $i_0 = 13$ indicates that cell 13 is the output cell of the entire system.

3.3 Operating mechanism

Based on the tissue-like P system, the objects of the original echo state network will change within the membrane structure. The rules in each membrane will be executed independently and will not affect each other, which will greatly improve the calculation efficiency.

(1) Initialization rules

To perform the task of forecasting time series data, the system generates all initial objects in the input cell (cell 1), and each object represents a one-dimensional or multi-dimensional vector. The dimension of the time series data in this article is $w \times 1$, and then the objects are normalized in the input cell, mapped in the range of $[0 - 1]$, and these objects are transmitted to cell 2 through communication rules.

(2) Evolutionary rules

The principle of extreme parallelism is adopted when the rules in the P system are executed, that is, all the rules that meet the conditions are executed in the system. The execution of the rules in cells 2–11 is done according to Eq. 1. Linearly combines the object of the previous cell with the object of the current cell and transports it to the next cell. At the same time, all the objects in the cell are sent to the cell 12 for calculation according to Eq. 2. The final result $y(n)$ is obtained and transferred to cell 13 for storage. We assume that the state in cell 1 is $x(1)$, and the state in cell 2 is $x(2)$. In cell 2, $x(1)$ and $x(2)$ are combined according to the weight by executing the evolution rule, and the result is output to cell 12. The above evolution process iterates multiple times in cell 2 to cell 11 until the optimal result is obtained.

(3) Communication rules

The communication rules in this system are all one-way transmission rules. There is a one-way communication rule in each cell channel. The communication rules for cell 1 and cell 12 are specific to one cell only. Cell 1 only transmits information to cell 2, and cell 12 only transmits information

to cell 13. However, other cells transmit messages to both cells separately. One is the next cell and the other is cell 12. For example, $r_{1,2} = (1, x(1)/2, \lambda)$, $r_{2,3} = (2, x(2)/3, \lambda)$, $r_{2,12} = (2, x(2)/12, \lambda)$. The rule $r_{1,2}$ transfers $x(1)$ inside cell 1 to cell 2. $x(2)$ in cell 2 is transferred to cell 3 through $r_{2,3}$, and at the same time through $r_{2,12}$ transfer $x(2)$ into cell 12.

(4) Termination condition and output

After the objects in the former cell 12 complete all the evolutionary rules, the cell 12 delivers the final result to the output cell (cell 13), the calculation process stops, and the membrane terminates the evolution. Finally, all objects in the output film are considered the final result.

4 Numerical experiments and results analysis

4.1 Comparative methods

Many traditional methods, such as ARMA [30] and ARIMA [35], have been well applied in stationary time series prediction. However, in the our application, time series is non-stationary, which limits the application of the above stationary method and reduces the generalization ability of the traditional time series prediction method. Therefore, we used the deep learning model as the benchmark. To evaluate the performance of deep echo state networks based on tissue-like P system, we compare some well-known RNN [4, 5] prediction models in deep learning: convolutional neural network (CNN) [36], long short-term memory (LSTM) [4], traditional ESN [6]. In the past studies, these models have been achieved the great

success and widely applied in time series prediction tasks. CNN can solve problems, such as regression and classification in various application fields. RNN, as a deep learning model for processing sequence data, also shows good performance in time series and speech processing.

4.2 Experimental data and experimental environment

This paper selects the benchmark task commonly used in time series prediction, the NARMA signal. The non-linear auto-regressive (NARMA) data set were originally proposed by Jaeger and included modeling of the following R-order system outputs:

$$y(n + 1) = 0.3y(n) + 0.55y(n) \left[\sum_{i=0}^r y(n - i) \right] + 1.5x(n - r)x(n) + 0.1. \tag{8}$$

The input $x(n)$ of the system is the noise randomly distributed between [0, 1]. The NARMA task requires memory of at least r past time steps, since the output of the system $y(n + 1)$ is determined by the input and the outputs of the r time steps. 10,000 data points were collected as experimental data. The first 80% of data points were used to construct the training set, and the remaining 20% were used as the test set. Figure 4 shows the change trend.

