Chinese Journal of Electronics
vol. 33, no. 1, pp. 101–111, January 2024
https://doi.org/10.23919/cje.2022.00.292

RESEARCH ARTICLE

Echo State Network Based on Improved Knowledge Distillation for Edge Intelligence

Jian ZHOU[1,2], Yuwen JIANG[1,2], Lijie XU[1,2], Lu ZHAO[1,2], and Fu XIAO[1,2]

1. *College of Computer, Nanjing University of Posts and Telecommunications, Nanjing 210003, China*
2. *Jiangsu High Technology Research Key Laboratory for Wireless Sensor Networks, Nanjing 210003, China*

Corresponding author: Jian ZHOU, Email: zhoujian@njupt.edu.cn
Manuscript Received August 28, 2022; Accepted February 14, 2023

Abstract — Echo state network (ESN) as a novel artificial neural network has drawn much attention from time series prediction in edge intelligence. ESN is slightly insufficient in long-term memory, thereby impacting the prediction performance. It suffers from a higher computational overhead when deploying on edge devices. We firstly introduce the knowledge distillation into the reservoir structure optimization, and then propose the echo state network based on improved knowledge distillation (ESN-IKD) for edge intelligence to improve the prediction performance and reduce the computational overhead. The model of ESN-IKD is constructed with the classic ESN as a student network, the long and short-term memory network as a teacher network, and the ESN with double loop reservoir structure as an assistant network. The student network learns the long-term memory capability of the teacher network with the help of the assistant network. The training algorithm of ESN-IKD is proposed to correct the learning direction through the assistant network and eliminate the redundant knowledge through the iterative pruning. It can solve the problems of error learning and redundant learning in the traditional knowledge distillation process. Extensive experimental simulation shows that ESN-IKD has a good time series prediction performance in both long-term and short-term memory, and achieves a lower computational overhead.

Keywords — Echo state network, Reservoir structure optimization, Knowledge distillation, Edge intelligence, Time series prediction.

Citation — Jian ZHOU, Yuwen JIANG, Lijie XU, *et al.*, "Echo State Network Based on Improved Knowledge Distillation for Edge Intelligence," *Chinese Journal of Electronics*, vol. 33, no. 1, pp. 101–111, 2024. doi: 10.23919/cje.2022.00.292.

I. Introduction

With the rapid development of artificial intelligence (AI) and edge computing, edge intelligence emerges as time require [1]. Deploying time series prediction models on edge devices closer to related data can improve the real-time performance of time series prediction. However, due to the limited resources (e.g., computing and storage resources) of edge devices [2], [3], it is necessary to reduce the computational overhead incurred by time series prediction models as much as possible. Therefore, it is still a great challenge to implement a time series prediction model with high prediction performance and low computational overhead on edge devices.

Echo state network (ESN) [4], as a new recurrent neural network (RNN) [5], was firstly proposed by Jaeger to solve the problems of vanishing gradient and exploding gradient [6] during the training of traditional RNN. The hidden layer of ESN, also known as the reservoir, is a randomly generated sparse network with many neurons. The reservoir contains the network state and is affected not only by the current moment, but also by the past moment, thus it has the capability for short-term memory. The sparse structure of reservoir ensures the gradient stability during the training process. Compared to the traditional RNN, ESN only requires training the output weight, with a simple linear regression. Due to its great capability of processing non-linear feature, great short-term memory capability and relatively low training overhead, ESN stands out when performing a series of tasks including speech recognition [7], automated control [8] and so on, especially in time series prediction [9], [10]. However, the reservoir structure in the classic ESN is randomly generated. Although the reservoir in the

Associate Editor: Prof. Chi LIU, Beijing Institute of Technology.

classic ESN has a great short-term memory capability, its capability for long-term memory is slightly insufficient, which limits its capability to extract long-term correlation features in time series. Moreover, there are redundant connections of neurons in the randomly generated reservoir structure, increasing the computational overhead. We focus on the reservoir structure optimization of ESN through knowledge transfer to solve the problems of insufficient capability for long-term memory and redundant connections of neurons in the reservoir, so as to improve the prediction performance and reduce the computational overhead of ESN.

Knowledge distillation proposed by Hinton [11] can realize the knowledge transfer from the teacher network to the student network. So far, knowledge distillation has achieved quite excellent performance in computer vision, natural language processing and other fields [12]. Depending on the types of transferred knowledge, the related works of knowledge distillation can be roughly divided into three directions. Some researchers [13] took the output of the teacher network as the knowledge and distilled it to student network for target training. Another part of researchers [14] regarded the learned features of teacher network as the knowledge and transferred them from teacher network to student network. The last part of researchers [15] distilled the relationship between network layers of teacher network as the knowledge that is to be learned by student network. All the knowledge distillation methods mentioned above achieve the goal of transfer learning and improve the performance of student network. However, to the best of our knowledge, these methods are limited to classification tasks and rarely applied to regression tasks, e.g., time series prediction. In this paper, knowledge distillation is firstly introduced into the reservoir structure optimization, and traditional knowledge distillation is improved to solve the problems of error learning and redundant learning. Specifically, the output of the teacher network is distilled for ESN to optimize the reservoir structure, aiming to achieve better capabilities for long-term and short-term memory and simplify the structure further.

In order to improve the prediction performance and reduce the computational overhead, we propose an echo state network based on improved knowledge distillation (ESN-IKD) for edge intelligence. The contributions can be concluded as the following two aspects.

1) We firstly introduce knowledge distillation into reservoir structure optimization and propose ESN-IKD for edge intelligence. First, the model of ESN-IKD is constructed. In particular, with the help of ESN with double loop reservoir structure (ESN-DLRS), the classic ESN learns the long-term memory capability of long and short-term memory network (LSTM). Second, the training algorithm of ESN-IKD is proposed. In particular, the error learning is alleviated by the assistant network and the redundant learning is alleviated by the iterative pruning. The optimized reservoir structure enhances the long-term memory capability and prunes the redundant neurons.

2) We carry out a simulation analysis by applying ESN-IKD in three datasets of typical time series. The simulation results show that ESN-IKD has better prediction performance and lower computational overhead compared with the teacher network and the ESNs with other optimized reservoir structures, so it is more suitable for deploying on edge devices.

The rest of this paper is organized as follows. Section II introduces the related work. Section III proposes ESN-IKD. Section IV simulates and analyzes ESN-IKD. Finally, Section V summarizes the whole work.