To avoid the influence of value range on the model, the input was pre-processed. The original time series data were linearly transformed before entering the data to the interval [0,1]. The normalization formula is

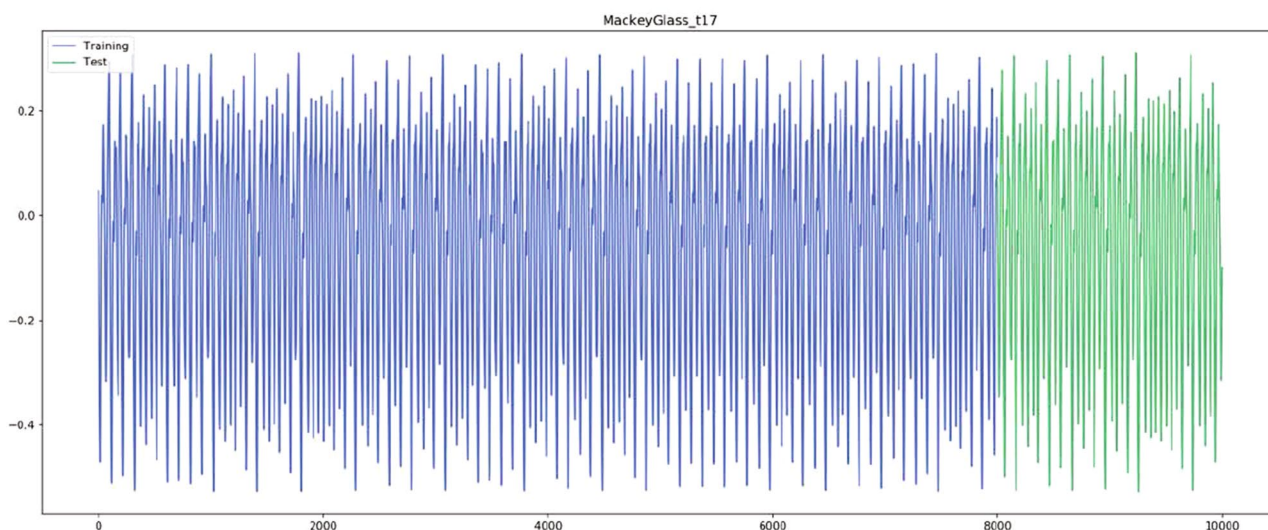


Fig. 4 NARMA data set

$$x_{\text{norm}} = \frac{x - x_{\text{min}}}{x_{\text{max}} - x_{\text{min}}}. \quad (9)$$

In this paper, the tensorflow open source platform is used as the deep learning platform, and Python 3.7 is used to write the experimental program. Meanwhile, some third-party libraries are used, such as Talib to calculate technical indicators and Keras to build the network structure. The experimental operating system is Windows 10.

4.3 Performance assessment

This paper evaluates all prediction models in terms of model accuracy. For model accuracy, we choose root mean square error (RMSE) as the measurement standard. The calculation formula of ERMSE can be expressed as

$$\text{RMSE} = \sqrt{\sum_{t=1}^T (y_t - f_t)^2 / T}, \quad (10)$$

where y_t and f_t are, respectively, the observed value and output value of the model at time t , and T is the number of data points. In this paper, the fitting and prediction accuracy of the model is quantitatively evaluated by calculating RMSE values of the training set and the test set, respectively. In this article, we do not expand the RMSE value of the training set.

5 Results

We expect to explore the performance of the proposed model for time series forecasting in the NARMA signal data set. As shown in Figs. 5, 6, 7 and 8 and Table 1, for NARMA signal, the ESN with a single reservoir architecture achieves the same performance as LSTM. The RMSE of the ESN is slightly higher than that of the LSTM. As we increase the number of the reservoir, the tissue-like p system based on deep echo state networks showed significant improvements in predictive performance. The proposed model achieves much better RMSE than other models. It is worth noting that the complexity of NARMA time series is relatively low. In the above experiment, we prove that the model we proposed is more effective than other RNN in time series prediction. The proposed model not only has a great improvement in the prediction accuracy, but also has a certain advantage in the computation time compared with the traditional DESN.