II. Related Work

With the gradual popularization of edge devices, edge intelligence has received extensive attention. Deng *et al.* [16] divided edge intelligence into AI for edge and AI on edge. The former focuses on using the effective technology of AI to provide better solutions to key problems in edge computing, while the latter studies how to deploy the AI models on edge devices. In the field of AI for edge, Shen *et al.* [17] proposed a power control method based on graph neural networks to achieve optimal management of energy consumption. Wang *et al.* [18] proposed a pricing prediction algorithm which can satisfy the service requirements of users by effectively utilizing edge computing resources. Chen *et al.* [19] proposed a load signature construction method to improve the recognition performance in load recognition task. In the field of AI on edge, Thakker *et al.* [20] proposed an implementation of compressing RNN cell to achieve a balance between inferencing accuracy and resource consumption. Ma *et al.* [21] proposed a truthful combinatorial double auction mechanism to guarantee truthfulness and budget-balance under locality constraints of mobile edge computing. Peng *et al.* [22] summarized the field-programmable gate array-based custom computing architecture for convolutional neural network to achieve a better appliance on edge devices. AI on edge has attracted more attention from academia and industry because of its practicality. Based on the new computing mode of edge computing, edge intelligence can deploy AI models on the edge devices closer to the data source, which greatly reduces the delay cost of AI applications. Although edge intelligence has been developed to a certain extent, how to further optimize the performance and reduce the cost of edge intelligence is still an important problem to be solved urgently. The fundamental challenge is the conflict between the huge computational overhead required by AI models and the limited computational resources of edge devices [23]. Network structure optimization can adjust the network structure of AI model to improve the model performance and reduce the model complexity, so as to provide better edge computing services on resource-constrained edge devices. Nowadays, network structure optimization is becoming an important research direc-

tion of edge intelligence.

The reservoir, as the core of the ESN, plays a crucial role in the performance of the entire network. In recent years, many researchers have studied the optimization of reservoir structure. The related work is mainly divided into the fixed reservoir, the growth reservoir, and the pruned reservoir [24]. In the field of the fixed reservoir, Rodan *et al.* [25] proposed a simple cycle reservoir (SCR), which organizes the neuros in the reservoir into a cycle and keeps the connection weights between the neurons in the reservoir identical. The simulation results show that SCR reduces the computational overhead of ESN. Based on SCR, an adjacent-feedback loop reservoir (ALR) was proposed by Sun *et al.* [26]. Adjacent neurons in ALR cooperate with each other through feedback connections, making ALR achieve better prediction performance than SCR. Zhou *et al.* [27] organized the neuros in the reservoir into a double loop structure, and then proposed ESN-DLRS. Compared with SCR, ESN-DLRS strengthens the processing capability for temporal features. In the above-mentioned studies, the fixed reservoir reduces the randomness and computational overhead in the forward propagation process of ESN, but also weakens the flexibility and generalization capability of ESN. In the field of the growth reservoir, Qiao *et al.* [28] divided the reservoir into multiple sub-modules and simplified the reservoir by gradually adding sub-modules. Kawai *et al.* [29] introduced the small-world topology into the reservoir and proposed an echo state network based on improved small-world (ESN-ISW) to realize sparse connection by gradually increasing connections between neurons in the reservoir. In the above-mentioned studies, the growth reservoir has a simple structure and low computational overhead, but it is difficult to find a suitable stop-adding criterion. In the field of the pruned reservoir, Wang *et al.* [30] proposed a sensitive iterative pruning algorithm, which simplifies SCR by pruning the neurons with low sensitivity. Scardapane *et al.* [31] proposed an effective criterion for pruning the connection in the reservoir, which guides the entire pruning process based on the state correlation between neurons. Li *et al.* [32] proposed a contribution-based pruning algorithm, which defines the contribution of each neuron through mutual information. This algorithm prunes the neurons with low contribution and finally obtains an echo state network based on contribution (ESN-C). Wang *et al.* [33] proposed an echo state network optimized by the bias dropout algorithm (ESN-BD), which has a simpler reservoir. In ESN-BD, the neurons with low activation values are divided into different contribution groups with different pruning probabilities. In the above-mentioned studies, the pruned reservoir removes the redundant neurons through defining the pruning criteria to simplify the ESN, but the performance improvement of ESN is limited. Among the related works mentioned above, although the computational overhead of ESN has been reduced to a certain extent, the problem of insufficient capability for long-term memory of the reservoir is still ignored. Therefore, it is necessary to design an optimization method for reservoir structure to realize the reservoir equipped with great capability for long-term memory and simplified structure. This paper focuses on how to realize the knowledge transfer of long-term memory capability, and how to integrate the advantages of the fixed reservoir and the pruned reservoir, aiming to enhance the long-term memory capability and prune the redundant neuron.

III. Design of Echo State Network Based on Improved Knowledge Distillation

In this work, regarding LSTM as the teacher network, the classic ESN as the student network, and ESN-DLRS as the assistant network, we propose ESN-IKD for edge intelligence. LSTM has good capability for long-term memory and performs well in dealing with time series problems with obvious long-term correlation [34]. Meanwhile, ESN-DLRS has good capability to extract nonlinear and temporal feature in time series. We use knowledge distillation to optimize the reservoir structure of ESN, so as to complete the knowledge transfer of long-term memory capability from LSTM to ESN and reduce the computational overhead.

Just like the teacher-student mode in real life, if the teacher network teaches wrong knowledge, or the student network deviates from the correct learning direction, the student network as the learning subject may be not able to capture the correct knowledge. Therefore, we add assistant network into the framework of traditional knowledge distillation and use ESN-DLRS as the assistant network for the teaching process of LSTM to reduce the adverse effects of error learning. The nonlinear and temporal features extracted by ESN-DLRS will also be used as tacit knowledge to guide the structure optimization process of the student network.

Besides, the problem of redundant learning is still showing in the student network of traditional knowledge distillation. Redundant learning can easily complicate the reservoir structure and the dynamic characteristic of the reservoir may be damaged. In this work, the redundant neurons generated during the learning process are iteratively pruned, which can not only alleviate the adverse effects of redundant learning, but also reduce the computational overhead of ESN.

ESN-IKD is applied to time series prediction. Let us assume that the dataset contains p true values sorted by time, denoted as $T_r = \{\mathrm{tr}(1), \mathrm{tr}(2), \ldots, \mathrm{tr}(t), \ldots, \mathrm{tr}(p)\}$, where $\mathrm{tr}(t)$ represents the true value at time t. If time t is taken as the current moment, the prediction task can be described as calculating the predicted value $\mathrm{tr}(t+1)$ of the next moment by inputting historical sequence $\mathrm{Tr}_K(t)$.Inparticular,$\mathrm{Tr}_K(t) = \{\mathrm{tr}(t-K+1), \mathrm{tr}(t-K+2), \ldots, \mathrm{tr}(t)\}$, where K is the size of sliding window. We construct the training set $U = \{(u_{\mathrm{train}}(t), y_{\mathrm{train}}(t+1)), K \leq t \leq p-1\}$, where $u_{\mathrm{train}}(t)$ is $\mathrm{Tr}_K(t)$ and $y_{\mathrm{train}}(t+1)$ is

tr$(t+1)$. The training set is further divided into teacher training set U_1, assistant training set U_2, and student training set U_3.

1. ESN-IKD model

Figure 1 shows the ESN-IKD model constructed in this work. In this model, the teacher network (LSTM) is responsible for teaching the long-term memory capability. It consists of the input layer, the hidden layer, and the output layer. The $u_{\text{train}}(t)$ in U_2 is fed into the trained LSTM, then LSTM calculates $y_{\text{train}}^{\text{teacher}}(t+1)$ by forward propagation and outputs the distillation knowledge $T_1 = \left\{\left(u_{\text{train}}(t), y_{\text{train}}^{\text{teacher}}(t+1)\right)\right\}$ to teach the assistant network. Similarly, the $u_{\text{train}}(t)$ in U_3 is fed into the trained LSTM, then LSTM calculates $y_{\text{train}}^{\text{teacherNew}}(t+1)$ by forward propagation and outputs the distillation knowledge $T_2 = \{(u_{\text{train}}(t), y_{\text{train}}^{\text{teacherNew}}(t+1))\}$ to teach the student network. The assistant network (ESN-DLRS) is responsible for supervising the distillation learning direction. It consists of the input layer, the double loop reservoir, and the output layer. The $u_{\text{train}}(t)$ in U_3 is fed into the trained ESN-DLRS, then ESN-DLRS calculates $y_{\text{train}}^{\text{assitantNew}}(t+1)$ by forward propagation and outputs the assistant knowledge $A = \left\{\left(u_{\text{train}}(t), y_{\text{train}}^{\text{assitantNew}}(t+1)\right)\right\}$ to teach the student network. As the principal part of the model, the student network (the classic ESN) is ultimately responsible for time series prediction. It consists of the input layer, the reservoir, and the output layer. The classic ESN is trained according to the algorithm in next section to optimize the reservoir structure. After the training, the historical sequence of the current moment is input, and the optimized student network performs forward propagation calculation to obtain the predicted value of the next moment.

The forward propagation process of the student network is as follows. Assume that the current moment is t

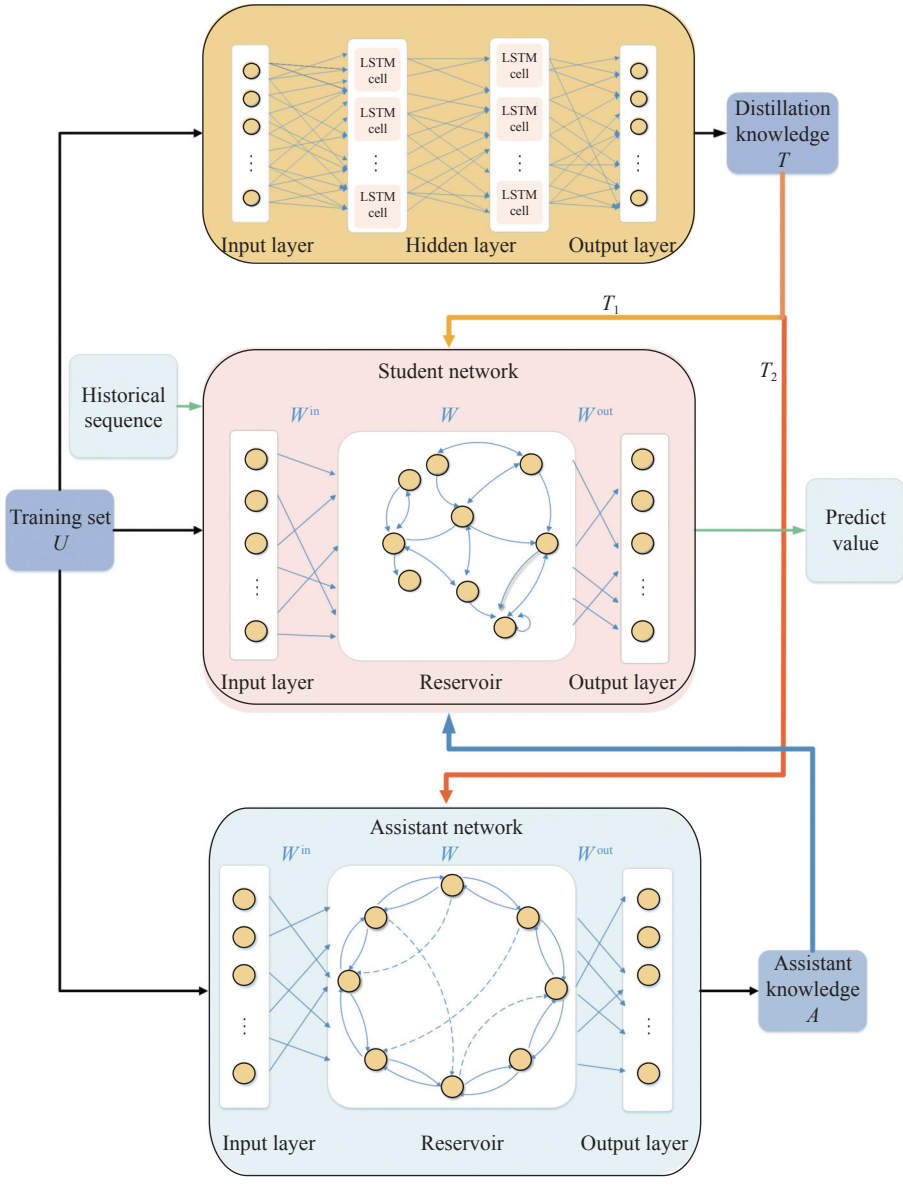

Figure 1 The model of ESN-IKD.

and the input vector is $u(t)$, the state vector $x(t)$ of the reservoir can be updated by

$$x(t) = f^{\text{in}}\left(W^{\text{in}}u(t) + Wx(t-1)\right) \tag{1}$$

where f^{in} is the reservoir activation function, W^{in} is the randomly generated input weight matrix, and W is the reservoir weight matrix. In the classic ESN, W is generated randomly and composed of a large number of zero-value elements, which makes the structure sparse. In ESN-DLRS, W is generated by the double loop construction algorithm [27]. In this work, the reservoir weight matrix W is optimized by the training algorithm described in next section.

The output $\hat{y}(t)$ of ESN is calculated by

$$\hat{y}(t) = f^{\text{out}}\left(W^{\text{out}}[u(t), x(t)]\right) \tag{2}$$

where f^{out} is the output activation function, and W^{out} is the output weight matrix obtained by the ridge regression [35].

The forward propagation process of LSTM can refer to [36]. The forward propagation process of ESN-DLRS is basically the same as that of the classic ESN, which is not discussed repeatedly.

2. ESN-IKD training

We propose the training algorithm of ESN-IKD in Algorithm 1, and the specific steps are as follows.

At the beginning, the parameters of networks including the teacher network (LSTM), the student network (the classic ESN), and the assistant network (ESN-DLRS) are set, then each part of ESN-IKD is initialized.

Teacher network training Depending on U_1, LSTM is trained by the adaptive moment estimation gradient descent algorithm [36]. The training objective function of LSTM is represented by

$$\min \sum_{t=0}^{m} \left(y(t) - \hat{y}^{\text{teacher}}(t)\right)^2 / m \tag{3}$$

where $y(t)$ is the true value, $\hat{y}^{\text{teacher}}(t)$ is the predicted value of LSTM, and m is the size of the training set.