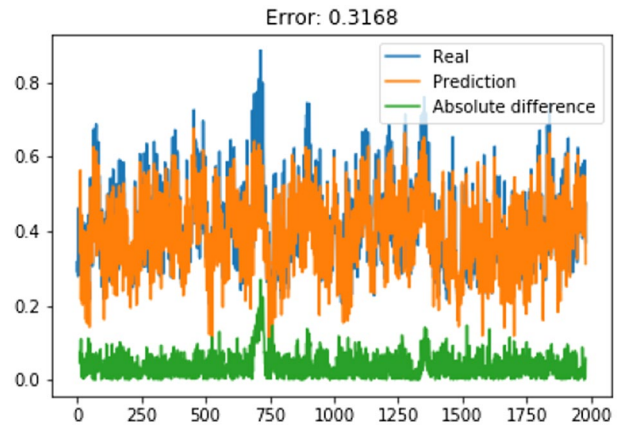


Fig. 5 CNN

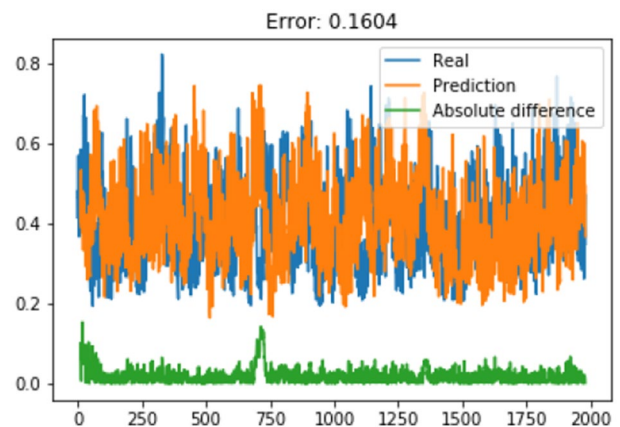


Fig. 6 LSTM

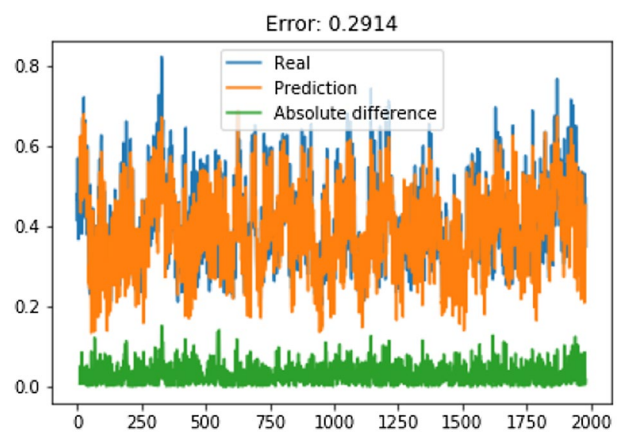


Fig. 7 ESN

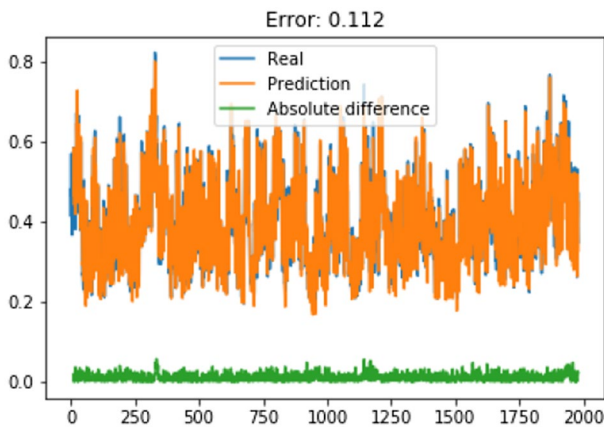


Fig. 8 Our model

Table 1 Prediction performance of models for NARMA signal

Models	RMSE
LSTM	0.0219
Traditional ESN	0.0192
CNN	0.301
Proposed model	1.01e−05

6 Conclusions

In the field of time series forecasting, we expect to combined membrane computing and neural network to build a more computationally efficient model that can directly process time series. Although we have only made some preliminary improvements, some useful conclusions that are helpful to the future research have been given in this paper.

In this paper, the tissue-like P system based on deep echo state networks model was presented for modeling time series. Although the deep ESN model effectively improves the accuracy of prediction, it also greatly increases the computational complexity. The parallel and integration of P system will greatly improve the efficiency of the model. The application of the tissue-like p system framework overcomes the inherently low efficiency of deep neural network. Therefore, the model we proposed has some advantages. Since the P system is applied on the basis of the model in this paper, it cannot operate independently from the model and is dependent on the original model.

This article focuses on use the tissue-like P system to implement the operation process of the neural network. However, P system also has many applications in finding optimal solutions. We expect to find the application of P system in the parameter optimization of neural networks.

Funding This research project is supported by National Natural Science Foundation of China (61876101, 61802234, 61806114, 62172226), Social Science Fund Project of Shandong Province, China (16BGLJ06, 11CGLJ22), Natural Science Fund Project of Shandong Province, China (ZR2019QF007), Postdoctoral Project, China (2017M612339, 2018M642695), Humanities and Social Sciences Youth Fund of the Ministry of Education, China (19YJCZH244), Postdoctoral Special Funding Project, China (2019T120607).