Afterwards, the $u_{\text{train}}(t)$ in U_2 is fed to the trained LSTM to output the distillation knowledge T_1 of LSTM.

Assistant network training Depending on U_2 and T_1, ESN-DLRS is trained by the ridge regression. The training objective function of ESN-DLRS is similar to that of LSTM. The target value in the training process is obtained by the fusion of the true value and the distillation knowledge T_1. It can be calculated by

$$y_{\text{train}}^{\text{assistant}}(t+1) = \alpha \cdot y_{\text{train}}(t+1) + \beta \cdot y_{\text{train}}^{\text{teacher}}(t+1) \tag{4}$$

where α and β are the assistant distillation learning rates, and α+β=1.

Afterwards, the $u_{\text{train}}(t)$ in U_3 is fed to the trained ESN-DLRS to output the assistant knowledge A of ESN-DLRS. Meanwhile, the $u_{\text{train}}(t)$ in U_3 is fed to the trained LSTM to output the distillation knowledge T_2 of LSTM.

Algorithm 1 Training algorithm of ESN-IKD

Require:
U: Training set.
Ensure:
W: The reservoir weight matrix.
W^{out}: The output weight matrix.
Initialize ESN-IKD;
for u in U_1: //Train the teacher network
  Input u into LSTM, calculate $\hat{y}^{\text{teacher}}(t)$;
  Train LSTM by equation (3);
for u in U_2: //Obtain the distillation knowledge T_1
  Input u into the trained LSTM, calculate $y_{\text{train}}^{\text{teacher}}(t+1)$;
for u in U_2 and T_1: //Train the assistant network
  Calculate $y_{\text{train}}^{\text{assistant}}(t+1)$ by equation (4);
  Train ESN-DLRS similarly by equation (3);
for u in U_3: //Obtain the assistant knowledge A and the distillation knowledge T_2
  Input u into the trained ESN-DLRS, calculate $y_{\text{train}}^{\text{assistantNew}}(t+1)$;
  Input u into the trained LSTM, calculate $y_{\text{train}}^{\text{teacherNew}}(t+1)$;
for u in U_3: //Train the student network
  Input u into the classic ESN, calculate $\hat{y}^{\text{student}}(t)$;
  Train the classcial ESN by equation (5) firstly;
for u in A and T_2: //Obtain the training set U_3'
  Calculate $y_{\text{train}}^{\text{student}}(t+1)$ by equation (6);
for u in U_3': //Transfer knowledge
  Calculate W by equation (8);
for neuron i in [0, N − 1]: //Prune iteratively
  Prune neuron i;
  if the objective function value decreases, then
    continue
  else
    undo pruning
for u in U_3: //Obtain the weight martrix W^{out}
  Input u into the trained classic ESN, calculate $\hat{y}^{\text{student}}(t)$;
  Train the classic ESN by equation (5) secondly.

Student network training Depending on U_3, the classic ESN is trained by the ridge regression. Thus, the output weight matrix W^{out} is obtained. The training objective function of the classic ESN is represented by

$$\min \sum_{t=0}^{m} \left(y(t) - \hat{y}^{\text{student}}(t)\right)^2 / m \tag{5}$$

where $\hat{y}^{\text{student}}(t)$ is the predicted value of the classic ESN.

Knowledge transferring Depending on T_2 and A, the long-term memory capability of the reservoir of the classic ESN is enhanced. For the classic ESN, construct the training set $U_3' = \left\{\left(u_{\text{train}}(t), y_{\text{train}}^{\text{student}}(t+1)\right)\right\}$, where $y_{\text{train}}^{\text{student}}(t+1)$ is the weighted average sum of the true

value $y_{\text{train}}(t+1)$, the predicted value of the teacher network $y_{\text{train}}^{\text{teacherNew}}(t+1)$, and the predicted value of the assistant network $y_{\text{train}}^{\text{assitantNew}}(t+1)$. $y_{\text{train}}^{\text{student}}(t+1)$ is calculated by

$$\begin{aligned} y_{\text{train}}^{\text{student}}(t+1) =& \mu \cdot y_{\text{train}}(t+1) + v \cdot y_{\text{train}}^{\text{teacherNew}}(t+1) \\ &+ o \cdot y_{\text{train}}^{\text{assitantNew}}(t+1) \end{aligned} \tag{6}$$

where μ, ν, o are the student distillation learning rates, and $\mu + v + o = 1$.

Depending on U_3', it can be deduced from (2) that

$$x(t) = \left(W^{\text{out}}\right)^{-1} f^{\text{arcout}}\left(y_{\text{train}}^{\text{student}}(t+1)\right) \tag{7}$$

where f^{arcout} is the inverse output activation function. Substitute the above equation into (1), then W can be calculated by

$$W = \left[f^{\text{arcin}}(x(t)) - W^{\text{in}} u_{\text{train}}(t)\right] / x(t-1) \tag{8}$$

where f^{arcin} is the inverse reservoir activation function. To preserve the echo property of the reservoir, W is scaled by

$$W = sr \cdot W / \lambda_{\max}(W) \tag{9}$$

where sr is the spectral radius and $\lambda_{\max}(W)$ is the maximum eigenvalue of W.

Iterative pruning Depending on U_3, the redundant knowledge of the student network is pruned to reduce the impact of redundant learning. Equation (5) is used as the evaluation criterion. Take neuron i in the reservoir as an example. If the objective function value decreases after pruning neuron i, that is, the prediction performance improves, then neuron i is pruned and the connections between other neurons and neuron i are eliminated. Specifically, let $w_{i,j} = w_{j,i} = 0$, where $w_{i,j}$ represents the connection weight between neuron i and neuron j, $0 \le j \le N-1$. If the objective function value increases or remains unchanged, there is no pruning.

Afterwards, iterative pruning is repeated until the predefined pruning iteration times is reached. Then, the weight matrix W of the reservoir is obtained.

Finally, depending on U_3, the optimized student network is optimized by the ridge regression. Thus, the final output weight matrix W^{out} is obtained.

IV. Simulation Analysis

Referring to [27] and [36], Table 1 shows the parameter settings of ESN-IKD. The double loop interval parameter D of ESN-DLRS is set to 10 and other parameters of ESN-DLRS are consistent with that of the classic ESN. Both α and β are set to 0.5. μ is set to 0.5, ν is set to 0.3, and o is set to 0.2. K is set to 3, and the number of pruning iterations is set to 50.

The normalized root mean square error (NRMSE) is used to evaluate the prediction performance, the smaller the better. NRMSE can be calculated by

Table 1 Parameter settings of ESN-IKD

Model	Parameter	Value
Teacher network (LSTM)	Input layer size	3
	Hidden layers number	2
	Neurons in first layer	512
	Neurons in second layer	128
	Output layer size	1
	Hidden activation	Relu
	Output activation	Linear
	Learning rate	0.001
Student network (the classic ESN)	Input layer size	3
	Reservoir size	50
	Output layer size	1
	Spectral radius	0.2
	Reservoir activation	Tanh
	Output activation	Identity
	Learning rate	0.01

$$NRMSE = \sqrt{\left(\sum_{i=1}^{n} \left(\hat{y}_{\text{test}}(i) - y_{\text{test}}(i)\right)^2\right) / n \cdot \sigma^2} \tag{10}$$

where n is the number of samples, $\hat{y}_{\text{test}}(i)$ is the predicted value, $y_{\text{test}}(i)$ is the true value, and σ^2 represents the variance.