Declarations

Conflict of interest The authors declare that there are no conflicts of interest regarding the publication of this paper.

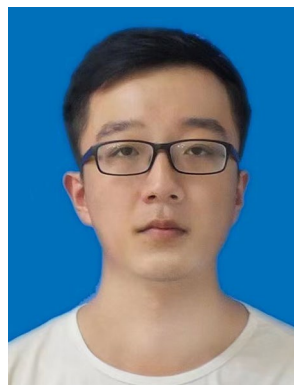
Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Cui, Z., Chen, W., & Chen, Y. (2021). Multi-scale attention convolutional neural network for time series classification. *Neural Networks*, 136, 126–140.
2. Sezer, O. B., Gudelek, U., & Ozbayoglu, M. (2020). Financial time series forecasting with deep learning: A systematic literature review: 2005–2019. *Applied Soft Computing*, 90, 106181.
3. Olsavszky, V., Dosius, M., Vladescu, C., & Benecke, J. (2020). Time series analysis and forecasting with automated machine learning on a National ICD-10 Database. *International Journal of Environmental Research and Public Health*, 17(14), 4979.
4. Hu, J., Wang, X., Zhang, Y., Zhang, D., Zhang, M., & Xue, J. (2020). Time series prediction method based on variant LSTM recurrent neural network. *Neural Processing Letters*, 52(2), 1485–1500.
5. Zhang, H., Wang, Z., & Liu, D. (2014). A comprehensive review of stability analysis of continuous-time recurrent neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 25(7), 1229–1262.
6. Weerakody, P., Wong, K. W., Wang, G., & Ela, W. (2021). A review of irregular time series data handling with gated recurrent neural networks. *Neurocomputing*, 441, 161–178.
7. Rizk, Y., & Awad, M. (2019). On extreme learning machines in sequential and time series prediction: A non-iterative and approximate training algorithm for recurrent neural networks. *Neurocomputing*, 325, 1–19.
8. Song, B., Li, K., Orellana-Martín, D., & Pérez-Jiménez, M. J. (2021). A survey of nature-inspired computing: Membrane computing. *ACM Computing Surveys*, 54(1), 1–31.

9. Song, B., Li, K., Orellana-Martín, D., Valencia-Cabrera, L., & Pérez-Jiménez, M. J. (2020). Cell-like P systems with evolutionary symport/antiport rules and membrane creation. *Information and Computation*, 275, 104542.
10. Pan, L., & Pérez-Jiménez, M. J. (2010). Computational complexity of tissue-like P systems. *Journal of Complexity*, 26(3), 296–315.
11. Song, B., Zeng, X., Jiang, M., & Perez-Jimenez, M. J. (2021). Monodirectional tissue P systems with promoters. *IEEE Transactions on Cybernetics*, 51(1), 438–450.
12. Lukosevicius, M., & Jaeger, H. (2009). Reservoir computing approaches to recurrent neural network training. *Computer Science Review*, 3(3), 127–149.
13. Jaeger, H. (2001). The “Echo State” approach to analysing and training recurrent neural networks. *überwachtes lernen*.
14. Bala, A., Ismail, I., Ibrahim, R., & Sait, S. M. (2018). Applications of metaheuristics in reservoir computing techniques: A review. *IEEE Access*, 6, 58012–58029.
15. Bo, Y.-C., Wang, P., & Zhang, X. (2020). An asynchronously deep reservoir computing for predicting chaotic time series. *Applied Soft Computing*, 95, 106530.
16. Lun, S., Yao, X., Qi, H., & Hu, H. (2015). A novel model of leaky integrator echo state network for time-series prediction. *Neurocomputing*, 159, 58–66.
17. PaulinCoulibaly. (2010). Reservoir computing approach to Great Lakes water level forecasting. *Journal of Hydrology*, 381(1–2), 76–88.
18. Bianchi, F. M., cardapane, S., Uncini, A., Rizzi, A., & Sadeghian, A. (2015). Prediction of telephone calls load using echo state network with exogenous variables. *Neural Networks*, 71, 204–213.
19. Cui, H., Liu, X., & Li, L. (2012). The architecture of dynamic reservoir in the echo state network. *Chaos*, 22(3), 033127.
20. Najibi, E., & Rostami, H. (2015). SCESN, SPESN, SWESN: Three recurrent neural echo state networks with clustered reservoirs for prediction of nonlinear and chaotic time series. *Applied Intelligence*, 43(2), 460–472.
21. Paun, G. (2000). Computing with membranes. *Journal of Computer and System Sciences*, 61, 108–143.
22. Peng, H., Wang, J., Shi, P., Pérez-Jiménez, M. J., & Riscos-Núez, A. (2017). Fault diagnosis of power systems using fuzzy tissue-like P systems. *Integrated Computer-Aided Engineering*, 24(4), 401–411.
23. Liu, X., Zhao, Y., & Sun, M. (2017). An improved apriori algorithm based on an evolution-communication tissue-like P system with promoters and inhibitors. *Discrete Dynamics in Nature and Society*, 2017(1), 1–11.
24. Song, B., Li, K., & Zeng, X. (2022). Monodirectional evolutionary symport tissue P systems with promoters and cell division. *IEEE Transactions on Parallel and Distributed Systems*, 33(2), 332–342.
25. Song, B., Hu, Y., Adorna, H. N., & Xu, F. (2018). A quick survey of tissue-like P systems. *Romanian Journal of Information Science and Technology*, 21, 310–321.
26. Song, B., Huang, S., & Zeng, X. (2021). The computational power of monodirectional tissue P systems with symport rules. *Information and Computation*, (1), 104751.
27. Song, B., & Pan, L. (2021). Rule synchronization for tissue P systems. *Information and Computation*, 281, 104685.
28. Pan, L., Alhazov, A., Su, H., & Song, B. (2020). Local synchronization on asynchronous tissue P systems with symport/antiport rules. *IEEE Transactions on Nanobioscience*, 19(2), 315–320.
29. Luo, Y., Zhao, Y., & Chen, C. (2021). Homeostasis tissue-like P systems. *IEEE Transactions on Nanobioscience*, 20(1), 126–136.
30. Song, Z. N., & Yang, L. J. (2022). Statistical inference for ARMA time series with moving average trend. *Journal of Nonparametric Statistics*, 34, 357–376.
31. Doucoure, B., Agbossou, K., & Cardenas, A. (2016). Time series prediction using artificial wavelet neural network and multi-resolution analysis: Application to wind speed data. *Renewable Energy*, 92, 202–211.
32. Smola, A. J., & Scholkopf, B. (2004). A tutorial on support vector regression. *Statistics and Computing*, 14, 199–222.
33. Hyndman, R. J., & Koehler, A. B. (2006). Another look at measures of forecast accuracy. *International Journal of Forecasting*, 22, 679–688.
34. Alpak, F. O., Araya-Polo, M., & Onyeagoro, K. (2019). Simplified dynamic modeling of faulted turbidite reservoirs: A deep-learning approach to recovery-factor forecasting for exploration. *SPE Reservoir Evaluation and Engineering*, 22, 1240–1255.
35. Schaffer, A. L., Dobbins, T. A., & Pearson, S. A. (2021). Interrupted time series analysis using autoregressive integrated moving average (ARIMA) models: A guide for evaluating large-scale health interventions. *BMC Medical Research Methodology*, 21, 1–12.
36. Lecun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86, 2278–2324.

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



Xiaojian Yang received the M.S. degree in management science and engineering from Shandong Normal University, Jinan, China, in 2022. His current research interests include machine learning, data mining, and non-stationary time series.



Qian Liu received the B.S. degree in engineering management from Business School at Shandong Normal University (SDNU), Jinan, PR China, in 2016. She is currently pursuing the M.S. degree in management science and engineering at SDNU. She has conducted research in the area of data mining and machine learning. Recently, she has published some journal papers.



Xiyu Liu received the Ph.D. degree in mathematical sciences from Shandong University, Jinan, China, in 1990. He is currently working as the Dean of the Academy of Management Science and Engineering, Shandong Normal University, Jinan, China. He has authored two books and more than 140 articles. His research interests include membrane computing, data mining, computational intelligence, and non-linear analysis. Dr. Liu was awarded the “Taishan Scholar”

of Management Science and Engineering, and the Vice President of the Computer Education Research Association of China Higher Normal Universities, and the Shandong Computer Society.



Jie Xue received the B.S. and Ph.D. degrees in management science and engineering from Shandong Normal University, Jinan, China, in 2010 and 2015, respectively. She is currently an Associate Professor with Business School, Shandong Normal University. Her current research interests include image processing, medical image analysis, and membrane computing. Dr. Xue won a National Visiting Scholar Program with the University of North Carolina from 2017 to 2018.