The multiply-accumulate operation (MACC) [37] is used to evaluate the computational overhead, the smaller the better. One calculation of a neuron in the forward propagation process is noted as 1 MACC.

1. Dataset description

Three typical time series datasets, including Mackey-Glass chaotic time series (MG) [4], Intel Berkeley Research Lab dataset (IBRL) [38] and Beijing University of Posts and Telecommunications Academic Network dataset (BUPTAN) [39], are used in our simulations.

MG represents the random motion that occurs in the determined system, which contains rich dynamic information and can be generated by

$$\mathrm{d}s(t)/\mathrm{d}t = 0.2s(t-\tau) / \left(1 + s^{10}(t-\tau)\right) - 0.1s(t) \tag{11}$$

where $s(t)$ is the true value at time t and τ is the delay time. In particular, τ is set to 17 [4]. Parts of the sample data of MG are shown in Figure 2(a).

IBRL is collected from 54 sensor nodes scattered in Intel Berkeley lab from February 28, 2004 to April 5, 2004. Temperature data of node 1 is selected as the simulation data in this work. Parts of the sample data are shown in Figure 2(b) (The ordinate is the normalized temperature value every 31 s). BUPTAN records the number of packets per minute from September 24, 2011 to September 30, 2011, in the backbone nodes of the academic network of

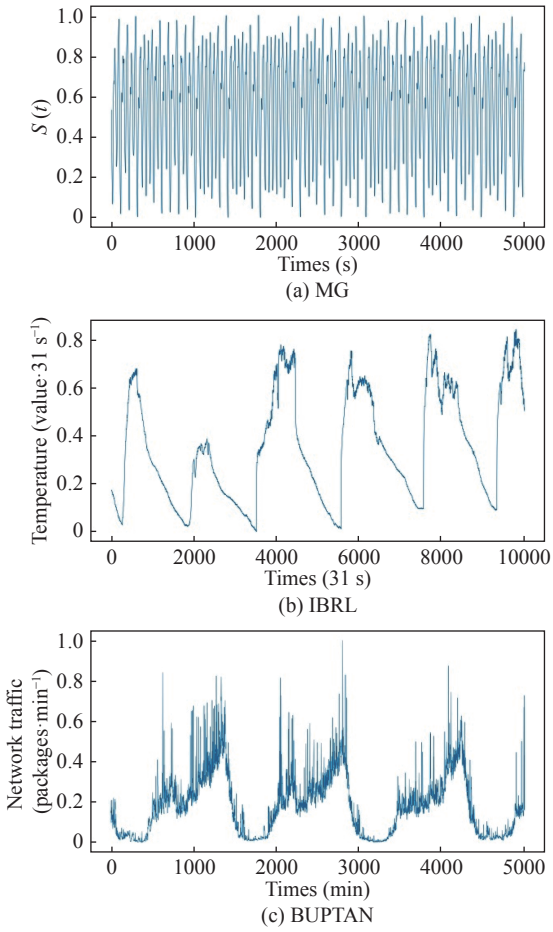

Figure 2 Parts of the sample data of time series datasets. (a) Mackey-Glass chaotic time series (MG); (b) Intel Berkeley Research Lab dataset (IBRL); (c) Beijing University of Posts and Telecommunications Academic Network data-set (BUPTAN).

Beijing University of Posts and Telecommunications. Parts of the sample data are shown in Figure 2(c).

As shown in Figure 2, the data generated by the equation in MG has the characteristics of drastic changes, small periodic intervals, and obvious short-term correlations. Unlike MG, IBRL and BUPTAN are collected in real world. The data in IBRL has the characteristics of relatively gentle changes, long periodic intervals, and obvious long-term correlations. The data in BUPTAN changes more complicatedly. It has the characteristics of gentle changes at the troughs and drastic changes at the peaks, showing a relatively balanced long-term correlation and short-term correlation. Considering the periodicity of each dataset, the first 3,000 data in MG, the first 10,000 data in IBRL and the first 10,000 data in BUPTAN are taken as the simulation data. The simulation data is divided into training set and test set according to 8:2, in which the training set is further equally divided into teacher training set, assistant training set and student training set. All data is normalized by the MIN-MAX algorithm [40].

2. Prediction performance analysis

To verify the effectiveness of optimizing the reservoir structure based on improved knowledge distillation, we compare ESN-IKD with the student network (the classic ESN), the assistant network (ESN-DLRS), and the teacher network (LSTM), repesctively. Figure 3 shows the average NRMSE of the above-mentioned 4 models over 20 experiments on each dataset.

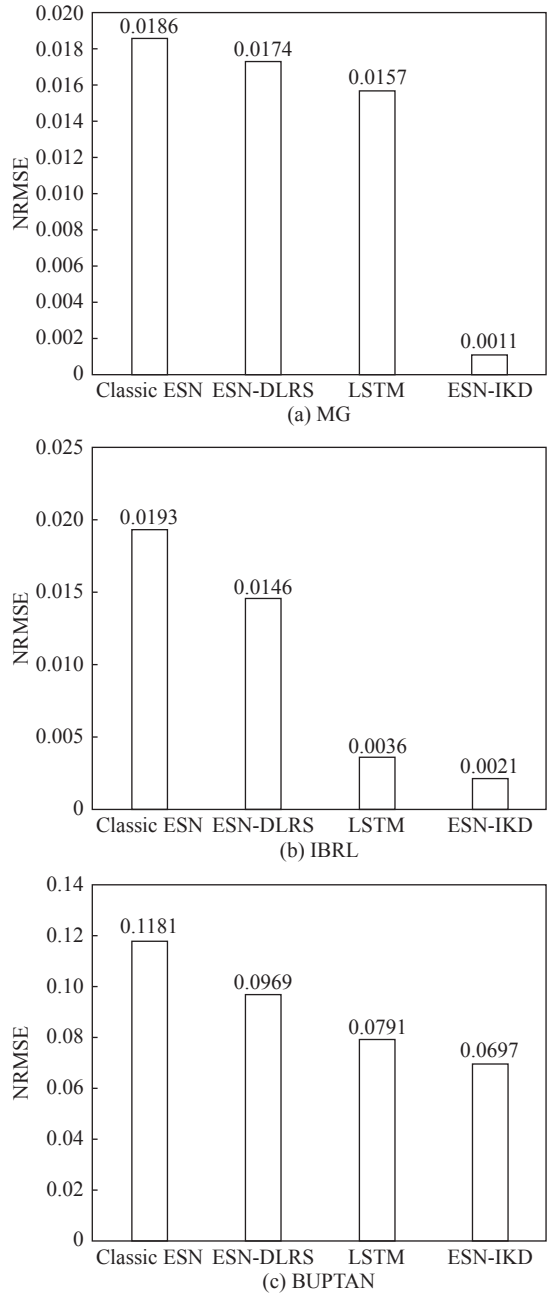

Figure 3 Comparison of prediction errors of the classic ESN, ESN-DLRS, LSTM and ESN-IKD models on datesets (a) MG, (b) IBRL, and (c) BUPTAN.

As shown in Figure 3, the prediction error of ESN-DLRS is lower than that of the classic ESN on all the three datasets. The reason is that the double reservoir structure has better temporal processing capability. On IBRL with obvious long-term correlation features, the prediction error of LSTM is significantly lower than that

of the classic ESN and ESN-DLRS. That is because the long-term memory capability of LSTM is better than that of the classic ESN and ESN-DLRS. On MG with obvious short-term correlation features and BUPTAN with balanced long-term correlation features and short-term correlation features, the prediction error of LSTM is slightly lower than that of the classic ESN and ESN-DLRS. That is because the classic ESN and ESN-DLRS have great capability for short-term memory, making them extract short-term correlations efficiently, while LSTM uses multiple hidden layers to capture the deep features, and thus compared with the classic ESN or ESN-DLRS with single hidden layer, its feature extraction capability is better. ESN-IKD achieves the lowest prediction error on all the three datasets. Across all cases, the average advantages of ESN-IKD are 17.8% over ESN, 14.5% over ESN-DLRS, and 8.6% over LSTM. That is because ESN-IKD can improve the capability of extracting long-term correlation features after learning, and thus its reservoir has good capabilities for both long-term memory and short-term memory. In summary, the optimization of the reservoir structure based on improved knowledge distillation can enhance the long-term memory capability of the reservoir.

To verify the prediction performance of ESN-IKD, we compare ESN-IKD with the ESNs with other optimized reservoir structures, such as ESN-C [32], ALR [26], ESN-ISW [29] and ESN-BD [33]. Figure 4 shows the average NRMSE of 20 experiments on each dataset for the classic ESN, ESN-C, ALR, ESN-ISW, ESN-BD and ESN-IKD.

As shown in Figure 4, the prediction errors of ESN-C, ALR, ESN-ISW, ESN-BD and ESN-IKD are all lower than that of the classic ESN, which shows that optimizing the reservoir structure can improve the prediction performance of ESN. On the three datasets, we compare the performance of the pruned reservoir ESN-C and ESN-BD, the fixed reservoir ALR and the growth reservoir ESN-ISW. It can be seen from the figure that ESN-C has the worst prediction performance, ESN-BD has the best prediction performance. ALR and ESN-ISW have the prediction performance between the former two. That is because ESN-C aims to reduce the computational overhead only, and yet the different contribution groups of ESN-BD can make the optimization of reservoir more instructive. In addition, the small-world topology of ESN-ISW might generate complex reservoir structure. ALR strengthens the capability of processing temporal features through the fixed reservoir structure, but also weakens the flexibility to a certain extent. On all the three datasets, ESN-IKD outperforms ESNs with other optimized reservoir structures. Its average advantages are 15.3% over ESN-C, 13.2% over ALR, 10.1% over ESN-ISW, and 5.6% over ESN-BD. The reason is that the reservoir of ESN-IKD not only learns the long-term memory capability of LSTM, but also learns the temporal processing capability of ESN-DLRS with the fixed reservoir structure. The assistant network effectively reduces the impact of error learning in traditional knowledge distillation. At the same time, iterative pruning removes redundant neurons, reducing the impact of redundant learning. In summary, ESN-IKD integrates the advantages of the fixed reservoir and the pruned reservoir, so it has better prediction performance than ESNs with other optimized reservoir structures.

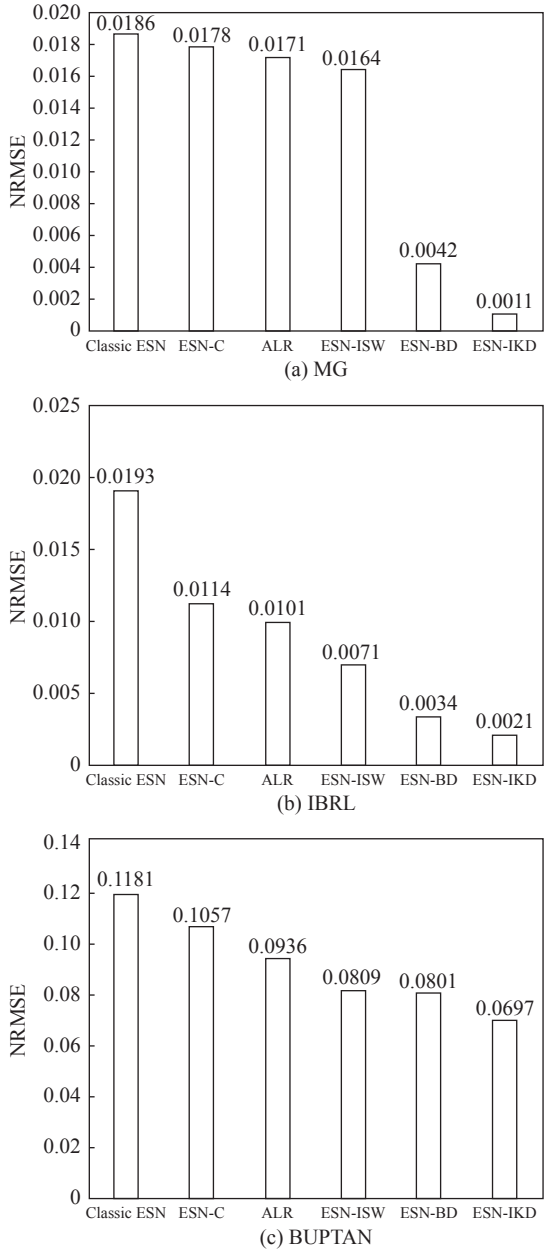

Figure 4 Comparison of prediction errors of the classic ESN, ESN-C, ALR, ESN-ISW, ESN-BD and ESN-IKD on datesets (a) MG, (b) IBRL, and (c) BUPTAN.

3. Computational overhead analysis

To verify the effectiveness of ESN-IKD in reducing the computational overhead of the reservoir, we compare ESN-IKD with ESN-C [32] and ESN-BD [33] which are the representatives of the pruned reservoir. Figure 5

shows the average number of pruned neurons of 20 experiments on each dataset for ESN-C, ESN-BD and ESN-IKD.

As shown in Figure 5, ESN-IKD has a larger number of pruned neurons than ESN-C and ESN-BD on MG and BUPTAN, while ESN-C and ESN-BD pruned more neurons on IBRL. That is because ESN-IKD is less affected by redundant learning during the knowledge distillation on IBRL and more significantly affected on MG and BUPTAN, making its reservoir have more redundant neurons. On the three datasets, the average advantages of ESN-IKD are 3.6% over ESN-C and 2.2% over ESN-BD.

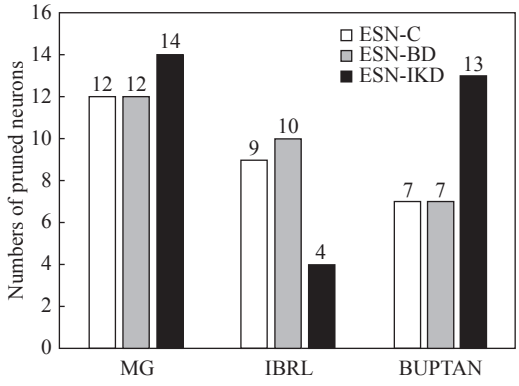

Figure 5 Comparison of numbers of pruned neurons of ESN-C, ESN-BD and ESN-IKD.

To verify the advantages of ESN-IKD in terms of computational overhead, we calculate the MACCs of the classic ESN, ESN-C, ESN-BD and ESN-IKD, as shown in Table 2. LSTM has a more complex network structure and uses two hidden layers, which makes it contain more neurons, so the MACC of LSTM far exceeds other models. ESN-DLRS, ALR and ESN-ISW do not prune neurons in the reservoir, and thus their MACCs are consistent with that of the classic ESN. Therefore, we do not discuss the MACCs of LSTM, ESN-DLRS, ALR and ESN-ISW.

Table 2 MACCs of the classic ESN, ESN-C, ESN-BD and ESN-IKD

Model	MG	IBRL	BUPTAN
Classic ESN	2754	2754	2754
ESN-C	1638	1890	2068
ESN-BD	1638	1804	2068
ESN-IKD	1480	2350	1558

As shown in Table 2, compared with the classic ESN, ESN-C, ESN-BD and ESN-IKD have lower MACCs. The reason is that all these three models prune redundant neurons in the reservoir, reducing computational overhead. The MACCs of ESN-IKD is lower than that of ESN-C and ESN-BD on MG and BUPTAN, and higher than that of ESN-C and ESN-BD on IBRL. That is because ESN-IKD prunes more redundant neurons on MG and BUPTAN. On the three datasets, the MACCs of ESN-IKD decreases 1.2% than ESN-C and 0.8% than ESN-BD on average. Compared with the classic ESN, the decreasing value can reach 11.6%. In summary, compared with the teacher network (LSTM), ESN-IKD adopts a single reservoir and has a simplified structure. Compared with ESNs with other optimized reservoir structures, ESN-IKD removes more redundant neurons in the reservoir, and thus reduces the computational overhead, making it more suitable for deploying on edge devices.

V. Conclusion

Edge intelligence is of great significance to time series prediction. In this paper, we proposed the echo state network based on improved knowledge distillation (ESN-IKD) for edge intelligence, aiming to improve the prediction performance while reducing the computational overhead. In ESN-IKD, the classic ESN learns the long-term memory capability of LSTM with the help of the ESN-DLRS. ESN-IKD reduces the adverse effects of error learning and redundant learning through the assistant network and iterative pruning. Finally, we conducted comprehensive experiments on three typical time series prediction tasks to evaluate the ESN-IKD. The experimental results show that ESN-IKD is superior to the teacher network and the ESNs with other optimized reservoir structures in terms of both prediction performance and computational overhead. After the optimization of the reservoir structure, ESN-IKD is more suitable for deploying on edge devices. The training of ESN-IKD is a little complex, and our future work will focus on how to optimize the training of ESN-IKD to further reduce the training overhead.

Acknowledgement

This work was supported by the National Natural Science Foundation of China (Grant Nos. 61972210, 6180 2206, and 61803212), the Science and Technology Planning Project of Jiangsu Province (Grant No. BE2020729) and the 1311 Talent Program of Nanjing University of Posts and Telecommunications.

References

[1] X. F. Wang, Y. W. Han, V. C. M. Leung, *et al.*, "Convergence of edge computing and deep learning: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 2, pp. 869–904, 2020.

[2] R. Gu, Y. Q. Chen, S. Liu, *et al.*, "Liquid: Intelligent resource estimation and network-efficient scheduling for deep learning jobs on distributed GPU clusters," *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 11, pp. 2808–2820, 2022.

[3] T. Wang, Y. Li, W. W. Fang, *et al.*, "A comprehensive trustworthy data collection approach in sensor-cloud systems," *IEEE Transactions on Big Data*, vol. 8, no. 1, pp. 140–151, 2022.

[4] H. Jaeger, "Reservoir riddles: Suggestions for echo state network research," in *Proceedings of 2005 IEEE International Joint Conference on Neural Networks*, Montreal, QC, Canada, pp. 1460–1462, 2005.

[5] H. G. Zhang, Z. S. Wang, and D. R. Liu, "A comprehensive review of stability analysis of continuous-time recurrent neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 7, pp. 1229–1262, 2014.

[6] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 157–166, 1994.

[7] Q. Y. An, K. J. Bai, L. J. Liu, *et al.*, "A unified information perceptron using deep reservoir computing," *Computers & Electrical Engineering*, vol. 85, article no. 106705, 2020.

[8] O. Orang, P. C. de Lima e Silva, R. Silva, *et al.*, "Randomized high order fuzzy cognitive maps as reservoir computing models: A first introduction and applications," *Neurocomputing*, vol. 512, pp. 153–177, 2022.

[9] M. L. Xu, M. Han, and H. F. Lin, "Wavelet-denoising multiple echo state networks for multivariate time series prediction," *Information Sciences*, vol. 465, pp. 439–458, 2018.

[10] H. C. Chen and D. Q. Wei, "Chaotic time series prediction using echo state network based on selective opposition grey wolf optimizer," *Nonlinear Dynamics*, vol. 104, no. 4, pp. 3925–3935, 2021.

[11] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *Computer Science*, vol. 14, no. 7, pp. 38–39, 2015.

[12] J. P. Gou, B. S. Yu, S. J. Maybank, *et al.*, "Knowledge distillation: A survey," *International Journal of Computer Vision*, vol. 129, no. 6, pp. 1789–1819, 2021.

[13] T. Furlanello, Z. C. Lipton, M. Tschannen, *et al.*, "Born-again neural networks," in *Proceedings of the 35th International Conference on Machine Learning*, Stockholm, Sweden, pp. 1602–1611, 2018.

[14] A. Romero, N. Ballas, S. E. Kahou, *et al.*, "FitNets: Hints for thin deep nets," in *Proceedings of the 3rd Conference on Learning Representations*, San Diego, CA, USA, pp. 1–13, 2015.

[15] J. Yim, D. Joo, J. Bae, *et al.*, "A gift from knowledge distillation: Fast optimization, network minimization and transfer learning," in *Proceedings of 2017 IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, HI, USA, pp. 7130–7138, 2017.

[16] S. G. Deng, H. L. Zhao, W. J. Fang, *et al.*, "Edge intelligence: The confluence of edge computing and artificial intelligence," *IEEE Internet of Things Journal*, vol. 7, no. 8, pp. 7457–7469, 2020.

[17] Y. F. Shen, Y. M. Shi, J. Zhang, *et al.*, "Graph neural networks for scalable radio resource management: Architecture design and theoretical analysis," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 1, pp. 101–115, 2021.

[18] T. Wang, Y. C. Lu, J. H. Wang, *et al.*, "EIHDP: Edge-intelligent hierarchical dynamic pricing based on cloud-edge-client collaboration for IoT systems," *IEEE Transactions on Computers*, vol. 70, no. 8, pp. 1285–1298, 2021.

[19] J. F. Chen and X. Wang, "Non-intrusive load monitoring using gramian angular field color encoding in edge computing," *Chinese Journal of Electronics*, vol. 31, no. 4, pp. 595–603, 2022.

[20] U. Thakker, J. Beu, D. Gope, *et al.*, "Run-time efficient RNN compression for inference on edge devices," in *Proceedings of the 2nd Workshop on Energy Efficient Machine Learning and Cognitive Computing for Embedded Applications*, Washington, DC, USA, pp. 26–30, 2019.

[21] L. B. Ma, X. Y. Wang, X. W. Wang, *et al.*, "TCDA: Truthful combinatorial double auctions for mobile edge computing in industrial internet of things," *IEEE Transactions on Mobile Computing*, vol. 21, no. 11, pp. 4125–4138, 2022.

[22] X. Y. Peng, J. X. Yu, B. W. Yao, *et al.*, "A review of FPGA-based custom computing architecture for convolutional neural network inference," *Chinese Journal of Electronics*, vol. 30, no. 1, pp. 1–17, 2021.

[23] Z. Zhou, X. Chen, E. Li, *et al.*, "Edge intelligence: Paving the last mile of artificial intelligence with edge computing," *Proceedings of the IEEE*, vol. 107, no. 8, pp. 1738–1762, 2019.

[24] L. Wang, J. F. Qiao, C. L. Yang, *et al.*, "Pruning algorithm for modular echo state network based on sensitivity analysis," *Acta Automatica Sinica*, vol. 45, no. 6, pp. 1136–1145, 2019.

[25] A. Rodan and P. Tino, "Minimum complexity echo state network," *IEEE Transactions on Neural Networks*, vol. 22, no. 1, pp. 131–144, 2011.

[26] X. C. Sun, H. Y. Cui, R. P. Liu, *et al.*, "Multistep ahead prediction for real-time VBR video traffic using deterministic echo state network," in *Proceedings of IEEE 2nd International Conference on Cloud Computing and Intelligence Systems*, Hangzhou, China, pp. 928–931, 2012.

[27] J. Zhou, X. Y. Yang, L. J. Sun, *et al.*, "Network traffic prediction method based on improved echo state network," *IEEE Access*, vol. 6, pp. 70625–70632, 2018.

[28] J. F. Qiao, F. J. Li, H. G. Han, *et al.*, "Growing echo-state network with multiple subreservoirs," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 2, pp. 391–404, 2017.

[29] Y. Kawai, J. Park, and M. Asada, "A small-world topology enhances the echo state property and signal propagation in reservoir computing," *Neural Networks*, vol. 112, pp. 15–23, 2019.

[30] H. S. Wang and X. F. Yan, "Improved simple deterministically constructed cycle reservoir network with sensitive iterative pruning algorithm," *Neurocomputing*, vol. 145, pp. 353–362, 2014.

[31] S. Scardapane, G. Nocco, D. Comminiello, *et al.*, "An effective criterion for pruning reservoir's connections in echo state networks," in *Proceedings of 2014 International Joint Conference on Neural Networks*, Beijing, China, pp. 1205–1212, 2014.

[32] D. Y. Li, F. Liu, J. F. Qiao, *et al.*, "Structure optimization for echo state network based on contribution," *Tsinghua Science and Technology*, vol. 24, no. 1, pp. 97–105, 2019.

[33] H. S. Wang, Y. X. Liu, P. Lu, *et al.*, "Echo state network with logistic mapping and bias dropout for time series prediction," *Neurocomputing*, vol. 489, pp. 196–210, 2022.

[34] K. Greff, R. K. Srivastava, J. Koutník, *et al.*, "LSTM: A search space odyssey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 10, pp. 2222–2232, 2017.

[35] H. Jaeger, M. Lukoševičius, D. Popovici, *et al.*, "Optimization and applications of echo state networks with leaky-integrator neurons," *Neural Networks*, vol. 20, no. 3, pp. 335–352, 2007.

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

[37] J. Chen, J. C. Li, Y. Li, *et al.*, "Multiply accumulate operations in memristor crossbar arrays for analog computing," *Journal of Semiconductors*, vol. 42, no. 1, article no. 013104, 2021.

[38] T. X. Shu, J. H. Chen, V. K. Bhargava, *et al.*, "An energy-efficient dual prediction scheme using LMS filter and LSTM in wireless sensor networks for environment monitoring," *IEEE Internet of Things Journal*, vol. 6, no. 4, pp. 6736–6747,

2019.

[39] J. Zhou, T. T. Han, F. Xiao, *et al.*, "Multiscale network traffic prediction method based on deep echo-state network for internet of things," *IEEE Internet of Things Journal*, vol. 9, no. 21, pp. 21862–21874, 2022.

[40] A. Jain, K. Nandakumar, and A. Ross, "Score normalization in multimodal biometric systems," *Pattern Recognition*, vol. 38, no. 12, pp. 2270–2285, 2005.

Jian ZHOU was born in 1984. He received the Ph.D. degree from Nanjing University of Science and Technology, Nanjing, China, in 2012. He is currently a Professor in Nanjing University of Posts and Telecommunications, Nanjing, China. His recent research interests include edge intelligence, edge computing and time-series prediction.
(Email: zhoujian@njupt.edu.cn)

Yuwen JIANG was born in 1998. He received the B.S. degree in Nanjing University of Posts and Telecommunications, Nanjing, China, in 2020. He is currently pursuing the M.S. degree with the College of Computer, Nanjing University of Posts and Telecommunications, Nanjing, China. His recent research interests include edge intelligence, edge computing and wireless sensor networks.
(Email: 1220045023@njupt.edu.cn)

Lijie XU was born in 1983. He received the Ph.D. degree from Nanjing University, Nanjing, China, in 2014. He is currently an Associate Professor in Nanjing University of Posts and Telecommunications, Nanjing, China. His research interests include wireless rechargeable sensor networks, edge computing, mobile and distributed computing.
(Email: ljxu@njupt.edu.cn)

Lu ZHAO was born in 1990. He received the Ph.D. degree from Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2021. He is currently a Lecturer in Nanjing University of Posts and Telecommunications, Nanjing, China. His recent research interests include service computing, crowdsensing and edge computing.
(Email: luzhao@njupt.edu.cn)

Fu XIAO was born in 1980. He received the Ph.D. degree from Nanjing University of Science and Technology, Nanjing, China, in 2007. He is currently a Professor in Nanjing University of Posts and Telecommunications, Nanjing, China. His recent research interests include mobile computing, edge computing and Internet of Things.
(Email: xiaof@njupt.edu.cn)